A two stage neural filter and training algorithm for application in handsfree telephone acoustic echo cancellers

A. N. Birkett*

* Maplebrook Consulting Inc.,
74 Colonnade Road, Nepean, Ontario, K2E 7L2 Canada
e-mail: neil@maplebrook.com

1.0 Introduction

The objective of this paper is to present a two stage neural filter and training algorithm for application in handsfree telephone acoustic echo cancellers, where the loudspeaker nonlinearity limits the achievable echo cancellation. One of the limitations to achieving a high steady state *Echo Return Loss Enhancement* (ERLE) in linear *acoustic echo cancellers* (AECs) is loudspeaker nonlinearity [1]. Hence a two stage neural filter is developed to combat the effects of loudspeaker nonlinearity, consisting of a *tapped delay line neural network* (TDNN) arranged in parallel with a linear *Finite Impulse Response* (FIR) filter [2].

Simplicity of design is of utmost importance in the development AECs since the filter lengths required to cancel acoustic echoes are typically several hundreds of taps long [3]. It is therefore paramount that any nonlinear structure and training algorithms also be of low complexity. The *gradient backpropagation* (BP) algorithm [4] is an efficient training algorithm for neural networks, however, like its linear counterpart the *Least Mean Squares* (LMS) algorithm, it has slow convergence when a coloured signal like speech is used for training [5]. Training methods based on the *conjugate gradient* (CG) algorithm [6],[7] can be applied to neural networks to mitigate the slow convergence however, the complexity is higher than the BP algorithm. We propose a modified form of the partial CG algorithm [8] which uses a selectable gradient window and a fixed step size to train the network and provide a trade-off between complexity and speed.

The rest of the paper is organized as follows. In Section 2 the Acoustic Echo Cancellation problem is reviewed and the limitations to achievable steady state ERLE due to loudspeaker nonlinearity in a typical *handsfree telephone* (HFT) is presented. A two stage neural filter is developed in Section 3 and is shown to have improved steady state modelling accuracy. In Section 4, a fast conjugate gradient algorithm is developed, by modifying the partial conjugate gradient method to include a gradient window and fixed step size. Experimental results using real speech signals in a handsfree telephone conference environment are presented. Finally in Section 5 concluding remarks are presented.

2.0 Acoustic Echo Cancellation

A complete survey of the acoustic echo cancellation literature is beyond the scope of this paper, however, references [3] and [9] provide an exhaustive summary of over 100 papers in this area. An acoustic echo canceller for handsfree telephony must be capable of identifying a changing *Loudspeaker-Room-Enclosure-Room* (LREM) response which includes a room transfer function, a nonlinear loudspeaker and other components a shown in Figure 1. The adaptive filter takes the reference signal $r(n)$, generates an echo rep-

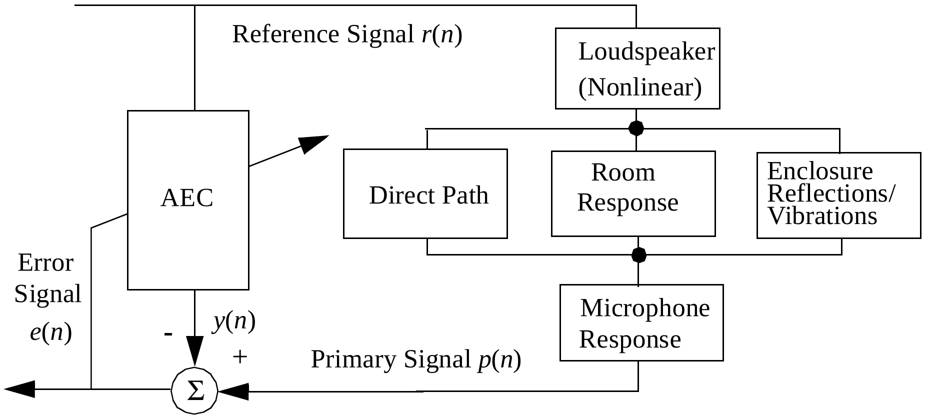


FIGURE 1. The LREM consists of both linear and nonlinear components.

lica $y(n)$ and subtracts it from the primary signal $p(n)$ to generate an error signal $e(n)$. Conventional AECs use linear FIR filters to model the LREM and cancel the unwanted echo signal, however, this architecture is incapable of reducing nonlinear loudspeaker distortion. A commonly used AEC performance metric is ERLE, which provides a measure of how much the echo is attenuated in the absence of measurement noise, defined by [10]

$$ERLE(dB) \;=\; \lim_{N \to \infty}\left[10\log\frac{E[p^2(n)]}{E[e^2(n)]}\right] \approx 10\log\left[\frac{\sigma^2{}_p}{\sigma^2{}_e}\right] \approx 10\log\left(\frac{\sum_{r=0}^{n_w}[p(n-r)]^2}{\sum_{r=0}^{n_w}[e(n-r)]^2}\right) \qquad (1)$$

where $\sigma^2{}_p$ and $\sigma^2{}_e$ refer to the variances of the primary and error signals respectively and E is the statistical expectation operator. For on-line measurements, the later expression in (1) can be used as an approximation to compute the ERLE at time n where n_w is the size of an averaging window. Typically high values of ERLE up to 45 dB are proposed for primary signals with large transmission delays [11], however current technology is unable to provide such high attenuations hence additional variable losses in the receive and/ or transmit path are frequently used. There is no mention in the literature of how physical limitations such as loudspeaker nonlinearity will affect the practical achievement of such high ERLE values without the inclusion of these additional losses.

2.1 AEC Performance Limitations

The steady state ERLE limitations of AECs in HFTs include [1] (i) undermodelling of the LREM (ii) enclosure vibration effects (iii) transducer nonlinearities (iv) room noise, DSP noise, finite precision and truncation. We concentrate here on nonlinearity and undermodelling.

Loudspeaker Nonlinearity. It has been shown [1] that the achievable steady state ERLE in desktop HFTs is limited as a function of the volume of the applied loudspeaker signal; at low volumes the ERLE is limited by noise and offsets and at medium to high volumes, nonlinearity in the loudspeaker and enclosure vibration effects dominate. For example, a *power spectral density* (PSD) plot of the primary signal obtained from a real HFT fed with a reference signal consisting of bandlimited noise is shown in Figure 2. The reference signal level is increased such that a *sound pressure level* (SPL) of between 60 and 100 dB is obtained as measured 0.5 m above the loudspeaker. An increasing volume level generates increasing nonlinear distortion products both in-band (i.e. 200-3400Hz) and out-of-band (3400-8000 Hz). The in-band distortion products are masked by the primary signal level, however, the out-of-band nonlinear and distortion products can be seen to increase with volume.

