A New Adaptive Noise Cancellation Scheme in the Presence of Crosstalk

Gagan Mirchandani, Richard L. Zinser, Jr., and Joseph B. Evans

Abstract—**The application of adaptive filters in noise canceling often requires the relative placement of the two transducers at a distance that necessitates a large order filter in order to obtain an adequate output signal-to-noise ratio. A new adaptive filter structure is introduced that permits a closer placement of the transducers and that allows the cancellation of noise in the presence of crosstalk. Algorithms are developed for the new transversal and lattice filter estimators. Simulations show considerable improvement in mean-square error over that obtained with standard noise canceling algorithms.**

I. Introduction

Studies on the performance of the linear predictive coder (LPC) in the presence of high-frequency noise show that the intelligibility of LPC-processed speech as measured by the diagnostic rhyme test (DRT) often falls below the minimum acceptable level [1]. Under high acoustic noise environments, performance of LPC vocoders can be seriously degraded when noise levels exceed 70 dB [2]. Noise levels ranging from 70 dB to 125 dB are commonly found in industrial environments in the presence of vibrations and accelerations. Several techniques exist for the enhancement of signals degraded by additive noise [3].

Adaptive noise cancellation represents one such potentially effective technique. However, the problem with this approach is that it is difficult to obtain a highly correlated noise signal with the reference transducer, without simultaneously obtaining a correlated speech signal [4]. Simulation experiments in a noisy environment show that filters requiring in excess of 1500 taps are often necessary to achieve desired noise reduction levels [5]. For applications requiring simple hardware, this would be difficult to implement in a real-time environment. In addition, long filter lengths can cause large misadjustment errors in the filter tap weights. Due to the feedback nature of an adaptive algorithm, the misadjustment errors can lead to echoes in the filtered speech [5]. A closer placement of the two transducers would reduce the order of the filter, were it not for the correlated speech or crosstalk that is consequently induced in the reference signal. Crosstalk has two effects: 1) the desired signal present in the reference transducer causes signal cancellation in the output, thus directly reducing the output signal-to-noise ratio, and 2) the adaptive algorithm can become biased so that the filter never converges to the correct model. Of these two effects, the second is the more important in actual practice.

Crosstalk occurs in many other environments. In time-division multiplexing, crosstalk, referred to as intersymbol interference, occurs when a pulse is spread out during transmission into an adjacent time slot. This effect is often alleviated using adaptive equalizers at the transmission line terminals. In frequency-division multiplexing, crosstalk results when a signal $s_i(t)$ modulates a carrier at frequency f_j assigned to a different signal $s_j(t)$. When due to nonlinearities, this effect is often controlled through use of feedback techniques. Another important source of crosstalk is coupling between physically separated but adjacent transmission media. In adjacent-channel interference (ACI) components of the desired signal, i.e., crosstalk, may be present in the reference interference signal, thereby rendering traditional adaptive noise canceling techniques ineffective. Several methods, including the one proposed in this paper have been attempted [6]–[8] to resolve the problem. Another example of a crosstalk-like effect in adaptive filtering occurs in echo cancellation where the echo is ideally free of near-end speech. In reality, this is not the case for there often is near-end speech present—a phenomenon referred to as doubletalk. Very little work has been done to address this problem in the context of adaptive filtering [9].

The effect of crosstalk in adaptive noise canceling has been determined for the unconstrained Wiener solution [10]. In practice, the observed effects are much more severe than those described in [10] because of the estimator bias introduced by the crosstalk [11]. Fig. 1 shows a block diagram of the adaptive noise canceller. Assuming that the signal is stationary and that the signal and noise are jointly stationary, the output signal-to-noise density ratio, after convergence, is

$$\rho_{\text{out}} = \frac{1}{\rho_{\text{ref}}(z)}$$

where

$$\rho(z) = \frac{\text{Signal Power Spectral Density}}{\text{Noise Power Spectral Density}}.$$

Manuscript received November 20, 1991; revised October 2, 1992. This paper was recommended by Associate Editor G. S. Moschytz.

G. Mirchandani is with the Department of Computer Science and Electrical Engineering, University of Vermont, Burlington, VT 05401. This work was completed while he was on a sabbatical at General Electric Company, CRD, Schenectady, NY 12311.

R. L. Zinser, Jr., is with General Electric Company, Corporate Research and Development, Schenectady, NY 12301.

J. D. Evans is with the Department of Electrical and Computer Engineering, University of Kansas, Lawrence, KS 66045-2228.

IEEE Log Number 9205186.

1057-7130/92$03.00

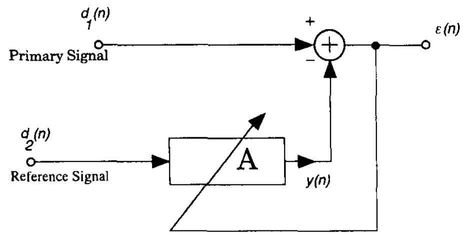


Fig. 1. Adaptive noise canceller.

The simulated effect of crosstalk with the causal adaptive noise canceller (ANC) were obtained using varying levels of crosstalk and different algorithms for the filter. A primary signal $s_1 + n_1$ consisting of 8 s of sinusoidal signal and 8 s of noise, each at 50-dB levels, and a reference signal $s_2 + n_2$ consisting of white uniformly distributed noise (50 dB) and crosstalk at levels of 1 dB, 0 dB, -2 dB, -7 dB, and -9 dB with respect to the primary signal, were utilized. For the reference, $s_2(n)$ was obtained as the output of a filter with transfer function $P_1(z^{-1}) = 0.4z^{-3} + 0.12z^{-2} - 0.1z^{-1} - 0.1$ and input $s_1(n)$. Similarly, colored noise $n_1(n)$ was obtained as the output of a filter with transfer function $P_2(z^{-1}) = z^{-3} + 0.24z^{-2} + 0.298z^{-1} + 0.058$ and input $n_2(n)$. All signals were sampled at 8 kHz. The primary and reference signals were processed through the stochastic gradient transversal, stochastic gradient lattice, least-squares lattice and fast Kalman adaptive filtering algorithms. Fig. 2 shows the mean-square error for the least-squares lattice for all five crosstalk-levels. Since the noise was white, results for the other configurations were not appreciably different. For these situations, as seen in the figure, the standard adaptive noise canceller provides, at best, very little benefit.

II. A New Structure

The problem, then, is the design of a structure for joint process estimation that will eliminate noise in the presence of crosstalk. The proposed structure, a dual joint process estimator, is shown in Fig. 3 [12]. In the noise canceling context it is referred to as a crosstalk-resistant adaptive noise canceller (CTRANC). To the standard adaptive noise estimator (filter A), a second adaptive filter (filter B) is added to estimate the crosstalk. That estimate is then subtracted from the reference input to provide a crosstalk-free signal to the noise estimator.

A. Transversal Filter Based Formulation

The operation of CTRANC may be understood as follows. In Fig. 3 assume that the primary signal $d_1(n)$ and reference signal $d_2(n)$ each consist of signal plus additive noise such that

$$d_1(n) = s_1(n) + n_1(n) \tag{1}$$

and

$$d_2(n) = s_2(n) + n_2(n) \tag{2}$$

where both signal and noise are, for the present discussion, considered to be deterministic signals. Furthermore, let it be assumed that $n_1(n)$ and $s_2(n)$ are delayed versions of $n_2(n)$ and $s_1(n)$, respectively, such that

$$N_1(z) = A(z^{-1})N_2(z) \tag{3}$$

$$S_2(z) = B(z^{-1})S_1(z) \tag{4}$$

where $A(z^{-1})$ and $B(z^{-1})$ are finite order polynomials in z^{-1} and $S_i(z), N_i(z)$ are z-transforms of $s_i(n), n_i(n)$, respectively, for $i = 1, 2$. This may physically exist for example, in a situation where the primary transducer receiving signal $s_1 + n_1$ is closer to the signal source while the secondary transducer receiving signal $s_2 + n_2$ is closer to the noise source. Furthermore, let the filters A and B be fixed and represented respectively by transfer functions $A(z^{-1})$ and $B(z^{-1})$. Expressing input–output relationships in z-transforms, where $E_i(z), D_i(z), Y_i(z)$ are z-transforms of $\epsilon_i(n), d_i(n)$ and $y_i(n)$ for $i = 1, 2$, respectively, we have

$$E_1(z) = D_1(z) - Y_1(z) \tag{5}$$

$$\begin{aligned} &= D_1(z) - A(z^{-1})E_2(z) \\ &= D_1(z) - A(z^{-1})[S_2(z) + N_2(z) - Y_2(z)] \\ &= D_1(z) - A(z^{-1})S_2(z) - A(z^{-1})N_2(z) \\ &\quad + A(z^{-1})B(z^{-1})E_1(z) \\ &= S_1(z) - A(z^{-1})B(z^{-1})S_1(z) \\ &\quad + A(z^{-1})B(z^{-1})E_1(z). \end{aligned} \tag{6}$$

Hence

$$E_1(z) - S_1(z) = [E_1(z) + S_1(z)]A(z^{-1})B(z^{-1}). \tag{7}$$

We assume that the network is *computable*, that is, there are no closed loops without delays in the network. Accordingly, since $A(z^{-1})$ and $B(z^{-1})$ cannot, by definition, be inverse functions, it follows that $\epsilon_1(n) = s_1(n)$, the desired signal. Furthermore, from (5) it is easily seen that

$$E_1(z) = D_1(z) - A(z^{-1})\left[D_2(z) - B(z^{-1})E_1(z)\right] \tag{8}$$

and

$$\begin{aligned} &[1 - A(z^{-1})B(z^{-1})]E_1(z) \\ &\quad = D_1(z) - A(z^{-1})D_2(z). \end{aligned} \tag{9}$$

Therefore,

$$E_1(z) = \frac{D_1(z)}{1 - A(z^{-1})B(z^{-1})} - \frac{A(z^{-1})}{1 - A(z^{-1})B(z^{-1})}D_2(z). \tag{10}$$

Hence, for the deterministic case, the feedback configuration is stable as long as the zeroes of $[1 - A(z^{-1})B(z^{-1})]$ are maintained within the unit circle. Fig. 4 illustrates the cancellation scheme.

