Acoustic Echo Cancellation for Full-Duplex Voice Transmission on Fading Channels

Sangil Park and Dion D. Messer

Motorola Inc.
Digital Signal Processing Operations
Austin, TX 78735

ABSTRACT

This paper discusses the implementation of an adaptive acoustic echo canceler for a hands-free cellular phone operating on a fading channel. The adaptive lattice structure, which is particularly known for faster convergence relative to the conventional tapped-delay-line (TDL) structure, is used in the initialization stage. After convergence, the lattice coefficients are converted into the coefficients for the TDL structure which can accommodate a larger number of taps in real-time operation due to of its computational simplicity. The conversion method of the TDL coefficients from the lattice coefficients is derived and the DSP56001 assembly code for the lattice and TDL structures is included, as well as simulation results and the schematic diagram for the hardware implementation.

1.0 Introduction

Adaptive signal processing for echo cancellation structures has a variety of usages in *telecommunication* applications due to *multi-path* and *impedance mismatches* in communication channels. Echo cancellation is required especially for full-duplex voice transmission where the microphones and speakers are located in places such that an acoustic echo is created. One such application is a *hands-free cellular phone* which allows full duplex operation by preventing the phone from breaking into oscillations [1]. The ability to provide hands-free operation of cellular (mobile) phones offers users a safer and more convenient way to use their cellular phones while driving a car as shown in Figure 1.

In the cellular phone application, there needs to be two echo cancellers in the system, one to cancel the phone line (electrical) echo and the other to cancel the acoustic echo, which is the signal from the loudspeaker echoed back into the microphone. In this paper only the acoustic cancellation problem is considered. Figure 2 shows the model and the hardware schematic of the acoustic echo canceller. The adaptive algorithm shown as the adaptive digital filter (ADF) block in Figure 2 minimizes the error signal which is the difference between the actual transmitted signal and the estimated transmitted signal by the linear combination of the received data set. When the error terms are minimized the adaptive filter impulse response is said to have converged to the impulse response of the echo paths.

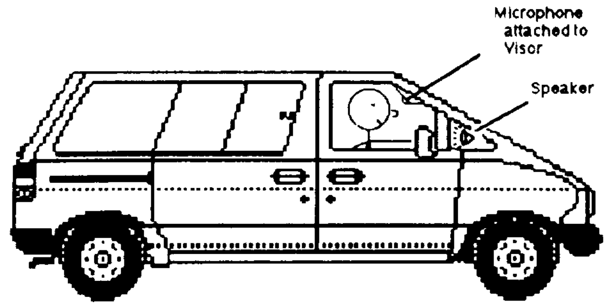

Figure 1 Depiction of a proposed hands-free cellular phone system

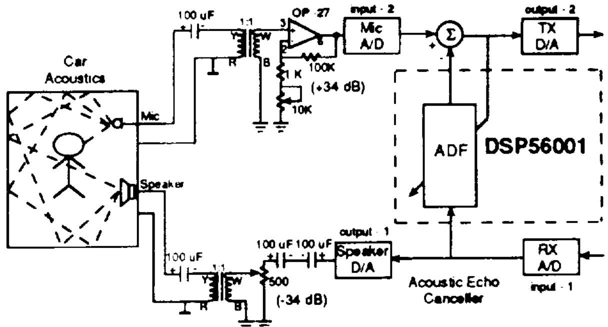

Figure 2 Block diagram of acoustic echo canceller

An implementation of the acoustic echo canceller for a *speakerphone* application to improve performance was introduced by one of the authors [2]. The previous paper uses the DSP56200 cascadable adaptive FIR filter peripheral which implements the conventional TDL structure with the least-mean-square (LMS) algorithm for adaptation [3]. An acoustic echo canceller needs an initialization period (training) before the phone can work properly for full-duplex operation. This may require up to 5 seconds of initialization time depending on the convergence property of the adaptive algorithm. Although the TDL structure [3,4] is a simple and commonly used stochastic approximation-type algorithm, the convergence time is slow, especially if the training signal is narrowband or has band-limited spectral content. Thus, pseudo-random noise that has a broad frequency spectrum is normally used for initialization. However, this type of random noise creates problems to the user who will be able to hear it during the initialization. The user will most likely interpret this noise to be a bad (or no) connection and will hang up before initialization is complete. In adaptive filtering applications such as hands-free cellular phone, therefore, very rapid convergence of the adaptive coefficients is a requirement.