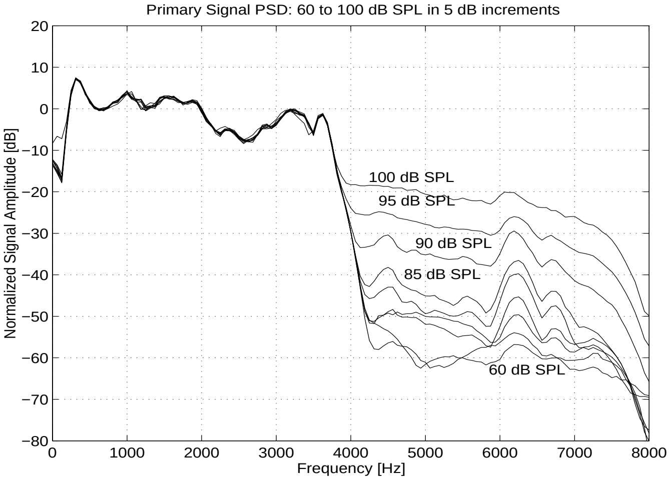


FIGURE 2. Primary signal PSD. Out-of-band components increase in level as the volume is increased from 60 to 100 dB SPL (as measured 0.5 m above loudspeaker).

The effect that the nonlinear products have on the achievable steady state ERLE is illustrated in Figure 3, which shows a comparison of the steady state ERLE vs. volume of six commercially available HFTs. The converged ERLE values are obtained by training a 1000 tap FIR filter with the *Normalized LMS* (NLMS) algorithm [5] for 80,000 iterations, using a normalized step size of 0.5, and averaging over the last 5000 iterations.

Undermodelling. An FIR structure can be used to model a transfer function where the number of parameters in the candidate system is less that required to exactly identify the system. This gives the undermod-

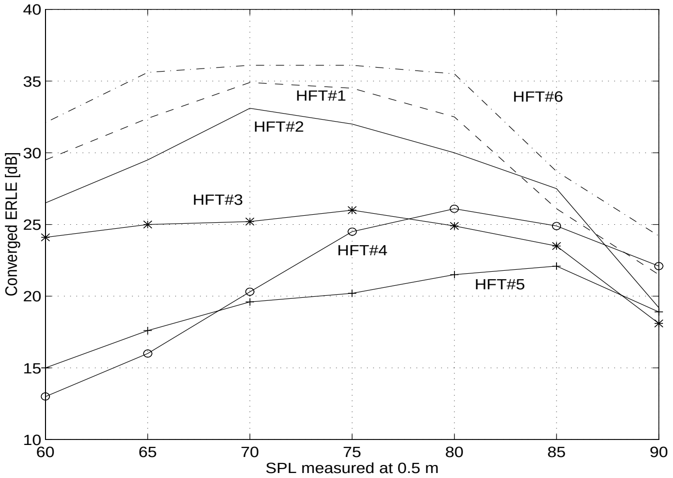


FIGURE 3. Converged ERLE for six commercially available HFTs at various sound pressure levels (measured at 1 m from loudspeaker in an anechoic chamber).

elled system: $deg(\hat{H}) < deg(H)$. Poltmann [12] showed that the achievable modelling error is a function of both the step size and magnitude of the modelled and undermodelled coefficients. For a system modelled by an FIR transfer function the achievable steady state ERLE can be calculated from;

$$ERLE(dB) = 10\log\left[\frac{2-\mu}{2}\left(\frac{\|h\|^2}{\|\Delta h\|^2}+1\right)\right] \approx TIP/TP \qquad (2)$$

where $\|h\|^2$ represents the power in the modelled coefficients up to order M-1 and $\|\Delta h\|^2$ represents the power in the tail portion of the LREM from M to infinity. If μ is set to zero, (2) is equal to the Total Impulse Power to the uncancelled Tail Power (TIP/TP) ratio originally proposed by Knappe and Goubran [13]. The TIP/TP ratio defines the achievable ERLE up to approximately 20 dB, beyond which other effects dominate. Experimental measurements in [13] show that even at ratios of (S+N)/N of greater that 40 dB the ERLE did not go beyond 25 dB, and suggest system nonlinearities as the cause. The ERLE follows the TIP/TP ratio very closely up to a certain number of taps according to (2), however, in real world experimental recordings, nonlinearities and other effects serve to limit the achievable ERLE.

Summary. The relative severity of the above limitations is illustrated in Figure 4. Vibration and nonlinearity are frequency and volume dependent. Given that vibration effects can be minimized by appropriate mechanical design procedures, nonlinear filters can have a positive effect on improving the ERLE when real world (i.e. nonlinear) loudspeakers are used.

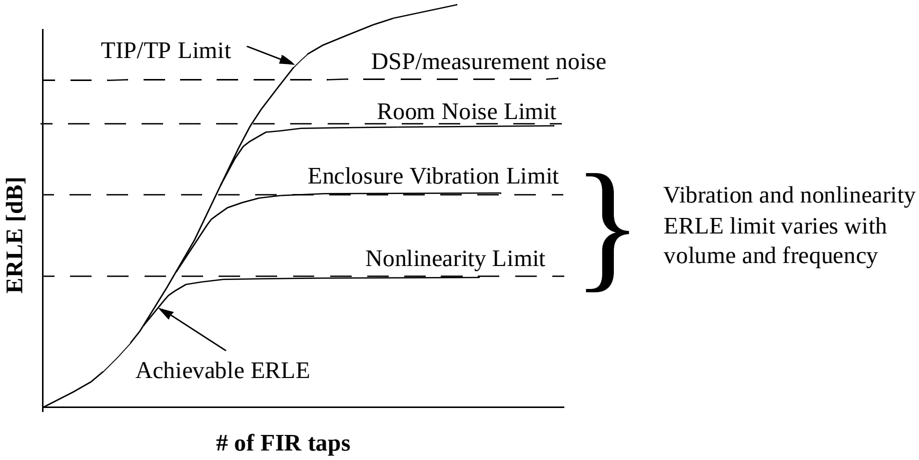


FIGURE 4. Achievable ERLE as a function of physical limitations.

3.0 Two Stage Neural Filter Architecture

The proposed neural filter structure shown in Figure 5 consists of both nonlinear and linear sections The nonlinear section consist of a two layer tapped delay line neural network (TDNN) that cancels the first part of the LREM impulse response where most of the energy is contained. The weight update equations for the nonlinear portion are based on the gradient backpropagation algorithm [4] with a normalized adaptive step size. The linear section consists of an FIR filter.

