Acoustic and line echo cancellation using adaptive filters

Vladimir M. Matić and Srđan N. Abadžić

Abstract – **In this work we present an echo canceller design using an adaptive filter theory. We compare conventional adaptive filters NLMS (*Normalized Least Mean Square*) and RLS (*Recursive Least Squares*), as well as more sophisticated block adaptive filters BFDAF (*Block Frequency Domain Adaptive Filter*) and MDBFDAF (*Multidelay Block Frequency Domain Adaptive Filter*). The goal of this work is to choose an optimal algorithm for cancelling both acoustic and line echo.**

Key words – **Adaptive filters, BFDAF, Echo cancellation, MDBFDAF, NLMS, RLS.**

I. Introduction

WE meet echo in our everyday life. Echo phenomena are interesting and entertaining, but their presence in communication networks are undesirable and represent a serious problem. Echo is delayed and degraded version of original signal which travels back to its source after several reflections or because of some other reason. Nature of echo signal can be either acoustic or electrical, and in order to reduce its undesired effect we employ echo cancellers. Design of echo cancellers requires an application of adaptive filter theory. Echo cancellers must work within specific time limits so adaptive algorithms must provide fast convergence of filter parameters. We've been applying echo cancellers successfully for many years, but we always tend to improve them and increase their efficiency.

II. Types of echo origin

When we speak about echo we distinguish acoustic echo (in transmission of speech signal), line and local echo (in data or speech signal transmission). Acoustic echo is usually present at hands free communication, which is used in cars to provide safety for drivers. It is also present during conversation while we use the loudspeaker telephone. A sound wave, which is radiated from a speaker, reflects from obstacles in the surrounding area (walls, ceilings etc.) and time–delayed signal, after several reflections, finally reaches the microphone. In that way, a person who has pronounced something will be able to hear an attenuated and echoic copy of his/her speech. This is an acoustic echo. Opposite to the acoustic echo is line echo which appears during the transmission of data as well

Vladimir M. Matić, School of Electrical Engineering, University of Belgrade (e-mail: MaticVL@yahoo.com).
Srđan N. Abadžić, School of Electrical Engineering, University of Belgrade (e-mail: srdjan_abadzic@yahoo.com).

during the transmission of speech signal. This effect is caused by the non–ideal behavior of hybrids due to improper impedance matching. The hybrid is a device which should provide the transition from a two–wire segment to the four–wire segment and vice versa. It should provide the separation of the receiving and transmitting part of the signal. In ideal case, hybrid should pass the receiving signal attenuating it by 3dB, and also not allowing its passage in the opposite direction.

Time delay in the acoustic echo is more prominent than in the line echo, so echo cancellation requires the application of adaptive filters of higher order.
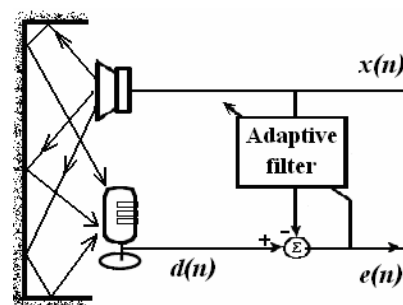

Fig. 1 Adaptive filtering of acoustic echo

Presently, the most efficient way of echo cancellation demands the usage of adaptive filters [1]. All adaptive filters have the ability to model an echo path very successfully. They estimate the filter parameters and replicate the echo signal. After generating the replica of echo signal, it is subtracted from the receiving signal. Aim of this is to avoid an echo signal from a system and to preserve only a desired part of the signal. The greatest advantage of adaptive filters is their ability to adapt filter parameters with small amount of previous knowledge about characteristics of echo path. The main objective is to monitor the changes which appear during the communication.

Figure 1 shows the position of an adaptive filter in the process of an acoustic echo cancellation. During line echo cancellation it is necessary to place a filter in a four–wire segment in transmission line, very close to the hybrid.

III. Adaptive filtering

Adaptive filters consist of two parts. The first one is a digital filter which estimates an echo path. An adaptive algorithm represents the other part and its purpose is to

update parameters of a digital filter periodically. The updates should occur fast enough in order to track the changes at the echo path. When modeling an echo path we mostly use FIR filters [1]. The most popular adaptive algorithms are NLMS and RLS. Their performance are evaluated and compared by the usage of criteria function.

