REAL TIME IMPLEMENTATION OF AN ADAPTIVE FILTER FOR SPEECH ENHANCEMENT

John Creighton and R.Doraiswami

Dept. of Electrical & Computer Engineering, University of New Brunswick
Fredericton, NB, Canada, E3B 5A3, E-mail: s243a@unb.ca

Abstract

An adaptive linear prediction filter for speech enhancement is implemented in real time on a PC interfaced to an A/D and D/A converter board. A Least Means Squares (LMS) algorithm is employed to update the filter weights where the learning factor is adaptively adjusted to provide faster convergence. The prediction horizon is chosen to be larger than the correlation length of the noise, thereby not restricting the noise to be white. The voice activity and noise segment of the speech waveform are detected by using the energy of the adaptive filter output. This is used to attenuate the noise-only portion. The proposed scheme is evaluated on number of speech samples.

***Keywords**: linear prediction, adaptive filter, speech enhancement, LMS algorithm, real time algorithm*

1. INTRODUCTION

Speech enhancement poses a challenge both from the theoretical development as well as its practical implementation. A number of schemes have been proposed in the literature, which range from very simple adaptive filtering, to spectral subtraction, to extremely complex signal space methods, [1], [2], [3], [5], and [6]. Although these schemes have been evaluated on general purpose computers using application software such as MATLAB, their performance in real time computing environment has not been given adequate attention. Moving an algorithm from a general purpose to a real time computing environment may not always be successful due to finite wordlength effects, and the stringent requirement of computational speed. There is a trade-off between complexity of the algorithm and the speed of computation, and the issues of robustness and accuracy. One may be surprised to find a simpler algorithm outperform a 'superior' algorithm in a real time environment due to lack of robustness to finite word length effects. Signal space based algorithms perform better but they are too complex to be implemented in real time. They employ the Singular Value Decomposition (SVD) technique, which has the computational complexity of the order N^2 where N is the data size. On the other hand, the simple scheme such as the adaptive filtering is implementable in real time as it is numerically robust, efficient and fast.

CCECE 2004 - CCGEI 2004, Niagara Falls, May/mai 2004
0-7803-8253-6/04/$17.00/© 2004 IEEE

Adaptive filtering to enhance speech has been proposed in [1],[2], and [3]. In [1] the pitch period is estimated and the adaptive linear predictor is employed with the prediction horizon equal to the pitch period with a view to increase the correlation between the speech samples which are one pitch period apart. Other methods [2] avoid pitch estimation using forward and backward-adaptive filter, but require speech/silence detectors which are prone to errors in presence of noise. In [3] oversampling is employed to increase the correlation between adjacent samples. Further a computationally intense and recursive least-squares method is used to update the weights. The performance of the adaptive linear prediction filter is compared with that of the spectral subtraction method. Their results show that the performance of adaptive linear prediction filter is as good if not better.

In this paper the main ideas proposed in [1], [2], and [3] are implemented but tailored to meet the computational requirements of an inexpensive A/D and D/A input-output board. No pitch estimation, speech/silence detection, or oversampling scheme is employed. Further a simple Least-Mean-Squares (LMS) algorithm is used for filter weight update. The order of the FIR predictor and the prediction horizon are determined from the iterative design involving off line coupled with real time evaluation. The offline evaluation helps to determine an appropriate theoretical design approach, while the real time evaluation helps to develop algorithms which are accurate and robust to finite wordlength effects, and meet the real time constraints of speed. The over all design is iterative. Various real time implementable algorithms were evaluated. The final choice of an algorithm is a best compromise between performance and robustness and speed of computation.

An iterative design and evaluation procedure is employed to determine: the order of the FIR filter, the prediction horizon, and the learning factor of the LMS algorithm. Initial evaluation of the algorithm was based on objective measures, namely, Itakura-Saito, and SNR distance measures between the original clean speech and its estimate obtained from the adaptive filter [5]. However, the final evaluation was based on a subjective listening test measure. It is assumed that the noise corrupting the speech is uncorrelated with the speech waveform. The learning factor was chosen to be adaptive and large enough to provide rapid convergence, and small enough to reduce the output noise variance.