3.1 Mixed Linear-Sigmoid Activation Function

A neural filter will generate a finite amount of distortion due to the nonlinear nature of the sigmoid and will perform slightly worse than a conventional FIR adaptive filter, at low distortion values. In order to mitigate this effect, a mixed linear-sigmoid activation function is proposed. The activation function consists of a linearized hyperbolic tangent function which is linear for inputs below a user definable amplitude *p*, where $0 \le p \le 1$. By setting the parameter *p* our simulations have shown that it is possible to reduce the

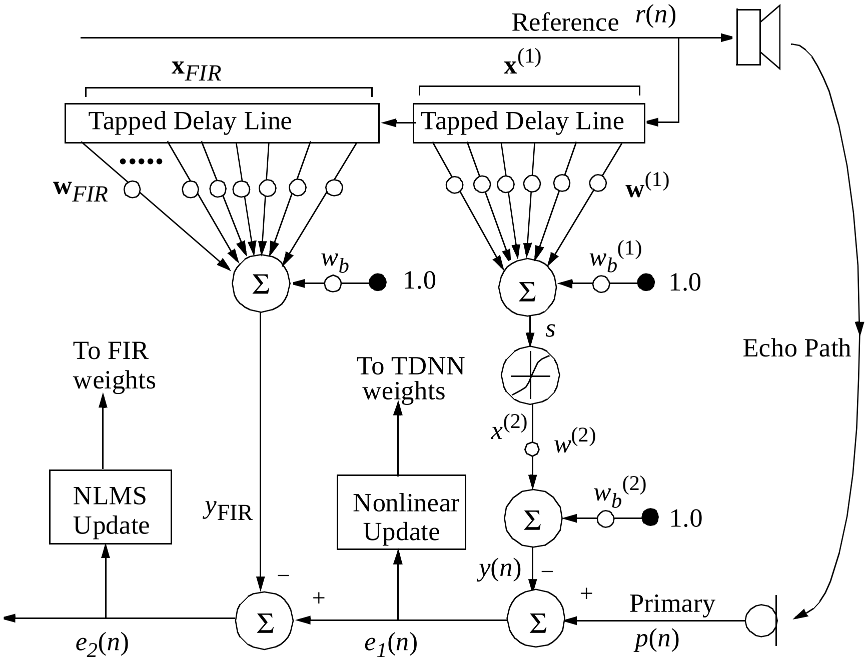


FIGURE 5. Proposed nonlinear AEC structure consists of a nonlinear tapped delay line neural network (TDNN) and linear FIR portions.

modelling error by a few dB in low distortion environments compared to a conventional (i.e. *p*=0) sigmoid. The node activation function φ(*s,p*) is defined by;

$$\varphi(s,p) = \begin{cases} s & ;|s| \le p \\ \operatorname{sign}(s)\left[(1-p)\cdot\tanh\left(\frac{|s|-p}{1-p}\right)+p\right] & ;|s| > p \end{cases} \tag{3}$$

where *s* is the input. Differentiating (3) with respect to *s*, we obtain the slope of the activation function:

$$\varphi'_s(s,p) = \begin{cases} 1 & ;|s| \le p \\ \operatorname{sign}(s)[1-\tanh(\theta)^2] & ;|s| > p \end{cases} \tag{4}$$

$$\text{where } \theta = \left(\frac{|s|-p}{}\right.$$

Figure 6 shows the activation function of equation (3) with values of *p* equal to 0.0, 0.5, and 0.9, along with the associated $\varphi'_s(s,p)$ values.

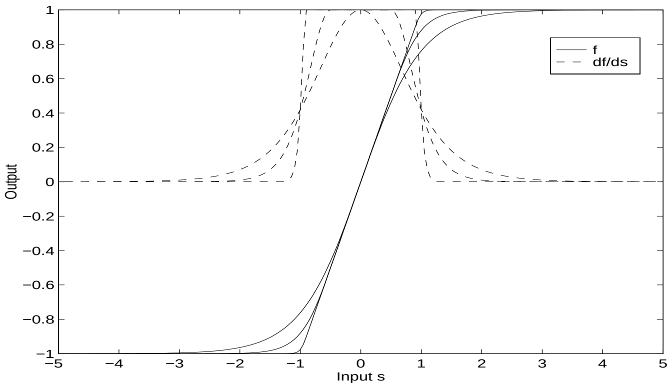


FIGURE 6. Activation function and derivative with respect to *s* for *p*=0.0, 0.5 and 0.9.

For data that is weakly nonlinear, the weights in the TDNN will adjust to provide an activation in the linear region of the sigmoid. Simulation results showing the effect of varying the linear region versus converged ERLE are shown in Figure 7 for a (10,5,1) TDNN filter. The primary signal is generated by passing filtered noise through a fixed nonlinearity which generates quadratic and cubic distortion according to

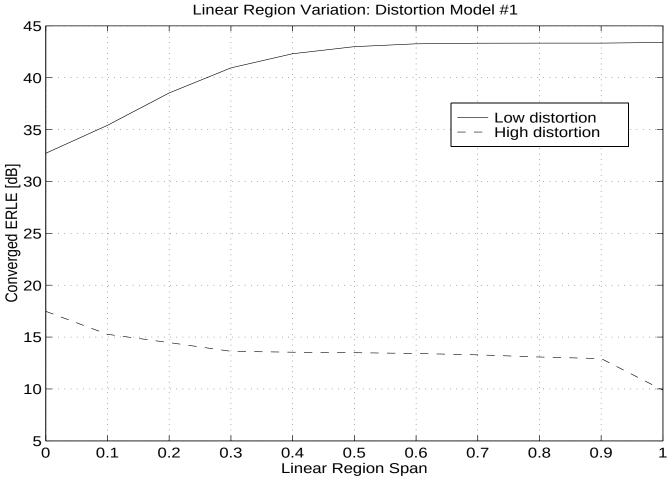


FIGURE 7. Effect of changing linear region in a mixed linear-sigmoidal activation function. (10,5,1) TDNN.

$$d(n) = \frac{ax(n) + bx^2(n) + cx^3(n)}{|a| + |b| + |c|} \qquad (5)$$

The output $d(n)$ is then convolved with a 10 tap randomly generated impulse response to yield the primary signal $p(n)$. The distortion parameters are set to a=1 and b=c=0.01 (low distortion) and b=c=0.5 (high distortion). The optimum value of linear region is highly dependent on the severity of nonlinearity encountered however the activation function can be made fully adaptive (for example see [15]). For our purposes however, the parameter p was set to 0.2 since it was found experimentally that this produced an ERLE approximately 1.5 dB higher than with a conventional (i.e. p=0) sigmoid and was considered as a good compromise between the two extremes.