A. NLMS algorithm

The criteria function for Normalized Least Mean Square (NLMS) algorithm is:

$$J(k) = e^2(k). \qquad (3.1)$$

Gradient of such defined function can be expressed as:

$$\hat{\nabla}(k) = -2e(k)\mathbf{X}(k). \qquad (3.2)$$

The error signal $e(k)$ is equal to the difference $e(k)=d(k)-y(k)=d(k)-\mathbf{X}(k)^T w(k)$, where $d(k)$ is a desired signal, $y(k)$ is filter output, while $X(k)$ is a vector–column of the input data $X(k)=[x(k)\ x(k–1)\ x(k–2)...x(k–M)]^T$, and $w(k)$ is a vector–column of the estimated parameters of the FIR filter at the instant k. Deploying the equation (3.2), we finally get the NLMS adaptive algorithm:

$$w_i(k+1) = w_i(k) + \frac{\alpha\left[e(k)x(k-i)\right]}{\sum_{j=0}^{N} x^2(k-j)}, \ i = 0,1,..N, \qquad (3.3)$$

where $N+1$ represents the order of a FIR filter, and α is a coefficient whose values are within the range of $0<\alpha<2$.

B. RLS algorithm

The Recursive Least Squares (RLS) algorithm uses criteria function defined as:

$$J(k) = \frac{1}{2}\sum_{i=0}^{k} e^2(i). \qquad (3.4)$$

The similarity of (3.1) and (3.4) is easily noticed. We also may anticipate that RLS algorithm requires high computational complexity. RLS algorithm is defined as:

$$\mathbf{H}(k+1) = \mathbf{H}(k) + \\ \mathbf{K}(k+1)\left[d(k+1) - \mathbf{X}^T(k+1)\mathbf{H}(k)\right]. \qquad (3.5)$$

Matrix $\mathbf{K}$ is defined as following:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1)\mathbf{X}(k+1) \\ = \mathbf{P}(k)\mathbf{X}(k+1)\left[1 + \mathbf{X}^T(k+1)\mathbf{P}(k)\mathbf{X}(k+1)\right]^{-1} \qquad (3.6)$$

where variables $w(k)$, $d(k)$ i $X(k)$ are defined similarly as in the NLMS algorithm, and $P(k)=[Z^T(k)\ Z(k)]^{-1}$, while $Z(k)$ is:

$$Z(k) = \begin{bmatrix} x(0) & 0 & 0 & \cdots & 0 \\ x(1) & x(0) & 0 & \cdots & 0 \\ x(2) & x(1) & x(0) & \cdots & 0 \\ & & & \ddots & \vdots \\ x(k) & x(k-1) & x(k-2) & \cdots & x(k-M) \end{bmatrix}$$

In spite of the great similarity we do not use RLS algorithm in cancelling acoustic echo because it cannot achieve a computational efficiency which is required for adaptive filters with lengthy impulse response.

IV. BLOCK FREQUENCY DOMAIN ADAPTIVE FILTERING

A. Block Frequency Domain Adaptive Filtering

Conventional adaptive algorithms (NLMS and RLS) adapt filter parameters on every single input data point. In order to reduce the computational complexity, one existing method is to perform Block Frequency Domain Adaptive Filtering (BFDAF) with a block updating schedule. This block updating scheme allows the filter output signal and the adaptive tap weight to be computed in the frequency domain instead of the time domain. Hence, time domain convolution and correlation operations are computed in the frequency domain using the advantages of the Fast Fourier Transform (FFT) [2]. However, there are two major drawbacks that come with block frequency domain adaptive filters. First, the inherent propagation delay involved for collecting blocks of data before processing. Secondly, the decrease in convergence speed causes by the block updating schedule. To reduce the inherent delay and also to improve the convergence speed, the Multidelay Block Frequency Domain Adaptive Filter (MDBFDAF) will be introduced in part B.