Once the design was satisfactory, it was implemented in real time on a PC interfaced to an A/D and D/A input-output board.

2. ADAPTIVE LINEAR PREDICTOR

The adaptive linear predictor is given by

$\hat{s}(k) = \mathbf{w}^T(k)\mathbf{y}(k-d) \quad where$

$\mathbf{y}(k-d) = [y(k-d) \quad y(k-1-d) \quad \cdots \quad y(k-M+1-d)]^T$

$\mathbf{w}(k) = [w_0(k) \quad w_1(k) \quad \cdots \quad w_{M-1}(k)]^T$

where $\{w_i(k)\}$ are the filter weights, $\hat{s}(k)$, is the estimate of the speech waveform, s(k), and y=s+v, is the noisy speech, and v is the noise. It is assumed that the noise v is: uncorrelated with s; the delay, d, chosen such that and v(k) and v(k-d) are uncorrelated and s(k), s(k-d) are highly correlated. The weights are determined from the minimization of

$\min_{\{w_i\}}\{J(k) = e^2(k)\} \quad where \quad e(k) = y(k) - \hat{s}(k)$

The gradient of J(k) with respect to **w**(k), yields

$\nabla J(k) = -2e(k)\mathbf{y}(k) \quad where$

$$\nabla J(k) = \left[\frac{\partial J(k)}{\partial w_0(k)} \quad \frac{\partial J(k)}{\partial w_1(k)} \quad . \quad \frac{\partial J(k)}{\partial w_{M-1}(k)}\right]^T$$

The weight, **w**(k), is in general a random vector that depends upon **y**(k)

2.1 Mean-squared error performance

We will show that the d-step predictor, $\hat{s}(k)$, obtained by minimizing the prediction error, e(k), is the best estimate of s(k) in statistical mean-squares sense.

A. Minimizing $e^2(k)$ implies minimizing $(s(k)-\hat{s}(k))^2$

Consider the variance of the prediction error e(k)

$E[e^2(k)] = E[(y(k)-\hat{s}(k))^2] = E[(s(k)+v(k)-\hat{s}(k))^2]$

Since E[s(k)v(k-d-i)]=0 for all i, we get

$E[e^2(k)] = E[(s(k)-\hat{s}(k))^2] + E[v^2(k)]$

This shows that minimizing the prediction error results in the minimization of the error in estimating the signal

$\arg\left\{\min_{\{w_i\}}\{E[e^2(k)]\}\right\} = \arg\left\{\min_{\{w_i\}}\{E[(s(k)-\hat{s}(k))^2]\}\right\}$

Using the orthogonality condition namely

$E[e(k)y(k-d-1)] = 0$

It can be shown that e(k) is a white noise,

$E[e(k)e(k-i)] = 0 \quad i \neq 0$

The property that the prediction error is a white noise is exploited in evaluating the design, especially in the choice of the delay d.

B. The mean of the prediction error is zero if the filter order is equal or grater than the order of the speech model

Taking the expectation of $(s(k)-\hat{s}(k))$ we get

$E[s(k)-\hat{s}(k)] = E[s(k)] - \sum_{i=0}^{M-1} w_i(k)E[y(k-i-d)]$

Since v(k) is a zero mean random variable

$E[s(k)-\hat{s}(k)] = E\left[s(k) - \sum_{i=0}^{M-1} w_i(k)s(k-i-d)\right]$

If the speech waveform, s(k) satisfies the difference equation model

$s(k) = \sum_{i=0}^{M-1} w_i(k)s(k-i-d) + u(k) \quad where \quad E[u(k)] = 0$

Substituting for s(k) we get

$E[s(k)-\hat{s}(k)] = 0$

Thus if the order of the FIR filter is chosen large enough, the mean of the prediction error will be zero.