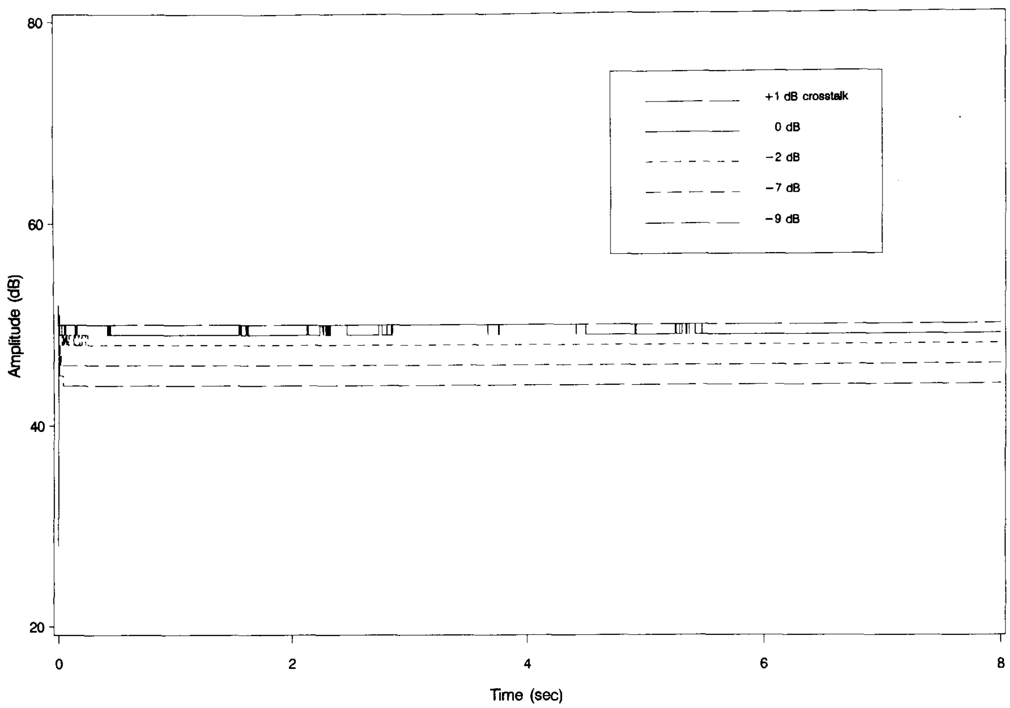


Fig. 2. Mean-square error for least-squares lattice-based ANC.

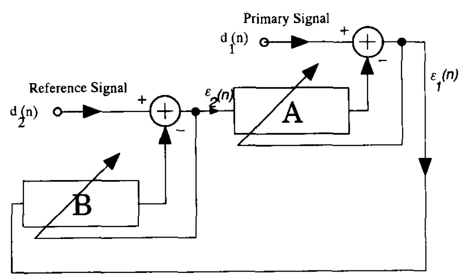


Fig. 3. Crosstalk resistant ANC.

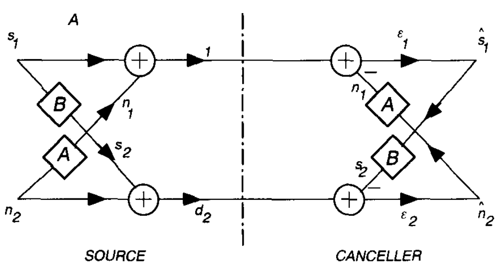


Fig. 4. Crosstalk cancellation scheme.

For the case where the signals are random processes assume that $d_1(n), d_2(n), \epsilon_1(n), \epsilon_2(n)$ are real wide-sense stationary (WSS) processes and jointly WSS when taken in pairs. We have

$$\epsilon_1(n) = d_1(n) - y_1(n). \tag{11}$$

Hence,

$$\begin{aligned} E\{\epsilon_1(k)\epsilon_1(n+k)\} &= E\{[d_1(k) - y_1(k)][d_1(n+k) - y_1(n+k)]\} \\ &= E\{d_1(k)d_1(n+k) - y_1(k)d_1(n+k) \\ &\quad - d_1(k)y_1(n+k) + y_1(k)y_1(n+k)\}. \end{aligned} \tag{12}$$

Hence,

$$r_{\epsilon_1\epsilon_1}(n) = r_{d_1d_1}(n) - r_{d_1y_1}(n) - r_{y_1d_1}(n) + r_{y_1y_1}(n) \tag{13}$$

$$= r_{d_1d_1}(n) - r_{d_1y_1}(n) + r_{d_1y_1}(-n) + r_{y_1y_1}(n) \tag{14}$$

where $r_{\epsilon_1\epsilon_1}(n), r_{d_1d_1}(n), r_{d_1y_1}(n)$, and $r_{y_1y_1}(n)$ are autocorrelation and cross-correlation sequences. With the z-transform of a sequence defined as power spectrum, we have from (14)

$$S_{\epsilon_1\epsilon_1}(z) = S_{d_1d_1}(z) + S_{d_1y_1}(z^{-1}) - S_{d_1y_1}(z) + S_{y_1y_1}(z). \tag{15}$$

With constant coefficient filters $A(z^{-1})$ and $B(z^{-1})$, we have $S_{y_1y_1}(z) = |A(z)|^2 S_{\epsilon_2\epsilon_2}(z)$, $S_{d_1y_1}(z) = A(z^{-1})S_{d_1\epsilon_2}(z)$, and $S_{d_1y_1}(z^{-1}) = A(z)S_{d_1\epsilon_2}(z^{-1})$. Hence

$$S_{\epsilon_1\epsilon_1}(z) = S_{d_1d_1}(z) + A(z)S_{d_1\epsilon_2}(z^{-1}) - A(z^{-1})S_{d_1\epsilon_2}(z) + |A(z)|^2 S_{\epsilon_2\epsilon_2}(z). \tag{16}$$

Similarly, it can be seen that

$$S_{\epsilon_2\epsilon_2}(z) = S_{d_2d_2}(z) + B(z)S_{d_2\epsilon_1}(z^{-1}) - B(z^{-1})S_{d_2\epsilon_1}(z) + |B(z)|^2 S_{\epsilon_1\epsilon_1}(z). \tag{17}$$

Substituting for $S_{\epsilon_2\epsilon_2}(z)$ in (16), it is easily seen that

$$\begin{aligned} S_{\epsilon_1\epsilon_1}(z) &= S_{d_1d_1}(z) + A(z)S_{d_1\epsilon_2}(z^{-1}) - A(z^{-1})S_{d_1\epsilon_2}(z) \\ &\quad + |A(z)|^2 S_{d_2d_2}(z) + |A(z)|^2 B(z) S_{d_2\epsilon_2}(z^{-1}) \\ &\quad - |A(z)|^2 B(z^{-1}) S_{d_2\epsilon_1}(z) \\ &\quad + |A(z)|^2 |B(z)|^2 S_{\epsilon_1\epsilon_1}(z). \end{aligned} \tag{18}$$

Hence,

$$\begin{aligned} &\left(1 - |A(z)|^2|B(z)|^2\right) S_{\epsilon_1\epsilon_1}(z) \\ &\quad = S_{d_1d_2}(z) + |A(z)|^2 S_{d_2d_2}(z) \\ &\qquad + S_{d_1y_1}(z^{-1}) - S_{d_1y_1}(z) \\ &\qquad + |A(z)|^2 S_{d_2y_2}(z^{-1}) \\ &\qquad - |A(z)|^2 S_{d_2y_2}(z) \end{aligned} \tag{19}$$

where we have also utilized $S_{d_2y_2}(z^{-1}) = B(z)S_{d_2\epsilon_1}(z^{-1})$ and $S_{d_2y_2}(z) = B(z^{-1})S_{d_2\epsilon_1}(z)$. Accordingly, to maintain a stable output process $\epsilon_1(n)$ in the CTRANC feedback structure, it is necessary that the zeroes of $(1 - |A(z)|^2|B(z)|^2)$ lie within the unit circle.

B. Mean-Square Error Surface

Let $s_i(n)$ and $n_i(n)$, $i = 1, 2$ be real, discrete-time, random processes. The error signals at time n are given by

$$\epsilon_i(n) = d_1(n) - \underline{w}^{A^T}(n)\underline{\epsilon}_2(n) \tag{20}$$

$$\epsilon_2(n) = d_2(n) - \underline{w}^{B^T}(n)\underline{\epsilon}_1(n) \tag{21}$$

where T denotes the transpose operation,

$$\underline{w}^{A^T}(n) = \left[w_0^A(n) w_1^A(n) w_2^A \cdots w_N^A(n)\right]$$
$$\underline{w}^{B^T}(n) = \left[w_0^B(n) w_1^B(n) w_2^B(n) \cdots w_N^B(n)\right]$$

are the time-varying filter weights for the Nth-order filters A and B, and

$$\underline{\epsilon}_1^T(n) = [\epsilon_1(n)\epsilon_1(n-1) \cdots \epsilon_1(n-N)]$$
$$\underline{\epsilon}_2^T(n) = [\epsilon_2(n)\epsilon_2(n-1) \cdots \epsilon_2(n-N)].$$

The filter weights are to be adjusted so as to minimize the mean-square errors $\xi_1 = E\{\epsilon_1^2(n)\}$ and $\xi_2 = E\{\epsilon_2^2(n)\}$. The form of the error surfaces ξ_1 and ξ_2 can be determined as follows: Consider the filter weights to be fixed, that is,

$$\underline{w}^{A^T} = \left[w_0^A w_1^A \quad \cdots \quad w_N^A\right]$$
$$\underline{w}^{B^T} = \left[w_0^B w_1^B \quad \cdots \quad w_N^B\right].$$