The lattice structure based on the adaptive LMS algorithm has been widely accepted for applications where rapid transient adaptation is required and/or the eigenvalues of the input signal are highly disparate [5,6]. The lattice structure can be interpreted to be self-orthogonalizing which has been shown to speed up convergence. Cellular phones are normally used inside a car which has smaller acoustic reverberation and echo paths compared to the size of a office or conference room in which a conventional speakerphone must function. Thus, a fewer number of taps, which represents the time-window for the adaptive approximation, than the conventional TDL structure can be used for the adaptive process. However, due to the computational complexity of the lattice-LMS algorithm it is hard to apply a large number of coefficients (stages) to accommodate 0.1 second (which is a time window of 800 taps) of acoustic echo delay in real-time. Thus, the lattice structure is used only for the initialization stage and the coefficients are converted to the TDL structure. The TDL structure is computationally efficient and can accommodate larger number of coefficients in real-time to cancel the long delayed echoes.

2.0 Acoustic-Echo Canceller Model

Depending on the characteristics of the car's internal acoustics, the echo may be sufficiently strong, such that this echo must be removed at the microphone input. The term used to describe the amount of echo which can be removed by the echo canceller is Echo Return Loss Enhancement (ERLE) and can be defined as [2]:

$$\mathbf{ERLE}\,(dB) = 10\log_{10}\left[\frac{\mathbf{E}\,[y(k)^2]}{\mathbf{E}\,[e(k)^2]}\right] \qquad (1)$$

where $\mathrm{E}\,[y(k)^2]$ and $\mathrm{E}\,[e(k)^2]$ are the expected values of microphone input signal power and uncanceled echo signal power, respectively, as shown in Figure 2. The desired maximum amount A goal for ERLE is 30 dB due to ambient noise which is not created by the echo itself [3].

Due to advances in CMOS process technology, inexpensive adaptive digital filters are readily available. The DSP56001 can run upto 830 taps of a TDL-LMS adaptive filter at 8 kHz samples per second with 24-bit data and coefficients. As is shown in the following section, ERLE is a function of many parameters including the number of taps and the precision of the coefficients.

3.0 Echo Cancellation Algorithms

In this section, two adaptive algorithms are described with particular emphasis on echo cancellation applications.

3.1 Adaptive TDL-LMS algorithm

Figure 3 shows a block diagram of the adaptive echo canceler model which uses a TDL structure to provide adaptive coefficient adjustment. If $h_i(k)$ are the filter coefficients, $R_{xx}(k)$ is the auto-correlation matrix of the received line signal *x(k)* at time *k*, and $R_{xy}(k)$ is the cross-correlation vector between the received signal *x(k)* and the echo signal *y(k)*, then the optimum filter coefficient vector that minimizes the expected value of $e^2(k)$ in Figure 3 is given by [3]

$$\hat{H}(k) = R_{xx}^{-1}(k)\,R_{xy}(k) \qquad (2)$$

where *H (k)* is an N-element vector consisting of the filter coefficients at time *k* as

$$H(k) = \left[h_0(k)\;\; h_1(k)\;\; \ldots\ldots\;\; h_{N-1}(k)\right]^T \qquad (3)$$

and *T* denotes matrix transpose. The coefficients $h_i(k)$ are updated to minimize the error signal (residual echo), *e(k)*, which is the transmitting line signal from the echo canceler. *e(k)* can be expressed as