3.2 BP Weight Update Equations

In Figure 5, the output $y(n)$ of the neural network portion at time n is defined by;

$$y(n) = w^{(2)}(n)x^{(2)}(n) + w_b{}^{(2)}(n) \tag{6}$$

$$x^{(2)}(n) = f(s(n)) \tag{7}$$

$$s(n) = \mathbf{w}^{(1)}(n)^T\mathbf{x}^{(1)}(n) + w_b{}^{(1)}(n) \tag{8}$$

where $\mathbf{x}^{(l)}(n)$ represents the input vector to layer l, $\mathbf{w}^{(l)}(n)$ represents the weight vector in layer l, $w^{(l)}{}_b(n)$ represents the single bias weight for layer l, $s(n)$ represents the input to the nonlinear node and T is the transpose operator. The weight update equations are described by;

$$\mathbf{w}^{(l)}(n+1) = \mathbf{w}^{(l)}(n) - \mu_{TDNN}(n)\delta^{(l+1)}(n)\mathbf{x}^{(l)}(n) \tag{9}$$

$$w_b^{(l)}(n+1) = w_b{}^{(l)}(n) - \mu_{TDNN}(n)\delta^{(l+1)}(n) \tag{10}$$

$$\delta^{(l+1)}(n) = \begin{cases} -2e_1(n) & ;l=2,\text{output layer} \\ f'(s(n))\delta^{(l+2)}(n)w^{(l+1)}(n) & ;l=1,\text{hidden layer} \end{cases} \tag{11}$$

where $e_1(n) = p(n) - y(n)$, $f'(\,)$ represents the derivative of the activation function at the input value $s(n)$, $\delta^{(l+1)}(n)$ represents the local gradient “delta” term in layer $l+1$, and $\mu_{TDNN}(n)$ is the normalized step size parameter defined by;

$$\mu_{TDNN}(n) = \frac{\alpha}{2 + \mathbf{x}^{(1)}(n)^T \mathbf{x}^{(1)}(n) + [x^{(2)}]^2} \qquad (12)$$

The parameter α is a number between 0 and 2, and is set to 0.5. The weights in the linear portion of the proposed structure are updated using the NLMS algorithm which also has a DC bias compensation update to compensate for real world DC offsets;

$$\mathbf{w}_{FIR}(n+1) = \mathbf{w}_{FIR}(n) - \left[\frac{\alpha}{1 + \mathbf{x}_{FIR}(n)^T \mathbf{x}_{FIR}(n)}\right] e_2(\dot{n}) \cdot \mathbf{x}_{FIR}(n) \qquad (13)$$

$$w_b(n+1) = w_b(n) - \left[\frac{\alpha}{1 + \mathbf{x}_{FIR}(n)^T \mathbf{x}_{FIR}(n)}\right] e_2(n) \qquad (14)$$

3.3 TDNN Order Selection

Experimental data was applied to a TDNN filter to determine the optimum length for the delay line section. The results shown in Figure 8 illustrate that for an undermodelled system, a TDNN structure has improved ERLE performance compared to the stand alone FIR structure trained with the NLMS algorithm. The experimental data was obtained from HFT #6 transducer components recorded in an anechoic chamber at a volume of 100 dBSPL measured at 0.5 m. A (n_0,2,3,1) TDNN model is used where n_0 is a variable number of input taps. The best performance occurs when n_0=150 taps. Here the difference between the TDNN and FIR ERLE value is approximately 5.5 dB.

3.4 Measurement Setup

Measurements are performed in a low-noise, furnished conference room . A handsfree telephone (HFT #6) which has been modified to allow access to the primary and reference electrical signals is placed on top of the conference table. The reference source signal consists of white noise which is bandlimited from 300 Hz to 3400 Hz. The filtered reference signal is then amplified such that the loudspeaker produces a sound pressure level from 60dB to 95dB as measured 0.5m directly above the loudspeaker. The primary and reference signals are then recorded onto a TEAC Digital Audio Recorder (DAT). The DAT signals are down-

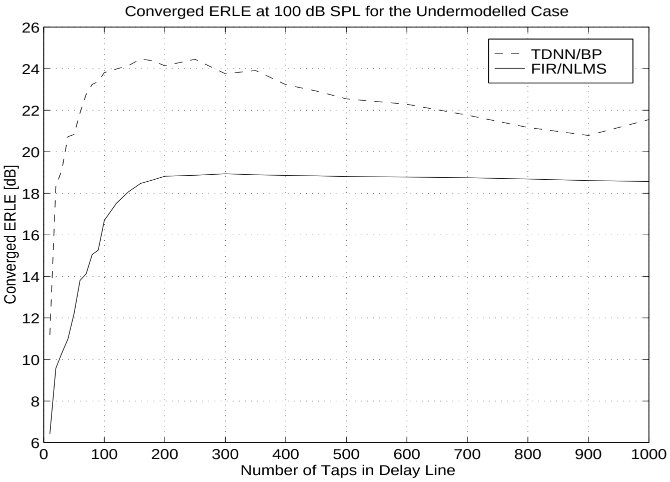
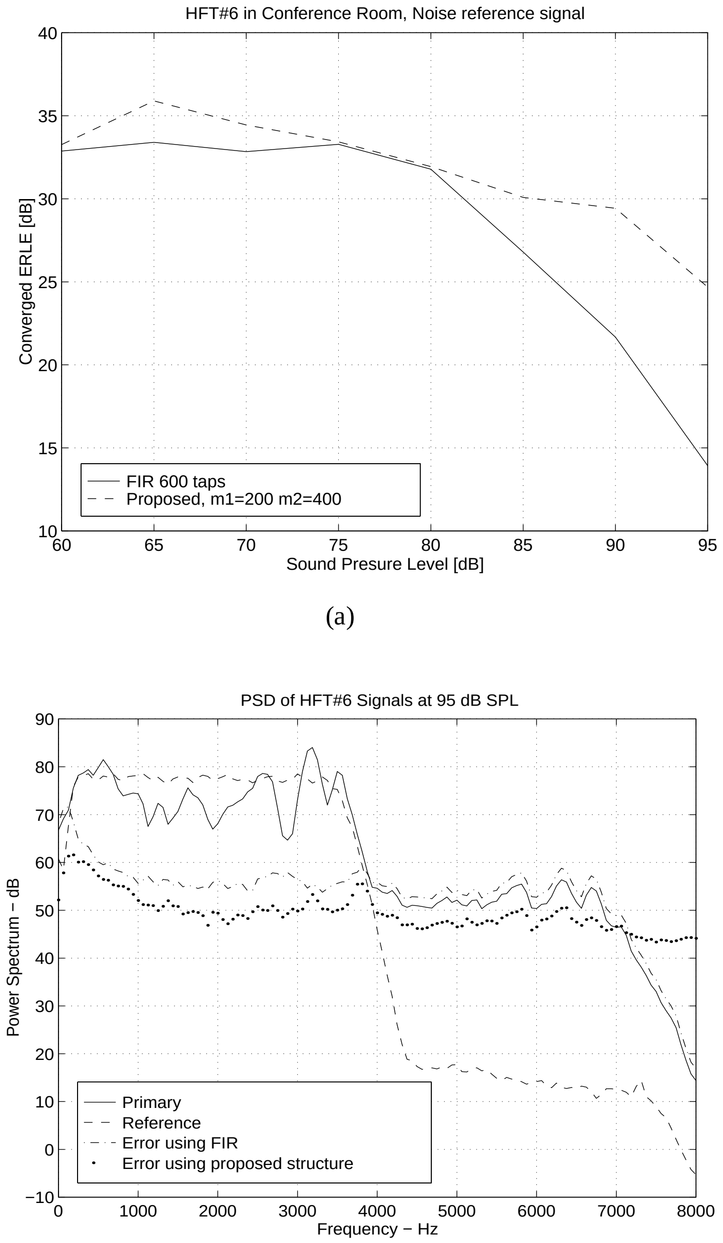


FIGURE 8. Experimental results for HFT components in anechoic chamber. A TDNN is capable of obtaining a better ERLE in an undermodelled state as compared with the NLMS algorithm. Results obtained at a high volume level of 100 dB SPL measured at a distance of 0.5 meter.

loaded to a computer via an ARIEL DSP96 board sampling at 16 kHz. These samples are then applied to both the proposed structure and a 600 tap linear adaptive FIR filter which has DC bias compensation and weights updated in the same fashion as equations (13) and (14). In the proposed structure, the number of taps in the nonlinear section delay line equals 200 to cover the bulk of the loudspeaker impulse response. The number of taps in the linear section is 400 for a total impulse length of 600 taps. For each SPL, both algorithms are tested with the same input data of length 80,000 to allow convergence to a steady state at which point the average ERLE is measured and plotted.