Then, the mean-square error is given by

$$\begin{aligned} \xi_1 &= E\{[d_1(n) - y_1(n)]\}^2 \\ &= E\left\{\left[d_1(n) - \sum_{k=0}^{N} w_k^A \epsilon_2(n-k)\right]\right\}^2. \end{aligned} \tag{22}$$

Since

$$\begin{aligned} \epsilon_2(r) &= d_2(r) - \sum_{s=0}^{N} w_s^B \epsilon_1(r-s) \\ \xi_1 &= E\left\{d_1(n) - \sum_{k=0}^{N} w_k^A \left[d_2(n-k) - \sum_{s=0}^{N} w_s^B \epsilon_1(n-k-s)\right]\right\}^2 \end{aligned} \tag{23}$$

$$= E\left\{\left[d_1(n) - \sum_{k=0}^{N} w_k^A d_2(n-k) + \sum_{k=0}^{N}\sum_{s=0}^{N} w_k^A w_s^B \epsilon_1(n-k-s)\right]\right\}^2 \tag{24}$$

it is easily seen that

$$\begin{aligned} \xi_1 = E\Bigg\{ & d_1^2(n) - 2\sum_{m=0}^{N} d_1(n) d_2(n-m) w_m^A \\ & + 2\sum_{t=0}^{N}\sum_{m=0}^{N} d_1(n)\epsilon_1(n-m-t) w_m^A w_t^B \\ & + \sum_{k=0}^{N}\sum_{m=0}^{N} d_2(n-m) d_2(n-k) w_m^A w_k^A \\ & - 2\sum_{k=0}^{N}\sum_{t=0}^{N}\sum_{m=0}^{N} d_n(n-k)\epsilon_1(n-m-t) w_k^A w_m^A w_t^B \\ & + \sum_{k=0}^{N}\sum_{s=0}^{N}\sum_{t=0}^{N}\sum_{m=0}^{N} \epsilon_1(n-k-s)\epsilon_1 \\ & \cdot (n-m-t) w_k^A w_s^A w_m^B w_t^B \Bigg\}. \end{aligned} \tag{25}$$

Assuming that $d_1(n), d_2(n), \epsilon_1(n), \epsilon_2(n)$ are WSS processes and jointly WSS when taken in pairs we have

$$\begin{aligned} \xi_1 = & r_{d_1d_1}(0) - 2\sum_{m=0}^{N} w_m^A r_{d_1d_2}(m) \\ & + 2\sum_{t=0}^{N}\sum_{m=0}^{N} w_t^A w_m^B r_{d_1\epsilon_1}(m+t) \\ & + \sum_{k=0}^{N}\sum_{m=0}^{N} w_k^A w_m^A r_{d_2d_2}(k-m) - 2\sum_{k=0}^{N}\sum_{m=0}^{N} w_k^A w_m^A \\ & \cdot \left(\sum_{t=0}^{N} w_t^B r_{d_2\epsilon_1}(t+m-k)\right) + \sum_{k=0}^{N}\sum_{m=0}^{N} w_k^A w_m^A \\ & \cdot \left(\sum_{s=0}^{N}\sum_{t=0}^{N} w_s^B w_t^B r_{\epsilon_1\epsilon_1}(m+t-k-s)\right) \end{aligned} \tag{26}$$

where

$$r_{d_id_i}(n) = E\{d_i(m)d_i(m-n)\}, \quad i = 1, 2$$
$$r_{\epsilon_i\epsilon_i}(n) = E\{\epsilon_i(m)\epsilon_i(m-n)\}, \quad i = 1, 2$$
$$r_{d_i\epsilon_i}(n) = E\{d_i(m)\epsilon_i(m-n)\}, \quad i = 1, 2$$

are the autocorrelation and cross-correlation sequences associated with the corresponding random processes. The second mean-square error $\xi_2 = E\{\epsilon_2^2(n)\}$ can be derived in a similar fashion or obtained from (26) simply by interchanging variables $\epsilon_1(n)$, $d_1(n)$, $y_1(n)$ and weights w^4 with variables $\epsilon_2(n)$, $d_2(n)$, $y_2(n)$, and w^B, respectively.

The error surface in matrix notation can be written as

$$\xi_1 = r_{d_1d_1}(0) - 2\underline{r}_{d_1d_2}^T\underline{w}^A + 2\underline{w}^{A^T}\underline{R}_{d_1\epsilon_2}\underline{w}^B + \underline{w}^{A^T}\underline{R}_{d_2d_2}\underline{w}^A - 2\underline{w}^{A^T}\underline{R}_{w^B}\underline{w}^A + \underline{w}^{A^T}\underline{R}_{w^Bw^B}\underline{w}^A \tag{27}$$

where

$$\underline{r}_{d_1d_2}^T = \left[r_{d_1d_2}(0)r_{d_1d_2}(1)r_{d_1d_2}(2)\cdots r_{d_1d}(N)\right]$$

$$\underline{R}_{d_1\epsilon_1} = \begin{bmatrix} r_{d_1\epsilon_1}(0) & r_{d_1\epsilon_1}(1) & \cdots & r_{d_1\epsilon_1}(N) \\ r_{r_1\epsilon_1}(1) & r_{d_1\epsilon_1}(2) & \cdots & r_{d_1\epsilon_1}(N+1) \\ \vdots & \vdots & & \vdots \\ r_{d_1\epsilon_1}(N) & r_{d_1\epsilon_1}(N+1) & \cdots & r_{d_1\epsilon_1}(2N) \end{bmatrix}$$

$$\underline{R}_{d_2d_2} = \left[r_{d_2d_2}(i-j)\right] \qquad 0 \le i, j, \le N$$

$$\underline{R}_{w^B} = \left[r(i-j)\right] \qquad 0 \le i, j, \le N$$

$$R_{w^Bw^B} = \left[r_{y_2y_2}(i-j)\right] \qquad 0 \le i, j, \le N.$$

This may be seen as follows. Identification of the first four terms in (26) with those in (27) is straightforward. In the triple sum, the inner sum constitutes the elements of the matrix $\underline{R}_{w^B}$. The (i, j) term of this matrix is

$$\sum_{k=0}^{N} w_k^B r_{d_2\epsilon_1}(k+j-i) = \sum_{k=0}^{N} w_k^B r_{\epsilon_1d_2}[(i-j)-k] \equiv r(i-j) \tag{28}$$

which represents the convolution of the cross-correlation sequence $r_{\epsilon_1d_2}(n)$ with the filter B weights. It is noted that were the filter coefficients w_k^B in (28) to be time reversed by replacing k by $-k$, then the output sequence would be $r_{y_2d_2}(i-j)$. In the last term in (26), the inner double sum represents the output autocorrelation sequence of filter B, evaluated at instance $(m-k)$.

In an analogous fashion, the second mean-square error surface is written as

$$\xi_2 = r_{d_2d_2}(0) - 2\underline{r}_{d_2d_1}^T\underline{w}^B + 2\underline{w}^{B^T}R_{d_2\epsilon_2}\underline{w}^A + \underline{w}^{B^T}\underline{R}_{d_1d_1}\underline{w}^B - 2\underline{w}^{B^T}\underline{R}_{w^A}\underline{w}^B + \underline{w}^{B^T}\underline{R}_{w^Aw^A}\underline{w}^B. \tag{29}$$

Hence it is seen that the error surfaces ξ_1 and ξ_2 are linear, bilinear, and quadratic forms of vectors $\underline{w}^A$ and $\underline{w}^B$, respectively, with coefficients of the quadratic forms that are linear and quadratic in vectors $\underline{w}^B$ and $\underline{w}^A$, respectively. Alternatively, the error surfaces may be seen as linear and quadratic in weights $\underline{w}^A$ or $\underline{w}^B$ when the other weights are fixed.

C. Optimal Filters

The necessary conditions for the optimal filter weight vectors $\underline{w}^A$ and $\underline{w}^B$ can be obtained from minimization of error surfaces ξ_1 and ξ_2. The problem is decomposed into two minimization problems. We choose to minimize ξ_1 and ξ_2 with respect to $\underline{w}^A$ and $\underline{w}^B$, respectively. For ξ_1, we must have the gradient vector

$$\nabla^A\xi_1 = (\nabla_0^A\xi_1\nabla_1^A\xi_1 \cdots \nabla_n^A\xi_1)^T = \underline{0} \tag{30}$$

where $\underline{0}$ is the $(N+1)$ dimensional vector of zeroes, and

$$\nabla_i^A\xi_1 = \frac{\partial\xi_1}{\partial w_i^A} \qquad i = 0, 1, \cdots, N.$$