$$e(k) = y(k) - H^T(k)\,X(k) \qquad (4)$$

where *X(k)* is the input data vector given by

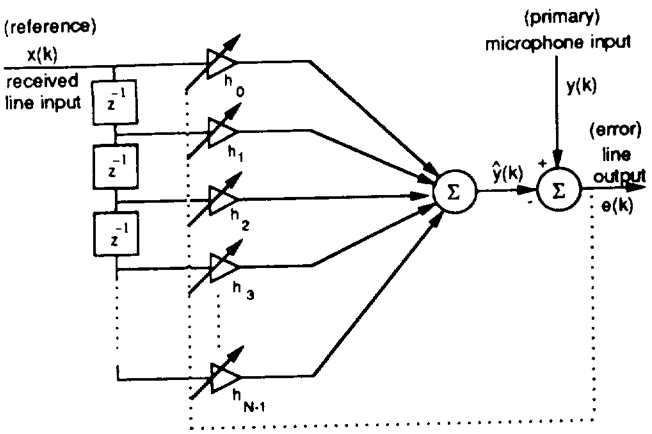

Figure 3 The tapped-delay-line (TDL) structure

$$X^T(k) = [x(k)\ x(k-1) \ldots\ldots x(k-N+1)] \quad (5)$$

The LMS algorithm, which is one implementation of the steepest decent method, updates the weight vector, $H(k)$, at each k via the following relation

$$H(k) = H(k-1) + \mu e(k) X(k) \quad (6)$$

where μ denotes the loop-gain factor (convergence parameter). The adaptive algorithm forces the error term toward zero. When the error terms are minimized, the adaptive filter impulse response is said to have converged to the impulse response of the echo path.

Convergence properties and stability aspects of the LMS algorithm have been well documented [4,7]. The general conditions in practice for the loop-gain factor is

$$0 < \mu < \frac{1}{tr[R_{xx}]} \quad (7)$$

where $tr[R_{xx}]$ denotes the trace of R_{xx}. The optimized DSP56001 assembly code for the TDL-LMS algorithm in (6) can be written as[8]:

```
clr   a       x0,x:(r0)+y:(r4)+,y0 ;clear a,x0=x(n)
move  x:(r0)+,x1y:(r4)+,y0         ;x1=x(n-1),y0=h(0)
do    #N/2,lms                     ;do N/2 times
mac   x0,y0,a  y0,b      b,y:(r5)+ ;a=h(0)*x(n),b=h(0)
macr  x1,y1,b  x:(r0)+,x0y:(r4)+,y0 ;b=h(0)+e*x(n-1)
                                   ;x0=x(n-2),y0=h(1)
mac   x1,y0,a  y0,b   b,y:(r5)+ ;a=a+h(1)x(n-1),b=h(1)
macr  x0,y1,b  x:(r0)+,x1y:(r4)+,y0 ;b=h(0)+e*x(n-1)
                                   ;x0=x(n-3),y0=h(1)
lms
 move b,y:(r5)+                    ;save new coeffs.
 move (r0)-n0                      ;pointer update
```

where **r0** is the register pointing to the input buffer which is modulo-addressed to accommodate 768 (N) current data points. **R4** and **r5** registers are pointing to the even and odd numbered current adaptive coefficients locations, respectively. The Modifier Registers, **m0**, **m4** and **m5** are set to be 767 (N-1), 383 (N/2-1) and 383 (N/2-1), respectively. This TDL-LMS algorithm requires only $2N+2$ instruction cycles per sample period. When 768 taps are used for an acoustic echo-canceler the processing requirement at 8 kHz of sampling rate is 12.3 million instructions per second (MIPS).

3.2 Adaptive Lattice-LMS Algorithm

The lattice predictor (often called as one-step predictor) structure was originally proposed by Itakura and Saito [9] for speech analysis. The one-step predictor has also been extended to a noise-canceler configuration as shown in Figure 4 [5]. If the inputs $x(k)$ and $p(k)$ are stationary, then it can be shown that the respective steady-state values of $e^2(k)$ and $v_N^2(k)$ for TDL and lattice models are the same. The filter model in Figure 4 consists of 3 stages which can be extended to M stages for mathematical analysis purpose. Its upper half (solid lines) is simply the (one-step) predictor model [6]. The lower portion (dashed lines) consists of M additional coefficients, v'_l, $1 \le l \le M$. The basic idea involved in obtaining an adaptive algorithm is to continuously adjust the lattice weights $v_l(k)$ and $v'_l(k)$. The $v_l(k)$ are adjusted to minimize the instantaneous error $e_l^2(k) + w_l^2(k)$ via the one-step predictor LMS algorithm as