2.2 Least Mean-Square Algorithm

An LMS algorithm is employed for weight update. It is derived from the steepest-descent method by replacing the expectation operation by instantaneous value

$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)\mathbf{y}(k)e(k)$

where μ(k) is a positive scalar, referred t as a learning factor (see Fig. 1.). To ensure stability of the LMS algorithm, μ(k), chosen to satisfy the following inequality

$0 < \mu(k) < \frac{2}{p(k)} \quad where \quad p(k) = \sum_{i=0}^{M-1} y^2(k-i)$

The term, p(k), is the short time energy of the of the input, {y(k-i)} over a window of length M.

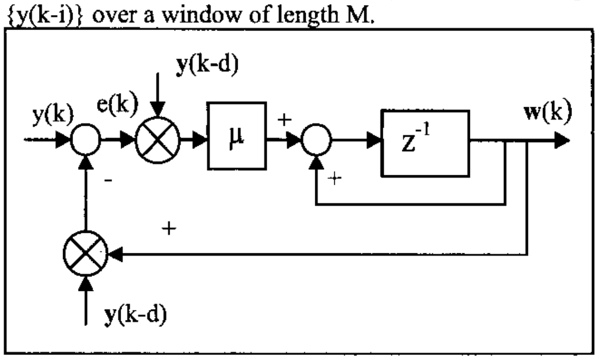

Fig. 1. LMS-based adaptive filter algorithm

There is a trade-off between the speed of convergence and the covariance of the weight estimates. Let **C**(n) be the covariance of **w**(k). It is shown in [4]

$$\mathbf{C}(k+1) = (I - \mu\mathbf{R})\mathbf{C}(k)(I - \mu R) + \mu^2 \mathbf{R} J_{\min}$$

where **R** is the correlation matrix and J_{min} is optimal error. The last term on the right hand side of the equation prevents C(k) = 0 as a solution. Accordingly the covariance C(k) is prevented from going to zero by this small term. In particular the estimation error approaches zero but then executes small fluctuation. Ideally the minimum mean - square error J_{min} is realized when the coefficient vector **w**(n) of the transversal filter approaches the optimal value $\mathbf{w}_0$, defined by the normal equation. The steepest descent algorithm does realize this idealized condition the exact measurements of R and p are used. On the other hand LMS algorithm relies on a noisy estimate of the gradient with the result the tap - weight vector **w**(k) approaches $\mathbf{w}_0$ after a large number of iterations and the executes a fluctuation about $\mathbf{w}_0$.

2.2 Computational complexity

The computational tasks of an LMS algorithm are

- The computation of the error, e(k)

$$e(k) = y(k) - \mathbf{w}^T(k)\mathbf{y}(k-d)$$

- The computation of the learning factor, μ

$$0 < \mu(k) < \frac{2}{\sum_{i=0}^{M-1} y^2(k-i)}$$

- Weight update

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)\mathbf{y}(k)e(k)$$

The learning factor is adaptive given by

$$\mu(k+1) = \alpha\mu(k) + k(1-\alpha)y^2(k), \quad 0 \le \alpha \le 1$$

The number of multiplications, divisions, additions and subtractions of the LMS algorithm per iteration are listed in the following table.

Table 1. Number of multiplications/additions

Task	*Multiply*	*Add*
Error computation	M	M
learning rate	3	1
Weight update	2M	M

Thus there are 3M+3 multiplications and 2M+1 additions. The computational complexity, is therefore of the order, M, denoted, O(M)

3. IMPLEMENTATION OF ALGORITHM

The adaptive speech enhancement algorithm was implemented on a PC using MATLAB/Simulink real time workshop. Because of limitations in the I/O board and the target computer we were constrained to sample the speech signal at 2 KHz, which is sufficient to demonstrate the speech enhancing capability.