Since the last two terms in (27) are not linear or quadratic terms, the formulation of the gradient can best be obtained from the scalar equation (26) by setting

$$\frac{\partial\xi_1}{\partial w_q^A} = 0 \qquad 0 \le q \le N.$$

Accordingly,

$$\begin{aligned}\frac{\partial\xi_1}{\partial w_q^A} = &-2r_{d_1d_2}(q) + 2\sum_{k=0}^{N} w_k^B r_{d_1\epsilon_1}(q+k) \\ &+2\sum_{k=0}^{N} w_k^A r_{d_2d_2}(k-q) \\ &-2\sum_{k=0}^{N}\sum_{m=0}^{N} w_k^A w_m^B r_{d_2\epsilon_1}(m+q-k) \\ &-2\sum_{k=0}^{N}\sum_{m=0}^{N} w_k^A w_m^B r_{d_2\epsilon_1}(m+k-q) \\ &+\sum_{k=0}^{N}\sum_{m=0}^{N}\sum_{p=0}^{N} w_k^A w_m^B w_p^B r_{\epsilon_1\epsilon_1} \\ &\cdot(m+k-q-p) \\ &+\sum_{k=0}^{N}\sum_{m=0}^{N}\sum_{p=0}^{N} w_k^A w_m^B w_p^B r_{\epsilon_1\epsilon_1} \\ &\cdot(m+q-p-k) = 0 \\ &\qquad 0 \le q \le N \end{aligned} \tag{31}$$

which may be written as

$$\begin{aligned}&-2r_{d_1d_2}(q) + 2\left[r_{d_1\epsilon_1}(1)\ r_{d_1\epsilon_1}(1+q)\ \cdots r_{d_1\epsilon_1}(N+q)\right]\underline{w}^B \\ &+2\left[r_{d_2d_2}(-q)\ r_{d_2d_2}(1-q)\ \cdots r_{d_2d_2}(N-q)\right]\underline{w}^A \\ &-2\underline{w}^{A^T}\underline{R}_{d_2\epsilon_1}(q)\underline{w}^B - 2\underline{w}^{A^T}\hat{R}_{d_2\epsilon_1}(q)\underline{w}^B \\ &+\left[\underline{w}^{A^T}A_0(q)\underline{w}^B\ \ \underline{w}^{A^T}A_1(q)\underline{w}^B \cdots \underline{w}^{A^T}A_N(q)\underline{w}^B\right]\underline{w}^B \\ &+\left[\underline{w}^{A^T}B_0(q)\underline{w}^B\ \ \underline{w}^{A^T}B_1(q)\underline{w}^B \cdots \underline{w}^{A^T}B_N(q)\underline{w}^B\right]\underline{w}^B = 0 \end{aligned} \tag{32}$$

where

$$R_{d_2\epsilon_1}(q) = \begin{bmatrix} r_{d_2\epsilon_2}(q) & r_{d_2\epsilon_1}(q+1) & \cdots & r_{d_2\epsilon_1}(q+N) \\ r_{d_2\epsilon_1}(q-1) & r_{d_2\epsilon_1}(q) & \cdots & r_{d_2\epsilon_1}(q+N-1) \\ \vdots & \vdots & & \vdots \\ r_{d_2\epsilon_1}(q-N) & r_{d_2\epsilon_1}(q-N+1) & \cdots & r_{d_2\epsilon_2}(q) \end{bmatrix}$$

$$\hat{R}_{d_2\epsilon_1}(q) = \begin{bmatrix} r_{2\epsilon_1}(-q) & r_{d_2\epsilon_1}(1-q) & \cdots & r_{d_2\epsilon_1}(N-q) \\ r_{d_2\epsilon_1}(1-q) & r_{d_2\epsilon_1}(2-q) & \cdots & r_{d_2\epsilon_1}(N+1-q) \\ \vdots & \vdots & & \vdots \\ r_{d_2\epsilon_1}(N-q) & r_{d_1\epsilon_1}(N+1-q) & \cdots & r_{d_2\epsilon_1}(2N-q) \end{bmatrix}$$

$$A_i(q) = \begin{bmatrix} r_{\epsilon_1\epsilon_1}(-q+i) & r_{\epsilon_1\epsilon_1}(-q+i-1) & \cdots & r_{\epsilon_1\epsilon_1}(-q+i-N) \\ r_{\epsilon_1\epsilon_1}(-q+i+1) & r_{\epsilon_1\epsilon_1}(-q+i) & \cdots & r_{\epsilon_1\epsilon_1}(-q+i+1-N) \\ \vdots & \vdots & & \vdots \\ r_{\epsilon_1\epsilon_1}(-q+i+N) & r_{\epsilon_1\epsilon_1}(-q+i+N-1) & \cdots & r_{\epsilon_1\epsilon_1}(-q+i) \end{bmatrix}$$

$$B_i(q) = \begin{bmatrix} r_{\epsilon_1\epsilon_1}(q+i) & r_{\epsilon_1\epsilon_1}(q+i-1) & \cdots & r_{\epsilon_1\epsilon_1}(q+i-N) \\ r_{\epsilon_1\epsilon_1}(q+i-1) & r_{\epsilon_1\epsilon_1}(q+i) & \cdots & r_{\epsilon_1\epsilon_1}(q+i-1+N) \\ \vdots & \vdots & & \vdots \\ r_{\epsilon_1\epsilon_1}(q+i-N) & r_{\epsilon_1\epsilon_1}(q+i-1+N) & \cdots & r_{\epsilon_1\epsilon_1}(q+i) \end{bmatrix}$$

where $0 \le i \le N$ and $0 \le q \le N$.

Hence, setting $\nabla^A_{\xi_1} = \underline{0}$ gives

$$2\underline{R}_{d_1\epsilon_1}\underline{w}^B + 2\underline{R}_{d_2d_2}\underline{w}^A - 2\underline{r}_{d_1d_2}$$
$$= \begin{bmatrix} 2\underline{w}^{A^T}\left(\underline{R}_{d_2\epsilon_1}(0) + \hat{\underline{R}}_{d_2\epsilon_1}(0)\right)\underline{w}^B & -\left[\underline{w}^{A^T}\underline{A}(0)\underline{w}^B \right. & \cdots & \left.\underline{w}^{A^T}\underline{A}_N(0)\underline{w}^B\right]\underline{w}^B \\ \vdots & -\left[w^{A^T}\underline{B}_0(0)\underline{w}^B \right. & \cdots & \left.\underline{w}^{A^T}\underline{B}_N(0)\underline{w}^B\right]\underline{w}^B \\ \vdots & \vdots & & \vdots \\ 2\underline{w}^{A^T}\left(\underline{R}_{d_2\epsilon_1}(q) + \left(\hat{\underline{R}}_{d_2\epsilon_1}(q)\right)\underline{w}^B\right. & -\left[\underline{w}^{A^T}\underline{A}_0(q)\underline{w}^B \right. & \cdots & \left.\underline{w}^{A^T}\underline{A}_N(q)\underline{w}^B\right]\underline{w}^B \\ \vdots & +\left[\underline{w}^{A^T}\underline{B}_0(q)\underline{w}^B \right. & \cdots & \left.\underline{w}^{A^T}\underline{B}_N(q)\underline{w}^B\right]\underline{w}^B \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (33)$$

where $\underline{R}_{d_1\epsilon_1}$, $\underline{R}_{d_2d_2}$, and $\underline{r}_{d_1d_2}$ have been defined earlier.

The minimum of the error surface ξ_2 is determined in a similar fashion by setting $\nabla^B\xi_2 = \underline{0}$. The resulting condition is similar to (33) and can be obtained from it by interchanging subscripts 1 and 2 and superscripts A and B. It is seen that (33) is linear in the weights w_i^A with coefficients that are polynomials of degree 2 in w_j^B for $i, j = 0, 1, \cdots, N$.

III. Algorithms

A. Transversal Filter Based Formulation

The adaptive joint process estimation algorithm for transversal filters A and B is generated by minimizing the mean-square errors $\overline{\epsilon_1^2(n)}$ and $\overline{\epsilon_2^2(n)}$ through standard stochastic gradient procedures. Let $\nabla_a\epsilon_1(n)$ and $\nabla_B\epsilon_2(n)$ be the gradients of $\epsilon_1(n)$ and $\epsilon_2(n)$ with respect to the filter weights $w_A(n)$ and $w_B(n)$, respectively, that is,

$$\nabla_A^T\epsilon_1(n) = \left[\frac{\partial\epsilon_1(n)}{\partial w_0^A(n)}\ \frac{\partial\epsilon_1(n)}{\partial w_1^A(n)}\ \frac{\partial\epsilon_1(n)}{\partial w_2^A(n)} \cdots \frac{\partial\epsilon_1(n)}{\partial w_N^A(n)}\right]$$

$$\nabla_B^T\epsilon(n) = \left[\frac{\partial\epsilon_2(n)}{\partial w_0^B(n)}\ \frac{\partial\epsilon_2(n)}{\partial w_1^B(n)}\ \frac{\partial\epsilon_2(n)}{\partial w_2^B(n)} \cdots \frac{\partial\epsilon_2(n)}{\partial w_N^A(n)}\right]. \quad (34)$$

It is seen after some calculation that

$$\nabla_A\epsilon_1(n) - \left[w_0^A(n)w_0^B(n)\right]\nabla_A\epsilon_1(n) = -\epsilon_2(n) + \sum_{k=1}^{2N} C_k(n)\nabla_A\epsilon_1(n-k) \quad (35)$$

and

$$\nabla_B \epsilon_2(n) - \left[w_0^A(n) w_0^B(n)\right] \nabla_B \epsilon_2(n) = -\epsilon_1(n) + \sum_{k=1}^{2N} D_k(n) \nabla_B \epsilon_2(n-k) \quad (36)$$

where $\epsilon_1(n)$ and $\epsilon_2(n)$ have been defined previously, and

$$C_i(n) = \begin{cases} \sum_{j=1}^{i} w_j^A(n) w_{i-j}^B(n-j) & 1 \le i \le N \\ \sum_{j=i-N}^{N} w_j^A(n) w_{i-j}^B(n-j) & N+1 \le i \le 2N \end{cases}$$

and

$$D_i(n) = \begin{cases} \sum_{j=0}^{i} w_j^B(n) w_{i-j}^A(n-j) & 1 \le i \le N \\ \sum_{j=i-N}^{N} w_j^B(n) w_{i-j}^A(n-j) & N+1 \le i \le 2N. \end{cases}$$

Equations (34) and 35) do not, as yet, form explicit recurrence relations for $\nabla_A \epsilon_1(n)$ and $\nabla_B \epsilon_2(n)$ since the previous gradients $\nabla_A \epsilon_1(n-k)$ and $\nabla_B \epsilon_2(n-k)$ are with respect to weights $\omega_A(n)$ and $\omega_B(n)$ and not with respect to the weights $w_A(n-k)$ and $w_B(n-k)$, respectively. Changing variables gives

$$\frac{\partial \epsilon_i(n-k)}{\partial w_j(n)} = \frac{\partial \epsilon_i(n-k)}{\partial w_j(n-k)} \frac{\partial w_j(n-k)}{\partial w_j(n)} \qquad j = 0, 1, 2, \cdots N \quad i = 1, 2. \quad (37)$$

Once the associated derivatives of the filter weights are determined, a recurrence expression for the gradients $\nabla_A \epsilon_1(n)$ and $\nabla_B \epsilon_2(n)$ becomes available.