BFDAF is based on the LMS algorithm. Filter parameters are updated periodically after N sampling periods, so we introduce new block time–discrete index which is represented by m. The estimation is done directly in the frequency domain:

$$\mathbf{W}(m+1) = \mathbf{W}(m) + 2\frac{\mu_B}{N}\mathbf{Fg}\mathbf{F}^{-1}\mathbf{X}^H(m)\mathbf{E}(m). \qquad (4.1)$$

Filter coefficients are defined as a $2N$ x 1 vector:

$$\mathbf{W}(m) = \left[\mathbf{W}_0(m)\ \mathbf{W}_1(m)\ ...\ \mathbf{W}_{2N-1}(m)\right]. \qquad (4.2)$$

Matrices $\mathbf{F}$ and $\mathbf{F}^{-1}$ represents FFT and IFFT $2N$ x $2N$ matrices:

$$\mathbf{F} = \left[w^{-kn}\right], \ \mathbf{F}^{-1} = \frac{1}{N}\left[w^{kn}\right] = \frac{1}{N}\mathbf{F}^H, \ w = e^{j\frac{2\pi}{N}}. \qquad (4.3)$$

Input data signal is stored into data blocks with an overlapping of 50%, which are used in formation of the input diagonal $2N$ x $2N$ frequency domain matrix, denoted as $\mathbf{X}(m)$. The diagonal of this matrix is created by multiplying input data block by FFT matrix. In order to present the flow of the algorithm in a simplier way we will use the following matrices:

$$\mathbf{q_F} = [\mathbf{I_N}\ \ \mathbf{0_N}], \mathbf{q_L} = [\mathbf{0_N}\ \ \mathbf{I_N}], \mathbf{g} = \begin{bmatrix} \mathbf{I_N} & \mathbf{0_N} \\ \mathbf{0_N} & \mathbf{0_N} \end{bmatrix}. \qquad (4.4)$$

Matrices consist of block unity and zero submatrices. Pre–multiplying by these matrices modify the corresponding vector by discarding the first N or the last N vector elements. They aslo provide necessary appending with zero elements. In both cases the number of affected elements is determined by the value N.

The frequency domain filter output is given by:

$$\mathbf{Y}(m) = \mathbf{X}(m)\mathbf{W}(m), \qquad (4.5)$$

and after both multiplying and performing an Inverse FFT (IFFT) we get:

$$y(m) = \mathbf{q_L}\mathbf{F}^{-1}\mathbf{Y}(m). \qquad (4.6)$$

The time domain error vector is computed as the difference between the desired and filter output signals:

$$e(m) = d(m) - y(m). \qquad (4.7)$$

Then, the error vector can be transformed to the frequency domain according to:

$$\mathbf{E}(m) = \mathbf{F}\mathbf{q}_L^T e(m). \qquad (4.8)$$

The block frequency domain instantaneous gradient estimate is an N x 1 vector and is given by:

$$\underline{\nabla}(m) = -\frac{2}{N} q_F \mathbf{F}^{-1} \underline{\mathbf{X}}^H(m) \underline{\mathbf{E}}(m). \qquad (4.9)$$

The convergence of the BFDAF algorithm can be improved by using the Self–Orthogonalization algorithm. It is well known that the eigen spread of the input autocorrelation matrix directly affects the convergence speed of adaptive algorithms in the LMS family [2]. BFDAF algorithm can be modified and written as:

$$\underline{\mathbf{W}}(m+1) = \underline{\mathbf{W}}(m) + 2\mathbf{F}\mathbf{g}\mathbf{F}^{-1}\boldsymbol{\mu}(m)\mathbf{X}^H(m)\underline{\mathbf{E}}(m) \qquad (4.10)$$

$$\boldsymbol{\mu}(m) = \alpha \mathbf{R}^{-1}(m) \qquad (4.11)$$

$$\mathbf{R}^{-1}(m) = diag\left\{ \left[P_0^{-1}(m) \quad \dots \quad P_{2N-1}^{-1}(m) \right]^T \right\} \qquad (4.12)$$

Fig. 3 Partition of adaptive filter on sub–filters

Sub–filter input signals represent delayed replicas of the original input signal. Hence, transportational delay is reduced. Each of these sub–filters are based on the BFDAF which is previously described. Implementation of the MDBFDAF (Figure 4) is extension of the BFDAF structure and replacement of value N with N_s.

Fig. 2 Block Frequency Domain Adaptive Filter

Matrix $\mathbf{R}$ presents an estimate of the autocorrelation matrix, where $P_k(m)$ is the estimated power in the kth frequency bin at block m. The corresponding updating of the diagonalized input autocorrelation matrix is:

$$\mathbf{R}(m) = \beta \mathbf{R}(m-1) + (1-\beta)\mathbf{X}^H(m)\mathbf{X}(m) \qquad (4.13)$$

BFDAF is shown as a schematic diagram in Figure 2.