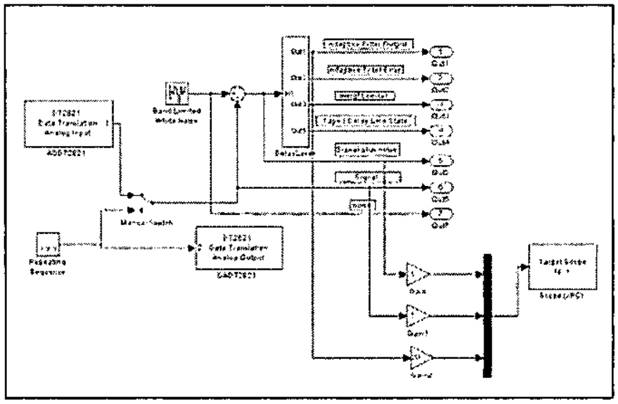

Fig. 2. Simulink model for real time workshop

Figs. 2, and 3 show how the blocks for target PC were incorporated into the offline simulink diagram. The above figure is a subsystem within the simulink model. This allows mask variables to be used as values specified within the realtime blocks (for instance the sampling rate). The switch shown is a manual switch. It changes position when the user double clicks on the switch. In one switch configuration the simulink model runs off line, and with the other switch configuration the simulink model reads data from the analog interface in realtime. A script file was developed to obtain the simulink model for any given filter order as shown in Fig. 3.

Fig. 3. FIR filter diagram developed from a script file

The adaptive filter was evaluated using noisy speech under different noise background. The FIR filter order, the learning rate and the prediction horizon were determined by iterative design and implementation procedure. The FIR filter order was M=30, the prediction horizon was 0.5 msec. The learning factor, μ, is adaptive using the energy estimate shown in Fig. 2.

Fig.4. shows the waveforms of the clean, noisy and the enhance speech obtained by attenuating the noise and enhancing the speech.

The adaptive filter is capable of adjusting the coefficients to not only different spectral characteristics of the speech but also to the spectral characteristics of the noise. See Fig.5. This ability is exploited to detect the voice activity from the portion containing only the background noise. The noisy speech and the noise only segments are detected using a simple scheme based on thresholding the energy of the adaptive filter output is used.

$$g(k) = ag(k-1) + (1-a)\hat{s}^2(k) \quad 0 \le a \le 1$$

where g(k) is the filtered energy of $\hat{s}(k)$

The speech-noise transition function, d(k) is determined

$$d(n) = \begin{cases} 0 & g(k) < Th \Rightarrow noise\ only \\ 1 & g(k) \ge Th \Rightarrow noisy\ speech \end{cases}$$

where Th is a threshold value.