3.5 Experimental Results Using Noise

In Figure 9 (a), over 11 dB of improvement can be seen at 95 dB SPL compared to the linear algorithm, and between 0-2 dB improvement is obtained over the rest of the volume range. At low volumes in the vicinity of 65 dB SPL, the proposed structure improves the ERLE by 3 dB as compared to the linear adaptive filter even though there is little nonlinear distortion in this range. In the low volume ranges, two-point suspension nonlinearities are present in the loudspeaker movement [14] and the proposed structure offers some improvement. In the medium volume range from 70-75 dB SPL, the proposed structure performs as well as the linear structure. However, in the vicinity of 80 to 95 dB SPL where nonlinear effects dominate, the proposed structure clearly outperforms the linear filter in terms of converged ERLE and demonstrates over 11 dB improvement at 95dB SPL. Figure 9 (b) shows the corresponding power spectral density of the primary and reference signals, as well as the error signals for the linear and nonlinear algorithms. The error signal generated by the FIR filter closely follows the primary signal out-of-band. The error signal for the proposed structure is several dB lower across the full spectrum.



(a)

(b)

FIGURE 9. Experimental results showing performance of the proposed structure using HFT #6 in a furnished conference room. (a) Converged ERLE, keys taped down. (b) plot of PSD of signals.

4.0 Fast Conjugate Gradient Backpropagation

In this section the *nonlinear fast conjugate gradient* (NFCG) backpropagation algorithm is presented as an alternative to the conventional BP algorithm to speed convergence. The conventional BP algorithm is probably the most widely used supervised learning algorithm in neural network applications. However, with a large number of weights, the BP learning time is excessively long and its use becomes impractical. The conjugate gradient algorithm is well suited for the neural network learning problem since it is fast, simple and requires little additional storage space. The CG method speeds up the BP learning time significantly and does not suffer from the inefficiencies and possible instabilities that arise using the BP with a fixed step size. In fact, the CG algorithm has been found in some studies [7] to be an order of magnitude faster than the conventional BP using momentum. However, the CG computational burden is still quite high compared to BP.

Partial CG methods (see [6][16]) can simplify the CG algorithm complexity and can be considered a stepping point for the formulation of *fast* (i.e. numerically less intensive) versions of the CG algorithm. Boray and Srinath [17] recently developed a *fast conjugate gradient algorithm* (FCG) for linear adaptive filtering using an averaged instantaneous gradient over a *window* of past sample values. They showed that the advantages of this windowed approach are (i) better tracking and convergence is achieved in nonstationary environments with correlated data compared to the *Recursive Least Squares* (RLS) algorithm, and (ii) there are no stability problems associated with an exponential forgetting factor as in the RLS algorithm.

Here we extend the FCG algorithm to the nonlinear case, for neural networks. The differences are (1) the network is nonlinear (2) the errors must be computed for hidden layers and not just the output layer (3) the previous values of the *hidden layer* outputs must be retained as well as the output layers in order to compute the gradient. The gradient is computed using the average squared error of a *window* n_w of training input/output pairs. Another important difference is that the optimum step size, which is calculated for each iteration in the standard CG is now replaced by a fixed step size, as proposed in [17]. This has the effect of substantially reducing the computational burden. Expressions for the CG and BP algorithms have been developed by several authors, including Charlambous [18], Johansson *et. al.* [7] , as well as Adeli and Hung [19]. However, these expressions were based on the *batch* training mode using the *full* set of input/output pairs as well as requiring the determination of the optimum step size by direct calculation or a line search.

The NFCG algorithm is summarized below. Errors are backpropagated to *previous* layers in the same way as the conventional BP algorithm. The important point is that the window is moved for each new sample of the input that comes in i.e. it is a *sliding* window n_w of past input/output pairs.

Nonlinear FCG (NFCG) Algorithm

Initialization: Set weights and biases to random values.

For each iteration *n*, do Steps 1 2 and 3.

Step 1. a) Starting with an initial weight vector $\mathbf{w}_0$, compute the following;

$$\mathbf{g}_0 = [\overline{\nabla f(\mathbf{w}_0)}] = \left(\frac{2}{n_w}\right)\left[\sum_{i=0}^{n_w-1} \mathbf{g}_{inst}(n-i)\Big|_{\mathbf{w}_0(n),\,\mathbf{u}^0(n-i),\,d(n-i)}\right] \tag{15}$$

where $\mathbf{g}_{inst}(n\text{-}i)$ is the *instantaneous* gradient calculated with the current network weight vector $\mathbf{w}_0(n)$ and past inputs $\mathbf{u}^o(n\text{-}i)$ and $d(n\text{-}i)$. Both $\mathbf{g}_{inst}(n\text{-}i)$ and $\mathbf{w}_0(n)$ are vectors of length *M*, where *M* is the total number of weights in the network.

b) set $\mathbf{d}_0 = -\mathbf{g}_0$

c) compute the normalized step size parameter α according to;

$$\bar{\alpha} = \frac{\gamma}{\varepsilon + \|\mathbf{u}(n)\|^2} = \frac{\gamma}{\varepsilon + \mathbf{u}^T(n)\mathbf{u}(n)} \tag{15.1}$$

Note that α could be replaced by a fixed step size here if desired;

Step 2. Repeat for k=0,1,. n_w-1 where $n_w \le m$

a) set $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha\mathbf{d}_k$

b) Compute an estimate of the gradient at $\mathbf{w}_{k+1}$;

$$\mathbf{g}_{k+1} = [\overline{\nabla f(\mathbf{w}_{k+1})}] = \left(\frac{2}{n_w}\right)\left[\sum_{i=0}^{n_w-1} \mathbf{g}_{inst}(n-i)\Big|_{\mathbf{w}_{k+1}(n),\, \mathbf{u}^0(n-i),\, d(n-i)}\right] \qquad (16)$$

c) Unless k=n_w-1, set $\mathbf{d}_{k+1}$= - $\mathbf{g}_{k+1}$ + $\beta_k\mathbf{d}_k$,where;

$$\beta_k = \frac{\mathbf{g}_{k+1}^T\mathbf{g}_{k+1}}{\mathbf{g}_k^T\mathbf{g}_k} \qquad (16.1)$$