$$v_l(k+1) = v_l(k) + (1-\beta)[e_l(k) w_{l-1}(k) + w_l(k) e_{l-1}(k)] \quad (8)$$

for $1 \le l \le M$, which we refer to as the lattice-LMS equation for a one-step predictor. In practice, a convenience choice for a β is $\beta = 1 - \mu$ in (7).

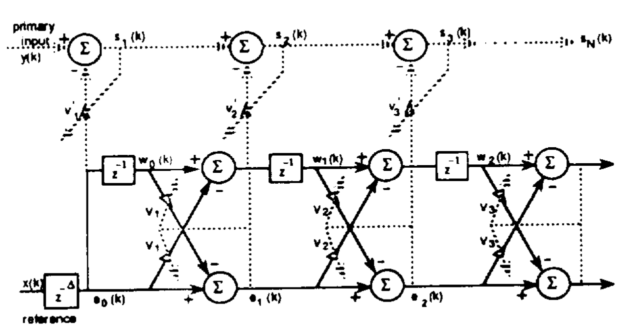

Figure 4 Lattice structure for noise-canceller

The lattice filtering computation at each stage gives successive orthogonalization process. Thus, the successive coefficients can be optimized independent of coefficients at other stages. As a result of this orthogonalization, the convergence rate of the lattice-LMS algorithm is not restricted by the eigenvalue structure of the input signal, which is not the case for the TDL-LMS algorithm.

Next, another set of coefficients, $v_l'(k)$ in Figure 4 provides the noise-cancelling substraction paths. The individual coefficients are adjusted to minimize the filter error $s_l^2(k)$ using a technique similar to (6). Thus we have [10]

$$v_l'(k+1) = v_l'(k) - \mu\left[\frac{\partial s_l^2(k)}{\partial v_l'(k)}\right], \quad 1 \le l \le M \qquad (9)$$

where μ is a convergence parameter. Again from Figure 4 it follows that

$$s_l(k) = s_{l-1}(k) - v_l'(k)\, w_{l-1}(k+1) \qquad (10)$$

with $s_0(k) = p(k)$. Substitution of (10) into (9) leads to

$$v_l'(k+1) = v_l'(k) - 2\mu s_l(k)\, w_{l-1}(k+1) \qquad (11)$$

for $1 \le l \le M$, The DSP56001 assembly code for the lattice-LMS adaptive filter algorithm in (9)-(11) can be written as:

```
dofilt
   move  y:xdatain,b          ; read in x-input
   move  y:ydatain,a          ; read in y-input
   move  b,x:(r0)+ a,y:(r6)   ; put input in array
   move  b,x:(r2)             ; move input to memory
   do    #order,endloop
   move  b,y:(r7)             ; store previous err. in memory
   move  y:(r5),y0            ; put v' in y0 for calculation
   move  x:(r2),x1 y:(r6),a   ; put w_n in x1,s state into a
   macr  -y0,x1,a y:(r4),y0   ; a=s_n=s_n-1-v'w_n(n+1),v in x0
   move  x:(r0),a  a,y:(r6)   ; put w_n in a, store s_n
   move  b,y1                 ; move error into y get e_n-1
   macr  -y0,y1,a a,x0        ; a=w_n-v*e_n=w_n+1, w_n into x0
   macr  -x0,y0,b a,x:(r0)+   ; b=e_n+1=e_n-v*w_n,store w_n+1
   move  a,x:(r2)             ; store wn+1
   move  x:(r3),x1            ; move 2*µ into x1
   mpy   x1,x0,a  y:(r6),y1   ; a=2*µ*w_n, s state into y1
   move  a,x1     y:(r5),a    ; move a into x1, k' into a
   macr  x1,y1,a  x:(r7+n7),y0 ; v'_n+1=v'+5*2*mu*w_n
   move  x:(r0),x1 a,y:(r5)+  ; move w_n+1 into x1, store k1
   mpy   x1,y0,a  b,x1        ; a=w_n+1*e_n-1(n-1), e into x1
   macr  x0,x1,a  x:(r3),x0   ;a=w_n-1e_n-1(n-1)+e_nw_n,2µ in x0
   move  a,x1     y:(r4),a    ; move a into x1, k into a
   macr  x0,x1,a  y:(r7)+,y0  ; a=v_n(new)=v_n(old)+2µ(a)
   move  a,y:(r4)+            ; store v_n(new)
endloop
   move  x:(r0)-,a            ; output from filter
   move  y:(r6),y0            ; output s_n
   move  a,y:filtout
   move  y0,y:errout
   jmp   dofilt
```