B. Multidelay Block Frequency Domain Adaptive Filtering

Block Frequency Domain Adaptive Filtering achieves computational savings comparing to the time domain adaptive algorithms. But, it also involves relatively long inherent transportational delay, because the block size is usually chosen to be equal to the filter length $2N$. A scheme that partitions a long filter impulse response into shorter sections is described to reduce the propagation delay by processing the input data with a bank of relatively short parallel BFDAFs. This is depicted in Figure 3. Since each sub–BFDAF has a shorter filter length, the convergence speed is also improved. Every sub–filter has to estimate one distinctive part of the original filter impulse response. Length of adaptive filter N and number of blocks M are chosen to be the appropriate powers of number 2. In that case, each sub–filters will have the length of impulse response $N_s = N/M$.
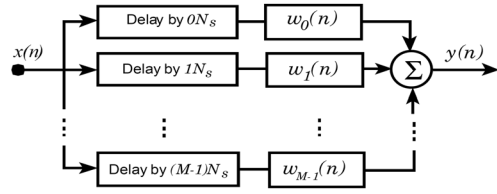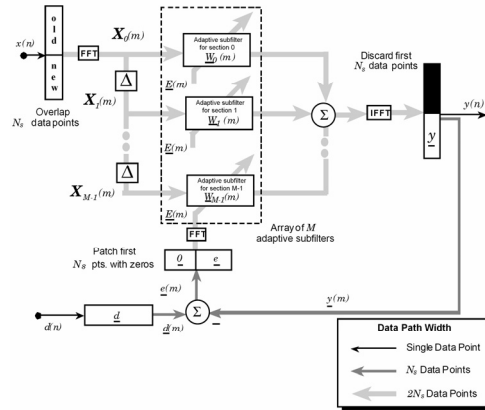

Fig. 4 MDBFDAF structure

Filter output is given as:

$$\underline{y}(m) = q_L \mathbf{F}^{-1} \sum_{p=0}^{M-1} \mathbf{X}_p^H(m) \underline{\mathbf{W}}_p(m). \qquad (4.14)$$

Each section of the MDBFDAF is computed recursivly:

$$\underline{\mathbf{W}}_p(m+1) = \underline{\mathbf{W}}_p(m) + 2\mathbf{F}\mathbf{g}\mathbf{F}^{-1}\boldsymbol{\mu}(m)\mathbf{X}_p^H(m)\underline{\mathbf{E}}(m). \qquad (4.15)$$

V. SIMULATION RESULTS

Echo path is estimated by FIR adaptive filter. Application of IIR filters may cause an unstable algorithm, which will diverge if the poles of the transfer function would be placed out of the unit circle.

The goal of simulation was to find the most efficient algorithm which would be applied in echo canceller. Algorithms were compared by their ability to cancel echo effect, by the speed of the algorithm convergence and by the computational complexity. Measure of the echo canceller efficiency was expressed by the Echo Return Loss Enhancement (*ERLE*) factor which is defined as [1]:

$$ERLE = 10\log_{10}\left\{ \frac{E\left[y^2(k)\right]}{E\left[e^2(k)\right]} \right\}. \qquad (5.1)$$

In simulations, evaluation of the *ERLE* factor was computed by arithmetical average, that is, a mathematical expectation was replaced by the short–time window estimate. Algorithms were simulated in Matlab 7.0 program. Input speech signal was sampled at the frequency of 8000 Hz.