For minimization of the mean-square errors $\overline{\epsilon_1^2(n)}$ and $\overline{\epsilon_2^2(n)}$, the usual stochastic gradient approximation for the gradient of the mean-square error is assumed. The filter weights are updated according to

$$\begin{aligned} w^A(n+1) &= w^A(n) - \mu_1(n) \nabla_A \epsilon_1^2(n) \\ &= w^A(n) - 2\mu_1(n) \epsilon_1(n) \nabla_A \epsilon_1(n) \quad (38) \end{aligned}$$

and

$$\begin{aligned} w^B(n+1) &= w^B(n) - \mu_2(n) \nabla_B \epsilon_2^2(n) \\ &= w^B(n) - 2\mu_2(n) \epsilon_2(n) \nabla_B \epsilon_2(n) \quad (39) \end{aligned}$$

where (34) and (35) determine $\nabla_A \epsilon_1(n)$ and $\nabla_B \epsilon_2(n)$, respectively, and $\mu_1(n) > 0$, $\mu_2(n) > 0$ for all n are the adaptive step sizes. The adaptation algorithm for the jth filter coefficient is

$$w_j(n+1) = w_j(n) - 2\mu(n) \epsilon(n) \frac{\partial \epsilon(n)}{\partial w_j(n)}.$$

Using this recurrence repeatedly, we have

$$w_j(n-k) = w_j(n) + 2 \sum_{i=1}^{k} \mu(n-i) \epsilon(n-i) \cdot \frac{\partial \epsilon(n-i)}{\partial w_j(n-i)}. \quad (40)$$

Hence, the derivatives required in (36) become

$$\frac{\partial w_j(n-k)}{\partial w_j(n)} = 1 + 2 \frac{\partial}{\partial w_j(n)} \sum_{i=1}^{k} \mu(n-i) \epsilon(n-i) \cdot \frac{\partial \epsilon(n-i)}{\partial w_j(n-i)}. \quad (41)$$

The necessary condition for convergence of the filter weights described by (37) and (38) requires that the error and the derivative of the error be orthogonal. The sum in (40) represents the weighted average of the error and derivative of the error. If the signal and therefore the adaptive step size $\mu(n)$ is assumed to be varying slowly with respect to the error and its derivative, then the step size can be factored out. The sum will, by the orthogonality condition, equal zero at convergence. Hence that term is ignored in the computation of the derivatives of the filter weights. Accordingly, the recursive relationships of (34) and (35) for $\nabla_A \epsilon_1(n)$ and $\nabla_B \epsilon_2(n)$ become

$$\nabla_A \epsilon_1(n) = C_0(n) \left[\sum_{k=1}^{2N} C_k(n) \nabla_A^k \epsilon_1(n-k) - E_2(n) \right] \quad (42)$$

$$\nabla_B \epsilon_2(n) = D_0(n) \left[\sum_{k=1}^{2N} D_k(n) \nabla_B^k \epsilon_2(n-k) - E_1(n) \right] \quad (43)$$

where the superscript k on the gradient operator implies differentiation with respect to weights evaluated at time $(n-k)$. The constants $C_0(n)$ and $D_0(n)$ are given by

$$C_0(n) = D_0(n) = \frac{1}{1 - w_0^A(n) w_0^B(n)}. \quad (44)$$

As in the standard adaptive noise canceller, the rate of convergence of the filter weights for the CTRANC is dependent on the input signal powers $\epsilon_1^2(n)$ and $\epsilon_2^2(n)$. Accordingly, to preclude instability of the algorithm due to large step sizes, the step size is normalized by an estimate of the signal power [13]. Hence for filter A,

$$\mu_1(n) = \frac{a_1}{\sigma_1^2(n) + b_1} \quad (45)$$

where

$$\sigma_1^2(n) = (1-a)\sigma_1^2(n-1) + \alpha \epsilon_2^2(n) \quad (46)$$

provides an exponentially weighted time-average of the input signal power, and α, a_1, and b_1 are constants. For filter B, $\mu_2(n)$ is calculated in a similar manner with respect to input signal power $\epsilon_1^2(n)$.

B. Lattice Filter Based Formulation

Lattice joint process estimators can be utilized in the CTRANC, as shown in Fig. 5. The lattice structures serve as predictors and therefore whitening filters. Hence the lattice is adapted in the same fashion as in the standard

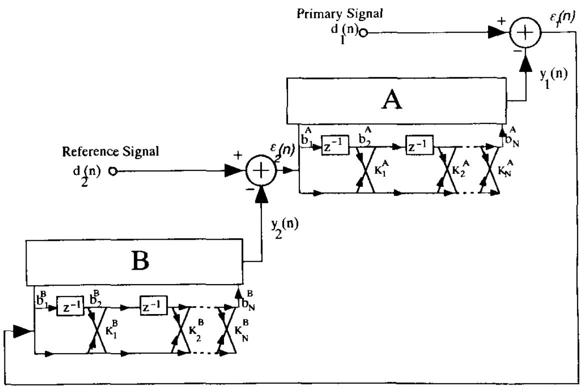


Fig. 5. Lattice-based CTRANC.

joint process estimation case where it is desired to maintain orthogonality of the backward prediction errors. The joint process coefficients are selected so as to achieve the desired transfer function or, equivalently, the desired joint process estimation.

The basic two-multiplier all-zero lattice is chosen for the orthogonalization process [14]. Blocks A and B represent transversal filters. Reflection coefficients K_M^A and K_M^B are chosen to minimize the sum of the mean-square forward and backward errors

$$E\left[f_{m+1}^{2^I}(n)\right] + E\left[b_{m+1}^{2^I}(n)\right] \quad I = A, B.$$

With the stochastic gradient algorithm utilized for minimization, the time and order update equations for the Nth-order lattice filter A are

$$b_1^A(n) = f_1^A(n) = \epsilon_2(n) \tag{47}$$

$$f_{m+1}^A(n) = f_m^A(n) - K_m^A(n) b_m^A(n-1) \tag{48}$$

$$b_{m+1}^A(n) = b_m^A(n-1) - K_m^A(n) f_m^A(n) \tag{49}$$

$$K_m^A(n+1) = K_m^A(n) + \frac{\left[f_{m+1}^A(n) b_m^A(n-1) + f_m^A(n) b_{m+1}^A(n)\right]}{d_m^{2^A}(n)} \qquad 1 \le m \le N-1 \tag{50}$$

where

$$d_m^{2^A}(n) = (1-\alpha)\, d_m^{2^A}(n-1) + f_m^{2^A}(n) + b_m^{2^A}(n-1). \tag{51}$$

The update equations for the Mth-order lattice filter B are identical to the above, with the superscripts changed from A and B and $\epsilon_2(n)$ replaced by $\epsilon_1(n)$.

The joint process estimator coefficients are determined by minimizing mean-square errors $E[\epsilon_1^2(n)]$ and $E[\epsilon_2^2(n)]$ through the stochastic gradient approximation. The transversal filter coefficients are updated as

$$w_k^A(n+1) = w_k^A(n) - 2\mu_k^A(n)\epsilon_1(n)\frac{\partial \epsilon_1(n)}{\partial w_k^A(n)} \qquad k = 1, \cdots N \tag{52}$$

$$w_l^B(n+1) = w_l^B(n) - 2\mu_l^B(n)\epsilon_2(n)\frac{\partial \epsilon_2(n)}{\partial w_l^B(n)} \qquad l = 1, \cdots M \tag{53}$$

where

$$\mu_i(n) = \frac{1}{\sigma_i^2(n)} \tag{54}$$

and

$$\sigma_i^2(n+1) = (1-\beta)\sigma_i^2(n) + \beta b_i^2(n) \quad i = k, l$$

for each of the filters A and B.

The major effort in the formulation lies in the determination of a recursive formula for the two gradients

$$\nabla_A^T \epsilon_1(n) = \left[\frac{\partial \epsilon_1(n)}{\partial w_1^A(n)} \frac{\partial \epsilon_1(n)}{\partial w_2^A(n)} \cdots \frac{\partial \epsilon_1(n)}{\partial w_N^A(n)}\right] \tag{55}$$

$$\nabla_B^T \epsilon_2(n) = \left[\frac{\partial \epsilon_2(n)}{\partial w_1^B(n)} \frac{\partial \epsilon_2(n)}{\partial w_2^B(n)} \cdots \frac{\partial \epsilon_2(n)}{\partial w_M^B(n)}\right]. \tag{56}$$

For transversal filter A we have

$$\epsilon_1(n) = d_1(n) - w_A^T(n) b^A(n)$$

where

$$b^{A^T}(n) = \left[b_1^A(n) b_2^A(n) \cdots b_N^A(n)\right]$$

and

$$w_A^T(n) = \left[w_1^A(n) w_2^A(n) \cdots w_N^A(n)\right].$$

It is easily seen that

$$\nabla_A \epsilon_1(n) = -b^A(n) - \sum_{i=1}^{N} w_i^A(n) \nabla_A b_i(n). \tag{57}$$

Now for lattice filter A we can write recursions for the backward prediction errors from (48) as

$$b^A(n) = K^A(n) b^A(n-1) + k^A(n)\epsilon_2(n) \tag{58}$$

where

$$K^A(n) = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots\cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & \cdots\cdots & 0 & 0 \\ K_1^A(n)K_2^A(n) & 1 & 0 & 0 & \cdots\cdots & 0 & 0 \\ K_1^A(n)K_3^A(n) & K_2^A(n)K_3^A(n) & 1 & 0 & \cdots\cdots & 0 & 0 \\ \vdots & & & & & & \\ K_1^A(n)K_{N-1}^A(n) & K_2^A(n)K_{N-1}^A(n) & & & \cdots\cdots & 1 & 0 \end{bmatrix} \tag{59}$$

and

$$k^{A^T}(n) = \left[1 - K_1^A(n) - K_2^A(n) \cdots -K_{N-1}^A(n)\right]. \quad (60)$$