where **r0** points to the stored filter coefficients, w_n. The buffer for **r0** is 65 ($M+1$) locations and is modulo addressed. The extra location is used because new values of the filter for the next time period are calculated before they are used in the present time period. The **r4** and **r5** registers point to v and v', respectively. Both are buffers of 64 (M) locations and are modulo addressed. The **r7** register points to the e_n values and is 128 ($2M$) locations to store two time periods of error values. The Modified Registers are used; **m0** is set to 64 (M); **m4** and **m5** are set to 63 ($M-1$); and **m7** is set to 127 ($2M-1$). This stage requires only 1280 instruction cycles per sample period, which yields a processing requirement of 10.25 MIPS at 8 kHz of sample rate.

4.0 Echo Characteristics of Car Interior

The acoustic path can be considered as a multi-reflection medium with an impulse response duration. Thus, the typical acoustics inside a car may have practically an infinite number of reflections which have different acoustical filtering effects with an exponentially decaying reverberation effect superimposed. In a typical car with reasonable acoustic treatment the reverberation time can be 0.1-0.15 seconds to reach the reverberation signal level decreased by 10 - 20 dB. However, when the car is moving, the background noise level due to the noises from engine and road may be high enough that the background noise can not be distinguished from uncancelled echoes due to an insufficient number of taps in the adaptive filters.

Echo characteristics can be measured by collecting reverberation and echo responses synchronized by an impulse output signal. The impulse can be generated in software and converted into an analog signal by a D/A converter followed by amplification to yield the audible impulse signal. Using a microphone, the residual analog signal can be converted to a digital signal by an A/D converter such as the DSP56ADC16 (16-bit Sigma-Delta A/D converter). Thus, an echo signal can be characterized by the impulse response of an acoustic chamber, or precisely, the paths from the loudspeaker to the microphone.

In this paper a simulated impulse response is used in order to characterize the convergence properties of both TDL and lattice structure. Figure 5 shows a simulated impulse response of a medium size car.

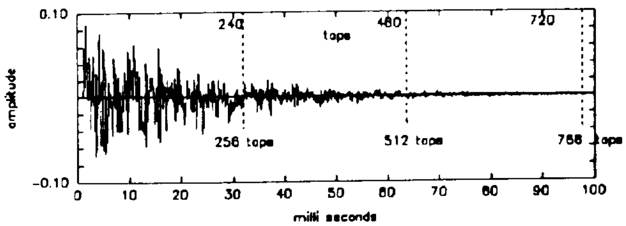

Figure 5 Impulse response of a simulated echo path

6.0 Simulation Results

A computer simulation was performed on a SUN-3/160 workstation, which modeled the system shown in Figure 2. It was assumed that the received signal was white Gaussian noise. The TDL and lattice algorithm in (7), (10) and (11) were used for the simulation. The variables in the simulation were μ, N, and M.

It has been shown that the convergence parameter μ, controls the convergence rate and the mean-square-error (MSE) of the adaptive system [4]. The constraints on the choice of μ are given by (7). The lattice-LMS and TDL-LMS algorithms are compared in terms of the MSE criteria which corresponds to the uncancelled echo. Figure 6 illustrates the MSEs of the algorithms when $\mu = 0.001$, $N = 768$ and $M = 64$. Note that the lattice-LMS algorithm converges much faster than the TDL-LMS algorithm. However, since the lattice structure used only 64 stages compared 768 taps for the TDL structure the uncancelled echo (error) is much larger than the counterpart.