Note that if $\beta_k > 1$, *go directly to Step three.*

Repeat *Step 2 a).*

Step 3. Replace $\mathbf{w}_0$ by $\mathbf{w}_k$ and go back to *Step 1.*

The calculation of individual elements of the instantaneous gradient vector $\mathbf{g}_{inst}(n-i)$ is done by performing the following steps;

$$g_{ij}^{(l)}(n-i) = \delta_j^{(l+1)}(n-i) \cdot u_i^{(l)}(n-i) \qquad (17)$$

where $g_{ij}^{(l)}(n-i)$ is the instantaneous gradient from the data i time steps in the past corresponding to weight $w_{ij}^{(l)}(n)$ in the *l-th* layer, and;

$$\delta_j^{(l)}(n-i) = \begin{cases} -2e(n-i)\varphi'(s_j^{(L)}(n-i)) & \ldots l = L \\ \varphi'(s_j^{(l)}(n-i)) \cdot \displaystyle\sum_{k=1}^{N_{L+1}} \delta_k^{(l+1)}(n-i) \cdot w_{jk}^{(l)}(n) & \ldots 1 \le l \le L-1 \end{cases} \qquad (18)$$

$$e(n-i) = d(n-i) - N[\mathbf{w}_{k+1}(n), \mathbf{u}^0(n-i)] \qquad (19)$$

Note that $N[\mathbf{w}_{k+1}(n), \mathbf{u}^0(n-i)]$ represent the nonlinear output of the neural network at time *n* using the current weight vector $\mathbf{w}_{k+1}(n)$ with past input vectors $\mathbf{u}^0(n-i)$.

Complexity. The choice of n_w =1 implies no averaging in the gradient estimate and the NFCG algorithm reverts to the BP algorithm. For higher values of n_w the complexity approaches that of algorithms that use the second derivative for obtaining the optimum step size and direction which have complexity $O(m^3)$[20] where *m* is the total number of weights in the network. The complexity of the NFCG algorithm is $O(mn_{\mathrm{w}}{}^2)$ since in *Step 2*, the weights are updated n_{w} times per iteration and the calculation of the averaged gradient is $O(mn_{\mathrm{w}})$.

4.1 Computer Simulation

In this section, we apply the NFCG algorithm to the identification of a nonlinear system constructed by generating a signal which is hard limited and convolved with an exponentially decaying 50 tap impulse.

The system is illustrated in Figure 10. The input signal *x*(*n*) is obtained by a first order autoregressive (AR) process according to

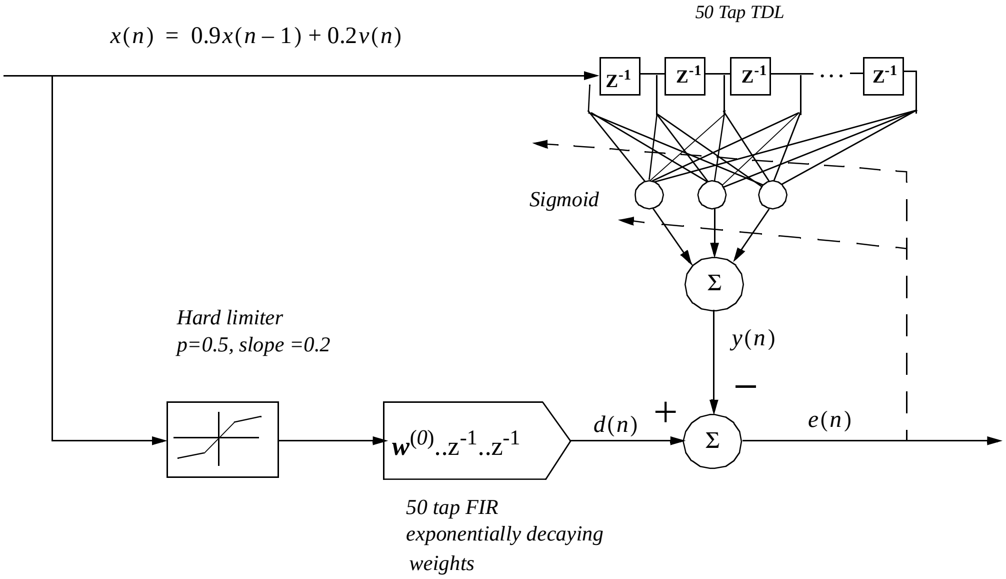


FIGURE 10. System identification model.

$$x(n) = 0.9x(n-1) + 0.2v(n) \qquad (20)$$

where *v*(*n*) is a unit variance white noise sequence. The hard limiter has a linear region up to 0.5, beyond which the output is clipped with a slope of 0.2. Two hundred independent trials are used in the averaging of the Normalized Mean Square Error (NMSE).

The results illustrated in Figure 11 show that for the AR input, the NFCG algorithm converges at a rate much faster than the conventional BP algorithm, depending on the size of the gradient averaging window n_w. The larger the choice of n_w, the higher the convergence rate. The final misadjustment is approximately -18 dB for all cases.

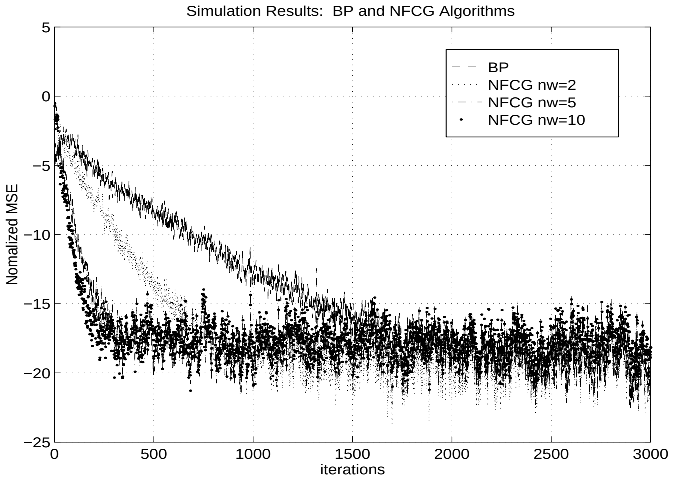


FIGURE 11. Simulation results showing the averaged NMSE performance of the BP and NFCG algorithms with n_w=2, 5, and 10 for the system identification model of Figure 10. Two hundred independent trials are used in the averaging process

Convergence Rate Improvement. The convergence rate improvement is not a linear function of the window size. For example, the BP algorithm, which is equivalent to the NFCG with n_w=1, takes approximately 1400 iterations to reach -15 dB NMSE. For window sizes n_w=2, 5, and 10, the number of iterations required to reach the same NMSE are approximately 600,200 and 150 respectively. As a result, it can be seen that the convergence rate improvement becomes progressively smaller for large window sizes, and that for n_w>5, the convergence rate improvements are small.