In order to examine the efficiency of acoustic echo canceller it was necessary to use a room acoustic impulse response, which had been recorded earlier (Figure 5). The results of the applied acoustic echo cancellers are presented in Figure 6. It is also very interesting to compare error signals in time domain (Figure 7).
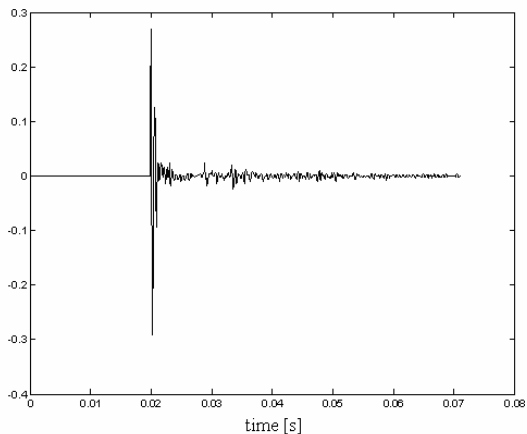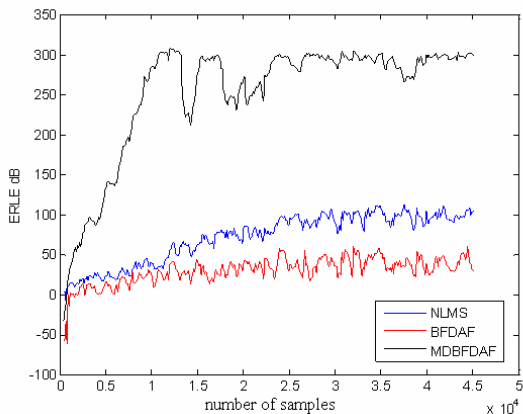

Fig. 5 Room acoustic impulse response
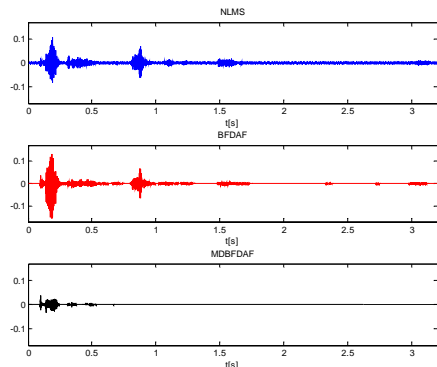

Fig. 6 *ERLE* factors in acoustic echo cancellation

Fig. 7 Error signals for different adaptive filters

Impulse response which was used to estimate echo path of line echo was artificially generated and was modeled by the lowpass FIR filter (Figure 8). Results of line echo cancellation are shown in Figure 9.
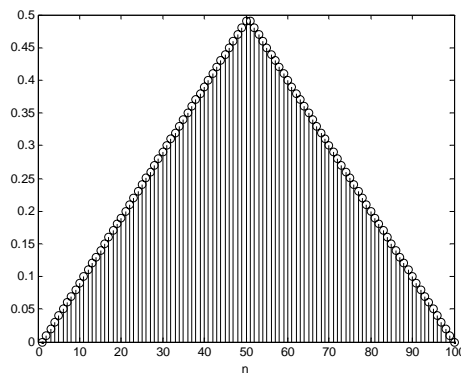
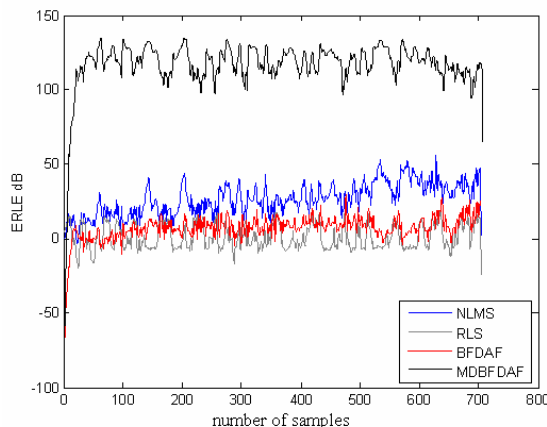

Fig. 8 Impulse response of line echo path

Fig. 9 *ERLE* factors in line echo cancellation

VI. CONCLUSION

Simulation results completely confirm theoretical assumptions and demonstrate the superiority of MDBFDAF. According to the theoretical results shown in [2] computational savings of the MDBFDAF (length of the impulse response was 4096 samples) comparing to the NLMS are 76%. This work encourages the use of the MDBFDAF in echo cancellers because of its advantages when comparing to the conventional adaptive filters. Further work may include the implementation of MDBFDAF on the Digital Signal Processor (DSP). This improvement should provide a comfortable conversation when hands free or loudspeaker communication devices are used.

LITERATURE

[1] B. Kovačević, Z. Banjac, M. Milosavljecić, *Adaptivni digitalni filtri*, Akademska misao, Beograd 2000.
[2] K. Chow, *Efficient Adaptive Acoustic Echo Cancellation, Methods and Structures*, Calgary, Alberta 1996.