Substituting (58) in (57), we get, after some calculations,

$$\nabla_A \epsilon_1(n) = -b^A(n) - \sum_{i=1}^{N-1} A(i)\nabla_A b_i^A(n-1) - A(N)\nabla_A \epsilon_2(n) \quad (61)$$

where

$$A(i) = w_{i+1}^A(n) + \sum_{k=i+1}^{N-1} w_{k+1}^A(n)K_i^A(n)K_k^A(n) \qquad 1 \le i \le N-1 \quad (62)$$

and

$$A(N) = w_1^A(n) - \sum_{i=2}^{N} w_i^A(n)K_{i-1}^A(n).$$

For transversal filter B we have

$$\epsilon_2(n) = d_2(n) - w_B^T(n)b^B(n) \quad (63)$$

where

$$w_B^T(n) = \left[w_1^B(n)w_2^B(n) \cdots\cdots\cdots w_M^B(n)\right]$$

$$b^B(n) = \left[b_1^B(n)b_2^B(n) \cdots\cdots\cdots b_M^B(n)\right].$$

Hence,

$$\nabla_A \epsilon_2(n) = \sum_{i=1}^{M} w_i^B \nabla_A b_i^B(n). \quad (64)$$

Now as in (58), recursions for the backward prediction errors for lattice filter B can be written as

$$b^B(n) = K^B(n)b^B(n-1) + k^B(n)\epsilon_1(n) \quad (65)$$

where

$$K^B(n) = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots\cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & \cdots\cdots & 0 & 0 \\ K_1^B(n)K_2^B(n) & 1 & 0 & 0 & \cdots\cdots & 0 & 0 \\ K_1^B(n)K_3^B(n) & K_2^B(n)K_3^B(n) & 1 & 0 & \cdots\cdots & 0 & 0 \\ \vdots & & & & & & \\ K_1^B(n)K_{M-1}^B & K_2^B(n)K_{M-1}^B(n) & & \cdots & \cdots\cdots & 1 & 0 \end{bmatrix}$$

and

$$k^{B^T}(n) = \left[1 - K_1^B(n) - K_2^B(n) \cdots\cdots -K_{M-1}^B(n)\right].$$

Therefore, the required $\nabla_A \epsilon_2(n)$ in (64) may be evaluated as follows. Substituting for $b_i^B(n)$ from (65), we get, after some calculations,

$$\nabla_A \epsilon_2(n) = -\sum_{i=1}^{M-1} B(i)\nabla_A b_i^B(n-1) - B(M)\nabla_A \epsilon_1(n)$$

where

$$B(i) = w_{i+1}^B(n) + \sum_{k=i+1}^{M-1} w_{k+1}^B(n)K_i^B(n)K_i^B(n) \qquad 1 \le i \le M-1 \quad (66)$$

and

$$B(M) = w_1^B(n) - \sum_{i=2}^{M} w_i^B(n)K_{i-1}^B(n).$$

Substituting for $\nabla^A \epsilon_2(n)$ in (61), we finally get

$$\nabla_A \epsilon_1(n) = \frac{1}{[1 - A(N)B(M)]} \cdot \left[-\sum_{i=1}^{N-1} A(i)\nabla_A^k b_i^A(n-1) + A(N)\sum_{i=1}^{M-1} B(i)\nabla_A^k b_i^B(n-1) - b^A(n)\right]. \quad (67)$$

Equations (66) and (67) give the recursive expressions for the gradients of the two error functions with respect to the weights in the A filter. The gradients of $b(n)$ are determined recursively using (58). As in the transversal-based CTRANC algorithm, the superscript k on the gradient operator implies differentiation with respect to weights evaluated at the same time $(n-k)$ as the evaluation of signals $b(m)$, and is justified by a similar orthogonal error argument.

The dual equations for the B filter are

$$\nabla_B \epsilon_2(n) = \frac{1}{[1 - A(N)B(M)]} \cdot \left[-\sum_{i=1}^{M-1} B(i)\nabla_B^k b_i^B(n-1) + B(M)\sum_{i=1}^{N-1} A(i)\nabla_B^k b_i^A(n-1) - b^B(n)\right] \quad (68)$$

and

$$\nabla_B \epsilon_1(n) = -\sum_{i=1}^{N-1} A(i)\nabla_B b_i^A(n-1) - A(N)\nabla_B \epsilon_2(n). \quad (69)$$

Equations (47)–(51) and dual equations for lattice B, (52) and (53) for the two transversal filters, and (67) and (68)

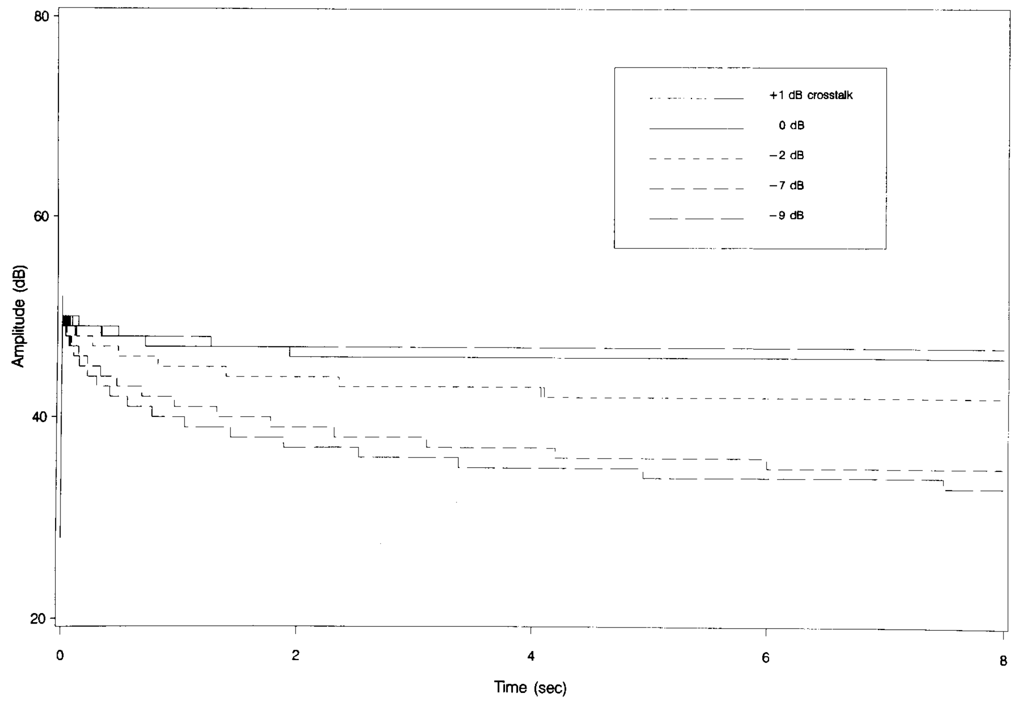


Fig. 6. Mean-square error for transversal-based CTRANC.

constitute the time and order update equations for the lattice based CTRANC.

IV. Simulation Results

The primary signal and reference signal containing crosstalk, applied earlier to ANC, were applied to the transversal-based CTRANC and lattice-based CTRANC. The transversal filters A and B, were in both cases set to order 4. Figs. 6 and 7 show that the mean-square errors obtained, ranged from 3 dB to 11 dB lower than those obtained with the standard adaptive noise canceller. All signals consisted of 1000 frames, with 64 points per frame.

A typical output is shown in Fig. 8, where the output signals for -9-dB crosstalk, from ANC and (transversal-based) CTRANC are compared with the original signal. All signals are the two frames 500 and 501. It is seen that the output signal of CTRANC is a close replica of the original signal. It should be observed that in the primary signal $s_1(n) + n_1(n)$, $n_1(n)$ is a delayed version of $n_2(n)$. Likewise, in the reference signal $s_2(n) + n_2(n)$, $s_2(n)$ is now a delayed version of $s_1(n)$. Under these circumstances, the finite impulse response filters A and B serve to estimate the delays and provide as outputs replicas of $s_1(n)$ and $n_2(n)$, respectively. At convergence, the output $e_1(n)$ should be a close replica of $s_1(n)$. Similarly, the error signal $e_2(n)$ of filter B should be a close approximation to the noise signal $n_2(n)$. This is so for the above examples. For the -9-dB crosstalk case, the output $\epsilon_2(n)$ of the transversal-based CTRANC is shown with the noise signal $n_2(n)$ in Fig. 9. It is seen that the lower adaptive filter configuration containing filter B removes the signal from its input to provide a good estimate of the noise $n_2(n)$. Equivalently, were the primary signal $s(n) + n(n)$ to consist of a delayed version of the desired signal filtered through filter $P_1(z^{-1})$ and the reference signal $s(n) + n(n)$ to consist of a delayed version of noise $n_2(n)$ filtered through $P_1(z^{-2})$ then the output $\epsilon_1(n)$ of filter A would, at convergence, provide a good estimate of the noise.

A second example utilized 800 ms of speech and white noise. The speech and noise were filtered in the same manner as in the previous example, and the delayed versions added to the primary and reference signals. The convergence of the filter weights for the transversal-based CTRANC is depicted in Figs. 10 and 11. The converged values of the mean-square errors for the transversal and lattice-based CTRANCs differed by no more than 1 dB. All results were averaged over 50 samples. Initial conditions for filter A weights were set to zero while those for filter B were set to 0.2, -0.1, -0.1, and 0.0. Initial conditions for the lattice step sizes were set to $1/2000$ (α/P), which was the asymptotic value of P, the average power $[b^2(n) + f^2(n)]$ that was being minimized with $\alpha = 0.001$. Similarly, the transversal filter step size initial condition was set at $1/1000$.