4.2 Experimental Results Using Noise and Speech Signals

In this section two computer experiments are performed using data collected from actual LREM and HFT components. The data collection method is similar to that presented in Section 3.4. In experiment #1, a filtered noise signal is applied to an HFT loudspeaker which is mounted in a standard loudspeaker baffle and placed inside an anechoic chamber. This is the reference signal. The primary signal is picked up by a microphone placed 10 cm. in front of the loudspeaker. The primary and reference signals are then applied to a conventional TDNN structure which is trained with the BP and NFCG algorithms.

In experiment #2, data is collected inside a furnished conference room using HFT#6. Speech signals were applied as the reference signal. The primary and reference signals are applied to the parallel cascade TDNN-FIR structure and the nonlinear section is trained with the BP and NFCG algorithms. For comparison purposes, the performance of an FIR filter trained with the accelerated *stabilized fast transversal filter* (SFTF) algorithm is also shown. The accelerated SFTF algorithm [21][22] is used to remove the long training time associated with LMS based training algorithms when using speech inputs, which may be as long as 10 seconds.

Experiment #1, Noise Input. The volume is 100 dB SPL as measured at 0.5 meters from the loudspeaker. The microphone is placed 15 cm. from the loudspeaker output. The signals are sampled at 16 kHz and are later transferred to a computer for off-line analysis. Two adaptive filter structures were tested to identify the system (i) a 150 tap linear transversal filter trained using the NLMS algorithm (ii) a 3 layer TDNN with 150 input taps trained with both the BP and NFCG algorithms. The experimental results shown in Figure 12 show the results for all cases. The NLMS has fast convergence but is incapable of obtaining an ERLE of greater than 19 dB due to the nonlinear loudspeaker. The TDNN trained with the BP algorithm is capable of identifying the system more effectively and achieves 25 dB ERLE but the initial convergence is much slower than the NLMS algorithm. Training the TDNN using the NFCG with a window size n_w=5 results in convergence speed equivalent to the NLMS structure as well as obtaining 24 dB ERLE.

Experiment #2, Speech Input. The average volume of the speech signal as measured 0.5 m from the loudspeaker is 95 dB SPL, which is a comfortable listening level 6-10 ft. from the HFT. The HFT is placed in the middle of the conference table. The parameters are listed in Table 1 .

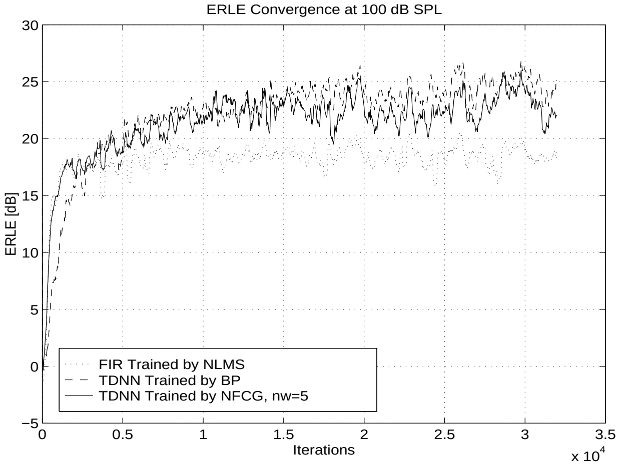


FIGURE 12. Experimental results comparing converged ERLE curves of a 150 tap FIR structure trained using the NLMS algorithm with that of a TDNN trained with the BP and NFCG algorithm.

TABLE 1. Experiment #2 parameters.

Item	**Parameters**
Data	160,000 samples @16 kHz sampling. 95 dB SPL average volume at 0.5 m.
FIR Trained with Accelerated SFTF	N=600, λ=0.9998, acceleration factor=0.95, soft initialization constant=200.
TDNN-FIR trained with NFCG algorithm	N_1=150, N_2=450, number of hidden nodes=1, neural network normalized step size α=0.5, nlms step size α=0.5,window size n_w=5 for TDNN section

Figure 13 shows the speech signal amplitude as a function of time. The converged ERLE results shown below in Figure 14 and Figure 15 indicate that the proposed structure/algorithm outperforms the FIR structure trained with the accelerated SFTF algorithm by approximately 5 dB.

4.3 Discussion

The results presented in this section have shown that the NFCG algorithm is capable of improving the convergence rate of neural network based adaptive filters. When applied to the TDNN-FIR structure, the NFCG algorithm achieves a 5 dB improvement in ERLE compared to the accelerated SFTF algorithm when trained with real speech signals at loud volumes where loudspeaker nonlinearities become signifi-

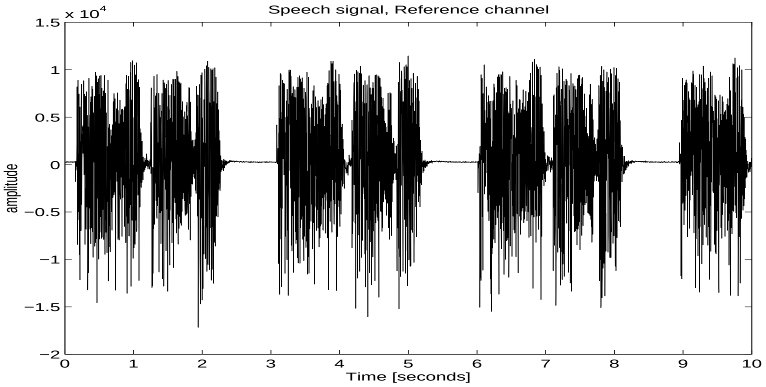


FIGURE 13. Reference signal speech signal.

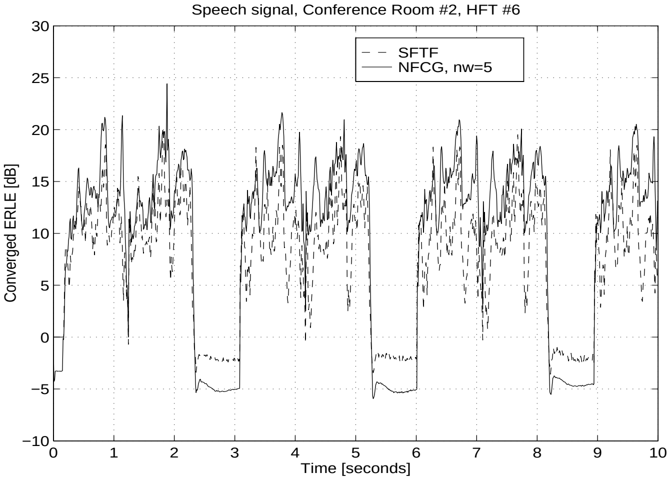


FIGURE 14. Experimental results. Converged ERLE results with speech input. Gaps show where pauses in speech are located.

cant. Simulation results in Section 4.1 also indicate that by varying the size of the gradient window n_w, we can obtain improved convergence speed with a corresponding increase in complexity. A window size of n_w=5 was found sufficient to speed the initial convergence rate of the TDNN-FIR structure to be no worse than the linear FIR trained with the NLMS algorithm, when applied to data collected from a loudspeaker/ microphone placed in an anechoic chamber.