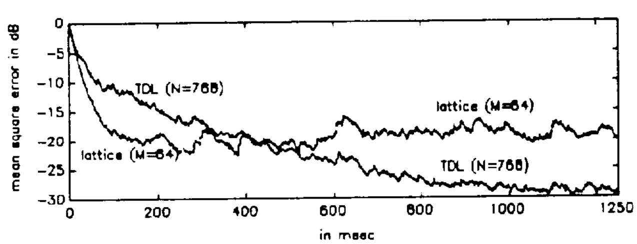

Figure 6 Mean-square errors for the lattice and TDL

After fast initialization using the lattice-LMS algorithm as shown in Figure 6, the coefficients are converted into TDL-LMS coefficients. Consider the equivalent M-TDL taps, defined in (3), from a set of M-lattice coefficients in (8) and (11). When $\hat{y}_M(k)$ is an estimate of $y(k)$ as shown in Figure 3, the corresponding error can be written as

$$s_M(k) = y(k) - \hat{y}(k) = y(k) - \sum_{i=0}^{M-1} h_{i,M}x(k-i) \quad (12)$$

where $h_{i,M}$ denotes the ith equivalent TDL tap when M is the total number of taps. Minimizing $e_N^2(k)$ with respect to the $h_{i,N}$ (new notation of h_i for the following derivation purpose), we can derive the following recursive algorithm to find a set of equivalent TDL taps using matrix bordering technique [11].

$$h_{i,L+1} = h_{i,L} + \nu'_{L+1}\alpha_{L+1-i,L} \quad ,i=0,1,\ldots L \quad (13)$$

where

$$\alpha_{i,L+1} = \alpha_{i,L} + \nu_{L+1}\alpha_{L+1-i,L} \quad ,i<L+1 \quad (14)$$

and $\alpha_{L+1,L+1} = -\nu_{L+1} \quad ,i=L+1$.

The recursion algorithm in (13) and (14) has to extend from $L=1$ through $L=M-1$ to find a set of M-TDL taps. The rest of the N-M coefficients in the TDL structure should also be initialized with zero before starting the adaptive process with the TDL-LMS algorithm.

Figure 7 shows the ERLE plots (defined in (1)). In order to smooth the output of adaptive filter the following smoothing functions were used [12].

$$E[y^2(k)] = \beta E[y^2(k-1)] + (1-\beta)y^2(k) \quad (15)$$

$$E[e^2(k)] = \beta E[e^2(k-1)] + (1-\beta)e^2(k) \quad (16)$$

where β=0.99 is the smoothing parameter. Note that the ERLE increases very rapidly at the initialization period. After the adaptive process is converted from lattice to TDL structure at $t=150$ *ms*, the ERLE increases slowly to the optimum solution. A total of 10,000 samples, corresponding to 1.25 seconds, were plotted to show the adaptive process when $\mu = 0.001$, $N = 768$ and $M = 64$.

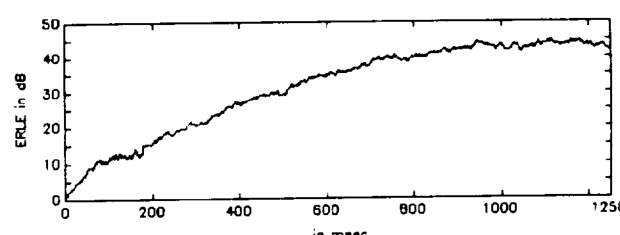

Figure 7 ERLE of the hybrid structure

7.0 Hardware Implementation Set-up

A block diagram of the hardware test implementation is shown in Figure 8. The SUN-3/160 workstation downloads assembled software into the DSP56001 Application Development System (ADS) which, in turn, controls the Ariel ADC56000 card. The ADS contains a DSP56001 general purpose digital signal processor chip which runs software in real time. The Ariel card has dual A/D and D/A converters, which convert the signals of the loudspeaker, microphone and the receive/transmit lines. This implementation allows real-time testing of the adaptive filter concepts discussed previously.