In the simulations attempted there were cases where the mean-square error increased sharply before settling to steady-state. Updates to the filter weights are dependent on the gradient, which for the transversal-based CTRANC is obtained through recursions (42) and (43) which have a

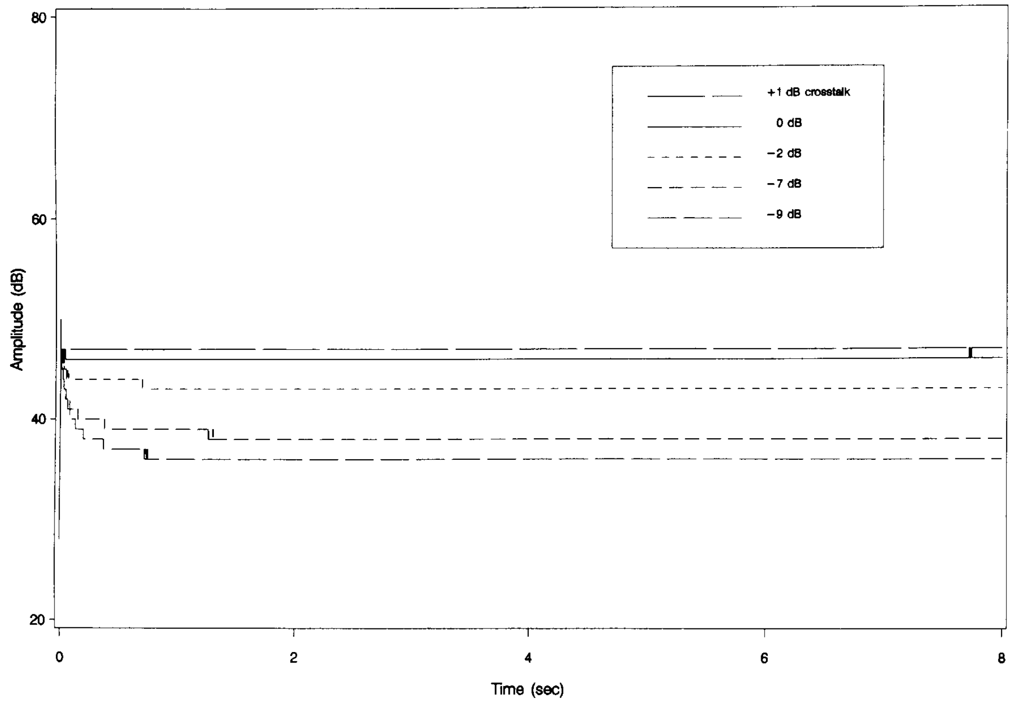


Fig. 7. Mean-square error for lattice-based CTRANC.

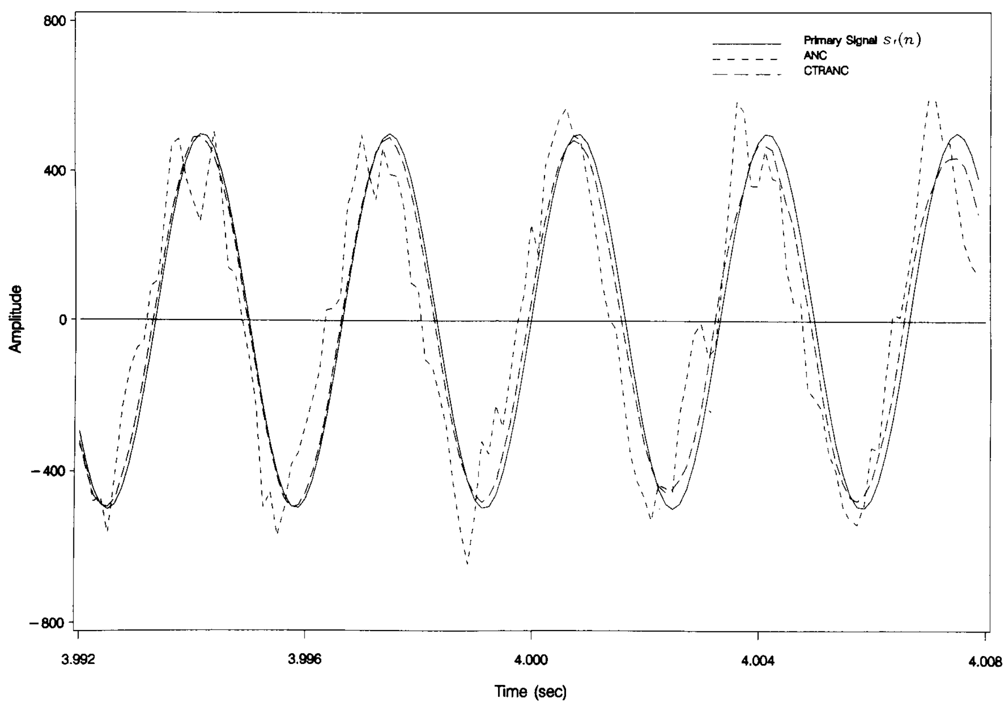


Fig. 8. CTRANC output $\epsilon_1(n)$ and ANC output for −9-dB crosstalk.

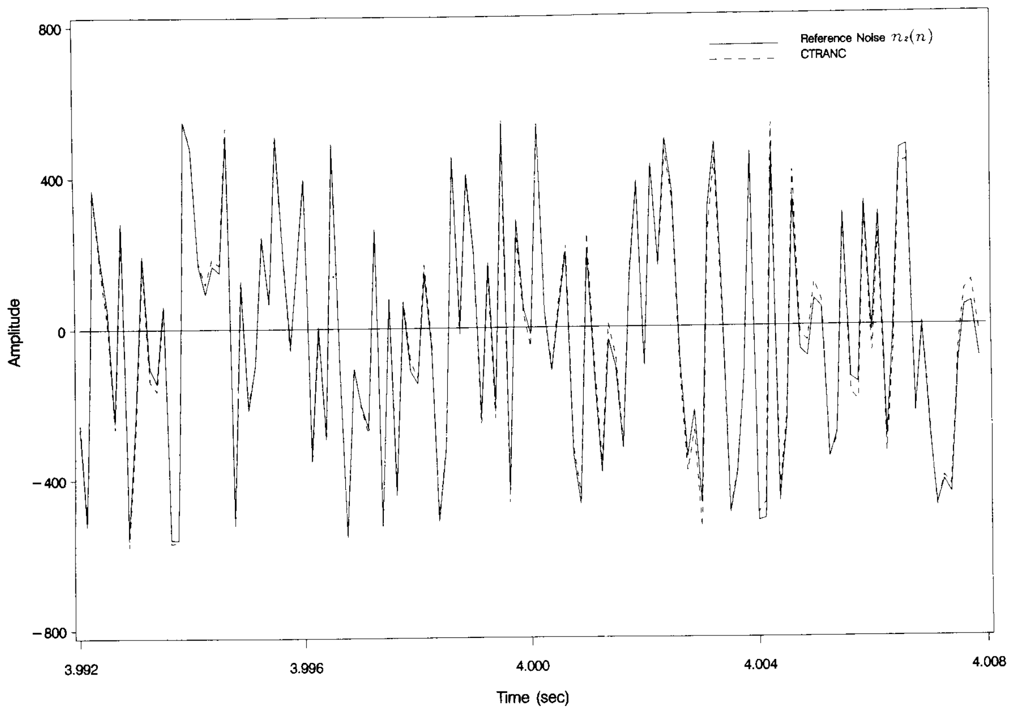


Fig. 9. CTRANC output $\epsilon_2(n)$ for -9-dB crosstalk.

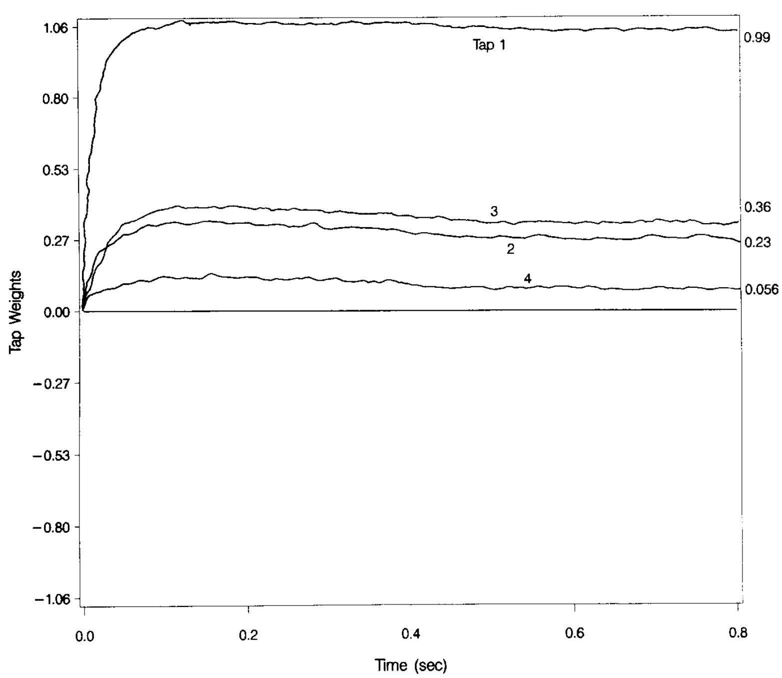


Fig. 10. Convergence of filter A tap weights.