One of the important features of the NFCG algorithm is that the gradient window n_w can be made arbitrarily small to “tailor” the algorithm to a particular application. Thus, where a modest increase in conver-

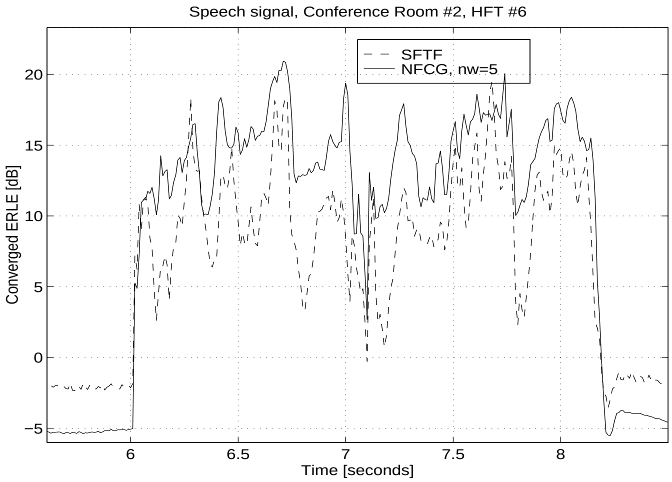


FIGURE 15. Experimental results. Close up of speech period between 6 and 8 seconds. TDNN-FIR nonlinear structure trained with proposed algorithm achieves a higher ERLE that the FIR filter trained with stabilized SFTF algorithm.

gence is desired without compromising tracking ability, a small n_w can be chosen. Low values of n_w will result in slower convergence, however, the advantages are reduced complexity and faster tracking capability.

5.0 CONCLUSIONS

A novel two stage neural filter for application in compensating system nonlinearities in handsfree acoustic echo cancellers was presented in this paper. A fast nonlinear training method based on the conjugate gradient algorithm has also been presented. Simulation results have shown that the training algorithm can provide a speed/complexity trade-off. Experimental results obtained from real world data have shown that the proposed structure is capable of achieving 11 dB of improvement in steady state ERLE when noise signals are applied at high volume to an HFT in a conference room environment. When trained with the NFCG algorithm, the proposed structure is capable of approximately 5 dB improvement in ERLE compared to a linear FIR trained with the accelerated SFTF algorithm.

REFERENCES

[1] A.N. Birkett, R. A. Goubran, “Limitations of Handsfree Acoustic Echo Cancellers due to Nonlinear Loudspeaker Distortion and Enclosure Vibration Effects”, in *1995 IEEE ASSP Workshop on Appl. of Sig. Proc. to Aud. and Acoustics*, New Paltz, New York, Oct. 1995.

[2] A. N. Birkett, R. A. Goubran, “Nonlinear loudspeaker compensation for handsfree acoustic echo cancellation”, *IEE Electronics Letters*, Vol. 32, No. 12, pp. 1063-1064, June 1996.

[3] E. Hansler, “The Hands-Free Telephone Problem: An Annotated Bibliogray ”, *Signal Processing,* Vol. 27,1992, pp. 259-271

[4] D.E. Rumelhart, G.E. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1. Cambridge, MA:MIT Press, 1986.

[5] S. Haykin, *Adaptive Filter Theory,*3rd Ed. Upper Saddle River, NJ: Prentice Hall, 1996

[6] M. R. Hestenes, *Conjugate Direction Methods in Optimization*, Springer-Verlag, 1980.

[7] E.M Johansson, F. U. Dowla, D. M. Goodman, “Backpropagation Learning for Multi-layer Feed-forward Neural Networks Using the Conjugate Gradient Method”, *International Journal of Neural Systems*, Vol.2, No. 4, pp. 291-302, 1991.

[8] A.N. Birkett, R.A. Goubran, “Fast nonlinear adaptive filtering using a partial conjugate gradient algorithm”, Proceedings ICASSP’96, Atlanta Georgia, Vol 6, pp. 3542-3545, May 1996.

[9] E. Hansler, “The Hands-Free Telephone Problem: An Annotated Bibliogray Update”, *Ann. Telecommun.* Vol. 49, No. 7-8, 1994, pp. 360-367.

[10] R. Wehrmann, J.V.D. List, P. Meissner, “A Noise Insensitive Compromise Gradient Method for the Adjustment of Adaptive Echo Cancellers”, *IEEE Trans. Comm*. COM-28, No. 5, 1980, pp. 753-759

[11] A. Gilloire, “Performance Evaluation of Acoustic Echo Control: required Values and Measurement Procedures”, *Annales des Telecommunications*, Vol. 49, No. 7-8, Jul.-Aug. 1994, pp. 368-372.

[12] R. D. Poltmann, “Stochastic Gradient Algorithm for System Identification Using Adaptive FIR-Filters with too Low Number of Coefficients”, *IEEE Trans. on Circ. and Syst.*, Vol. 35, No. 2, Feb. 1988, pp. 247-250.

[13] M.E. Knappe, R.A. Goubran,”Steady State Performance Limitations of Full-Band Acoustic Echo Cancellers”, Presented at *ICASSP* 1994, Australia.

[14] H. F. Olson, *Acoustical Engineering*, Toronto: D. Van Nostrand Company Inc., 1964.

[15] M. Teshnehlab, K. Watanabe, “Neural network controller with flexible structure based on feedback-error-learning approach”, Jun. of Intelligent and Robotic systems, Vol. 15, pp. 367-387, 1996.

[16] D. G. Leunberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, 1973

[17] G. K. Boray, M. D. Srinath, “Conjugate Gradient Techniques for Adaptive Filtering”, *IEEE Trans. on Circ. and Sys*. Vol. CAS-1, pp. 1-10, Jan. 1992.

[18] C. Charlambous, “Conjugate Gradient Algorithms for Efficient Training of Artificial Neural Networks”, *Proc. IEEE*, Vol. 139, No. 3, pp. 301-310, 1992.

[19] Adeli, H., and S.L. Hung, “An adaptive conjugate gradient learning algorithm for efficient training of neural networks”, *Applied Mathematics and Computation*, Vol. 62, 1994, pp. 81-102

[20] W. Buntine, A. A. Weigend,” Computing Second Derivative in Feed-Forward Networks: A review”; *IEEE Trans. Neural Networks*, Vol. 5, No. 3, pp. 481-488, May 1994.

[21] A Benallal and A Gilloire, "Improvement of the Tracking Capability of the Numerically Stable Fast RLS Algorithms for Adaptive Filtering", Proc. ICASSP 1989, pp. 1031-1034.

[22] A. Gilloire, T. Petillon, “A Comparison of NLMS and Fast RLS Algorithms for the Identification of Time-varying Systems with noisy outputs”, *Signal Processing V: Theories and Applications*, L. Torres, E. Masgrau and M.A. Lagunas (eds.), Elsevier Science Publishers B.V., 1990, pp.417-420.