The DSP56001 is a Harvard Architecture digital signal processor which has separate program and data memories as well as buses. It currently executes one instruction in 75 *ns* which means a 768-tap TDL-LMS adaptive filter can be performed in only 115.2 μs. This 13.5 MIPS rating is somewhat deceiving because, due to the dual buses and memories, more than one operation occurs in each instruction cycle.

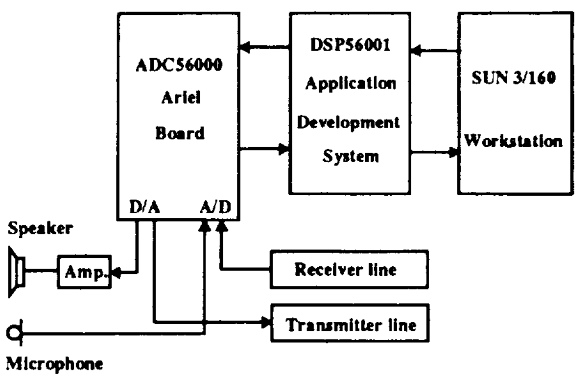

Figure 8 Hardware system set-up for experiment

8.0 Conclusions

The feasibility of implementing a full duplex hands-free cellular phone using one DSP56001 to cancel acoustic echo has been presented. Fast convergence has been achieved during the initialization stage with the lattice-LMS algorithm. After the lattice coefficients are converted to the conventional TDL structure which has 768 taps, better than 30 dB of acoustic ERLE can be theoretically achieved using a single DSP56001 by taking advantage of the 24-bit coefficient precision. The experimental set-up which will be used to verify these predictions was also described. It is hoped that sufficient and fast echo cancellation performance can be achieved by controlling the hybrid timing and the convergence parameters with this hybrid (lattice-TDL) structure.

REFERENCE

[1] K.H. Mueller, "New Digital Echo Canceller for Two-wire Full-duplex Data Transmission," *IEEE Trans. Commun.*, Vol. COM-24, pp. 956-962, Sept. 1976.

[2] S. Park and G. Hillman, "On Acoustic Echo-Cancellation Implementation with Multiple Cascadable Adaptive FIR Filter Chip," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 952-955, Glascow, Scotland, May 1989.

[3] B. Widrow, *et al.*, "Adaptive Noise Cancelling: Principles and Applications." *Proc. IEEE*, Vol. 63, No. 12, pp. 1692 - 1716, Dec. 1975.

[4] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1985.

[5] L.J. Griffiths, "A Continuously-Adaptive Filter Implemented as a Lattice Structure," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 683 - 686, March 1977.

[6] L.J. Griffiths, "An Adaptive Lattice Structure for Noise-Cancelling Applications," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 87 - 90, March 1978.

[7] T.C. Hsia, "Convergence Analysis of LMS and NLMS Adaptive Algorithms," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 667 - 670, Boston, Massachusetts, 1983.

[8] Motorola Inc., *DSP56000/DSP56001 Digital Signal Processor User's Manual*, Rev. 2, 1990.

[9] F. Itakura and S. Saito, "Digital FIltering Techniques for Speech Analysis and Synthesis," *Proc. 7th Int. Conf. Acoust.*, Budapest, pp. 261 - 264, 1971.

[10] S. Park and C. Chen, "Time-domain Algorithms for Adaptive Signal Detection Structures," *Proc. of the 22-nd Annual Conference on Information Sciences and Systems*, Princeton, NJ, Mar. 17-19, 1988.

[11] N. Ahmed and R.J. Fogler, "A matrix bordering approach for deriving lattice models," *IEEE Trans. on Aerospace and Electronics Systems*, Vol. AES-20, No. 6, pp. 835-838, Nov. 1984.

[12] N. Magotra, N. Ahmed, E. Chael, "A Comparison of Two Parameter Estimation Schemes," *Proc. IEEE*, Vol. 74, No. 5, pp. 760 - 761, May 1986.