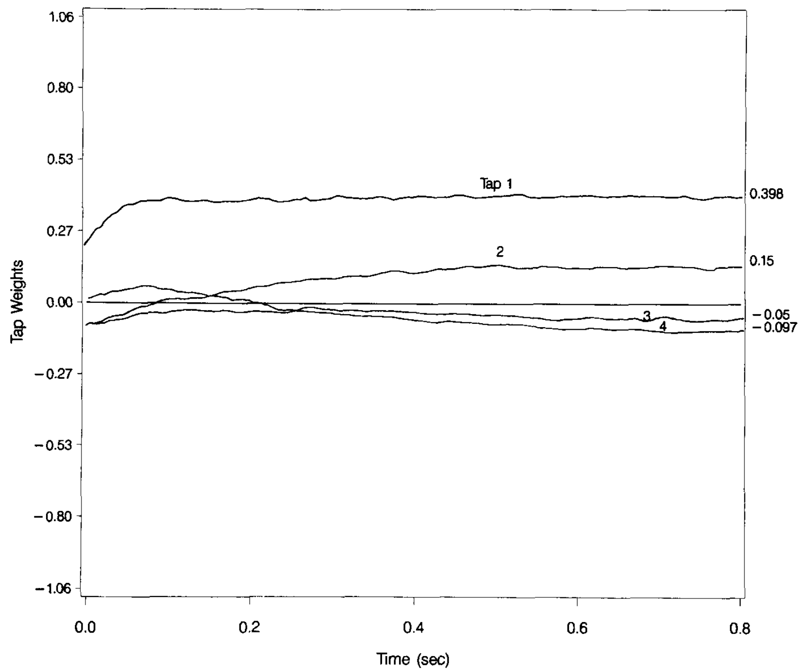


Fig. 11. Convergence of filter B tap weights.

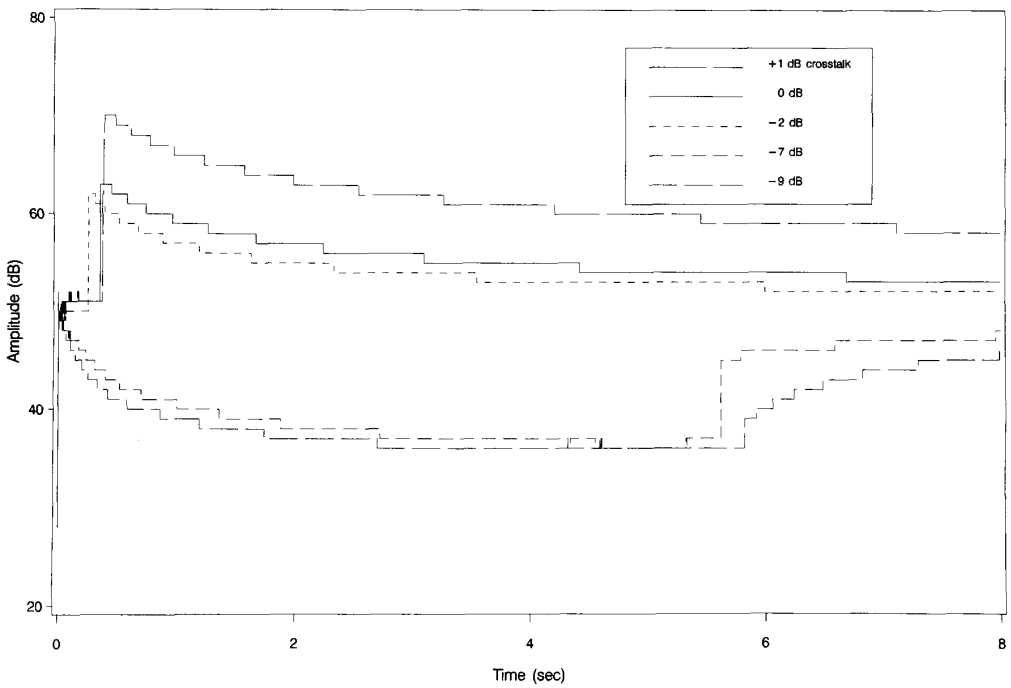


Fig. 12. Mean-square error for transversal-based CTRANC tenth-order filters. Recursion history of 20.

memory of twice the filter length. Accordingly, there is potential for instability when poles of the recursion lie outside the unit circle. Simulations have shown that the problem can be alleviated by restricting the memory of the recursion. Then the filter weights are updated through a shorter time history of the past. As an illustration of instability, the first example was simulated with tenth-order filters and a recursion history of 20 for the gradients. As seen in Fig. 12 the mean-square errors increased before settling to values that were 3 to 14 dB greater than those obtained with earlier versions of CTRANC and even ANC. Reducing the recursion memory to 2 with the same filter order of ten gave mean-square errors comparable to those obtained earlier with filter order of four and recursion memory of 4. Other sources of instability were due to the normalization procedures for the step sizes. Choosing a value for the weighting $(1 - \alpha)$ (46) close to 1 often provided good results. In many cases choosing the offsets b_1 and b_2 (45) proved to be critical, with small values often giving better results.

V. Conclusion

This paper has presented a novel structure for joint process estimation for noise canceling, when the reference signal is contaminated with crosstalk. For 1- to -9-dB levels of crosstalk, simulations with the transversal and the lattice-based CTRANCs showed 3 to 11-dB improvement in the mean-square error over that for the standard adaptive noise canceller. When the reference channel had filtered signal and filtered noise, the two adaptive filter weights converged to the desired weights.

Acknowledgment

The authors wish to thank C. M. Puckette for providing the resources and the open environment in which to carry out this work. Seth Silverstein was a participant in the original development of the new structure. Our colleague, L. Kathy Bechtel, provided valuable programming assistance. Thanks are also due to anonymous reviewers for suggestions for improving the paper.

References

[1] G. S. Kang and L. J. Fransen, "Experimentation with an adaptive-noise cancellation filter," *IEEE Trans. Circuits Syst.*, Special Issue on Adaptive Systems and Applications, vol. CAS-34, pp. 753–757, July 1987.

[2] C. F. Teacher, "Performance of LPC vocoders in a noisy environment," in *Proc. 1979 IEEE ICASSP*, pp. 216–291, Apr. 1979.

[3] J. S. Lim, Ed., *Speech Enhancement*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[4] S. F. Boll, "Noise suppression techniques for digital signal processing," in *Conf. Record Towards Robustness in Speech Recognition*, Voice Control Systems, Inc., Santa Barbara, CA, Nov. 1983.

[5] S. F. Boll and D. C. Pulsipher, "Suppression of acoustic noise in speech using two microphone adaptive noise cancellation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 752–753, Dec. 1980.

[6] B. A. Hedstrom, R. L. Kirlin, and P. F. Driessen, "Adaptive cancellation of adjacent channel interference," in *Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing*, pp. 577–579, Dec. 1989.

[7] R. L. Zinser, G. Mirchandani, and J. B. Evans, "Some experimental and theoretical results using a new adaptive filter structure for noise cancellation in the presence of crosstalk," in *Proc. 1985 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 1253–1256, Mar. 1985.

[8] G. Mirchandani, R. C. Gaus, Jr., and Kathy L. Bechtel, "Performance characteristics of a hardware implementation of the crosstalk resistant adaptive noise canceller," in *Proc. 1986 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 93–96, Apr. 1986.

[9] C. W. K. Gritton and D. W. Lin, "Echo cancellation algorithms," *IEEE ASSP Mag.*, pp. 30–38, Apr. 1984.

[10] B. Widrow *et al.*, "Adaptive noise canceling: Principles and applications," *Proc. IEEE* vol. 63, pp. 1692–1716, Dec. 1975.

[11] R. L. Zinser, private communication.

[12] R. L. Zinser, S. D. Silverstein, and S. R. Koch, "Two input crosstalk-resistant adaptive noise canceller," U.S. Patent 4 649 505, Mar. 1987.

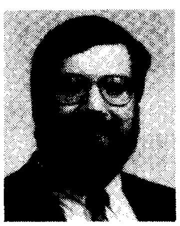
[13] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*. Boston, MA: Kluwer Academy Publishers, 1984.

[14] F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," in *Proc. 7th Int. Cong. Acoust.*, pp. 261–264, 1971.

Gagan Mirchandani (M'68) received the B.S. degree from Worcester Polytechnic Institute, the M.S. degree from Syracuse University, and the Ph.D. degree from Cornell University, all in electrical engineering, in 1958, 1960, and 1968, respectively.

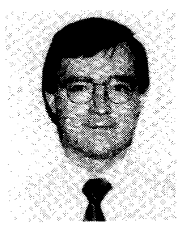
He is a Professor of computer science and electrical engineering at the University of Vermont, where he has been since 1968 and where he heads the Digital Signal Processing Group. His research interests are in both theoretical and implementation issues of digital signal processing. The current research activities of the group include image coding and compression techniques and parallel implementation of digital signal processing algorithms. He has lectured widely and has been a consultant to a number of industrial concerns in algorithm development and hardware implementation.

In 1977, Dr. Mirchandani was co-recipient of the Best Paper Award in the IEEE Transactions on Education.



Richard L. Zinser (S'80–M'85) received the B.S., M.E., and Ph.D. degrees in electrical engineering from Rensselaer Polytechnic Institute in 1981, 1982, and 1987, respectively.

In 1982, he joined the General Electric Research and Development Center in Schenectady, NY. His research activities there include speech and image coding, adaptive filtering, combined source and channel coding for mobile radio, and real time implementation methodologies. Presently, he manages a group performing research in image and speech processing.



Joseph B. Evans (S'82–M'89) received the B.S.E.E. degree from Lafayette College in 1983, and the M.S.E., M.A., and Ph.D. degrees from Princeton University in 1984, 1986, and 1989, respectively.

Since 1989 he has been an Assistant Professor of electrical and computer engineering at the University of Kansas. Prior to this, he held a postdoctoral position in the Network Systems Research Department at AT & T Bell Laboratories in Holmdel, NJ, where he was involved in the design of a high performance integrated network. While at Princeton, he was awarded an AT & T Bell Laboratories Graduate Fellowship for 1984–1988; during this period, he was also a part-time employee of Bell Labs, working in the field of speech processing algorithms for packet networks. His current research interests include adaptive signal processing, digital filtering, digital speech processing, special-purpose computer architecture, high-speed (gigabit) networks, and VLSI implementations.