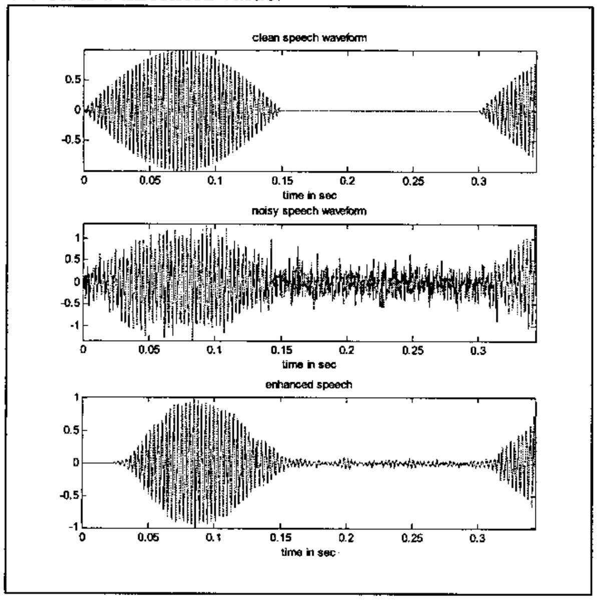

Fig. 4. Plots of the clean , noisy and the filtered speech

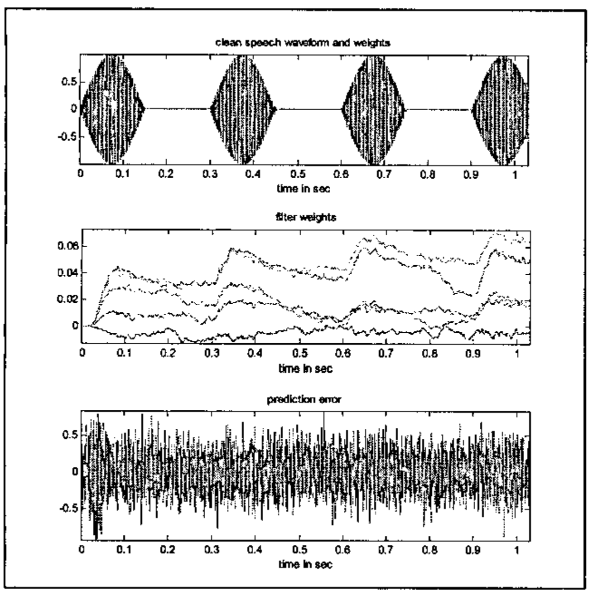

Fig. 5. The clean, filter weights and the noisy speech

The energy g(k) is compared with a threshold to detect the noisy speech and noise only segments. When a noise-only segment is detected, the waveform is gradually attenuated. Similarly when the noisy speech segment is detected, waveform is gradually accentuated. This will ensure a smooth transition between segments so that the speech is pleasant to hear. The performance of the detection and speech enhancement scheme are shown in Fig.6.

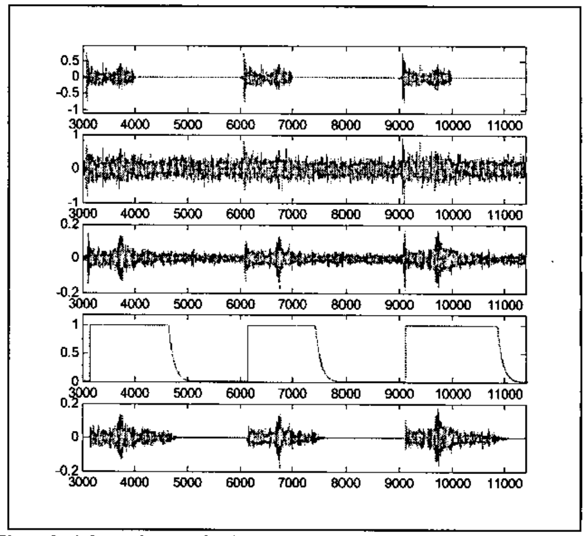

Fig. 6. The plots of clean speech, noisy speech, filtered speech, attenuation function, and enhanced speech.

5. CONCLUSIONS

A simple adaptive filter using LMS algorithm was found to be effective in enhancing speech waveform. The computational complexity is of the order of the filter order. The adaptive algorithm for learning rate ensures convergence. The adaptive filter has an ability to detect silence and voice activity portion of the noise speech. This is useful for increasing the intelligibility and to conserve bandwidth.

References

[1] M.Sambur, " Adaptive noise cancelling for speech signals" IEEE *Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-26, No.5, pp.419-423,1978

[2] J.Kim, and C.Un, " Enhancement of noisy speech by forward/backward adaptive filtering," *Proc. ICASSP*, Tokyo, Japan, volq.1, pp.419-423, 1978.

[3] M.B. Yeary and P.C. Loizou, " Adaptive filtering for speech enhancement, ".

[4] S. Haykin, "Adaptive filter theory," Prentice-Hall Inc, 1996

[5] J.R. Deller, J.G. Proakis and J.H.L. Hansen," Discrete-time processing of speech signals", Prentice Hall, 1987

[6] T. Quartieri, "Discrete-time speech signal processing", Prentice Hall, 2002.