BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LVII (LXI), Fasc. 2, 2011
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

ADAPTIVE FREQUENCY EQUALIZER

BY

DANIELA TĂRNICERIU*, LUCIAN TRIFINA and ELENA COJOCARIU

“Gheorghe Asachi” Technical University of Iaşi
Faculty of Electronics, Telecommunications and Information Technology

Received: September 5, 2010
Accepted for publication: December 15, 2010

Abstract. The implementation of a real-time frequency equalizer is proposed. The system will perform adaptive filtering of the original signal following an idea derived from LMS algorithm. The main difference compared to classical least mean squares (LMS) is that the adaptation technique is implemented in the frequency domain. Three different equalizing methods will be proposed, and their advantages and drawbacks will be presented.

Key words: adaptive filtering; equalization; least mean square algorithm.

1. Introduction

The characteristics of an audio signal are modified by passing through any channel, so that a modified version of the original sound will be heard. The sound will not only reach the ears directly from the source, but also through reflections from objects and walls. The signal distortion due to room acoustics can be modeled by a filtering operation. We use room impulse response (RIR) to describe the acoustical properties of a room. If someone filters the received signal with a filter that is the inverse of RIR, then he will compensate for the room introduced discrepancies and recover the original signal. This process is known as *equalization*.

* Corresponding author: *e-mail*: tarniced@etti.tuiasi.ro

In what follows we will propose a frequency equalizer method that aims at minimizing the errors between a target signal and the original filtered signal. Section **2** will present the commonly used equalization techniques, and section **3** will describe the theoretical background of the implemented equalizer. Experimental results will be shown in section **4** and section **5** will conclude the paper.

2. Adaptive Equalizers Overview

If the communication channel is known and static, then we can easily filter the received signal with the inverse channel filter and obtain the original signal. In practice, the channel response is not constant in time. To cope with this we need to use an equalization type that periodically updates the filter coefficients in order to track a time-varying communication channel. This method is known as *adaptive equalization*. Periodic adjustments are accomplished by periodically transmitting a short training sequence of digital data known by the receiver and the filter updates its coefficients based on this sequence. The filter coefficients can also be updated based on the signal statistics or by trying to minimize a certain error value. Fig. 1 presents the basic scheme of an adaptive equalizer. Signal x is filtered with the filter $\boldsymbol{f}$ to match the target signal, d. The adaptation algorithm has as inputs, x, d and the error, e, and acts to minimize this error.

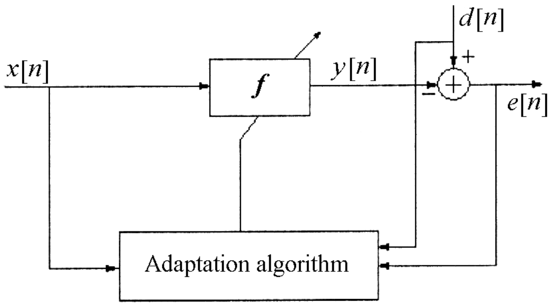


Fig. 1 – Block diagram of an adaptive filter.

If we consider the static case for Fig. 1, we can easily find the ideal value of filter coefficients by using the equation (Manolakis *et al.*, 2002)

$$\boldsymbol{f} = \boldsymbol{R}_x^{-1}\boldsymbol{r}_{dx}, \tag{1}$$

where $\boldsymbol{R}_x$ is the correlation matrix of the tap-input vector, $\boldsymbol{x}[n]$, and $\boldsymbol{r}_{dx}$ is the crosscorrelation vector between the tap-input vector, $\boldsymbol{x}[n]$ and the desired response, $d[n]$, of a certain length.

Finding $\boldsymbol{f}$ needs the inversion of the matrix $\boldsymbol{R}_x$, which can be unpractical in many cases because of the high amount of computations required.

Another way to address this problem is to use iterative algorithms that starts from an initial guess of the optimal filter and converges to a solution that is close to the optimal one. So the following equation may be obtained (Haykin, 2002):

$$\boldsymbol{f}_{n+1} = \boldsymbol{f}_n + m\,\boldsymbol{p}\,. \qquad (2)$$

The vector $\boldsymbol{p}$ is an update direction vector that we should choose adequately, along with a positive scalar, μ. The scalar, μ, is called the ***step-size*** parameter since it affects how small or how large the correction term is. $\boldsymbol{n}$ represents the current time index.

In order to guarantee convergence of $\boldsymbol{f}_n$ to $\boldsymbol{f}$, certain conditions must be imposed on the step size namely (Haykin, 2002).

$$\boldsymbol{p} = \boldsymbol{r}_{dx} + \boldsymbol{R}_x\,\boldsymbol{f}_n\,, \qquad (3)$$

$$0 < m < \frac{2}{l_{\max}}\,, \qquad (4)$$

where $\lambda_{\max}$ is the highest eigenvalue of correlation matrix $\boldsymbol{R}_x$. Except from computation bothersome we need to know the exact signal statistics (auto and cross correlation functions). But we can use stochastic gradient algorithms that have learning mechanism, which enables them to estimate the required statistics. Moreover, these methods present a tracking mechanism that allows them to follow the variations in the signal statistics and makes them applicable to the non-stationary case. Least mean squares (LMS) algorithm is one of the branches of stochastic gradient algorithms.

In this method we estimate the correlation, $\boldsymbol{R}_{x,n+1}$, and cross-correlation, $\boldsymbol{r}_{dx,\,n+1}$, by observing processes $x[n]$ and $d[n]$ and replacing the statistic averages by time averages. That is, we take

$$\boldsymbol{R}_{x,n+1} = x_{n+1}x_{n+1}^{H}\,, \qquad (5)$$

$$\boldsymbol{r}_{dx,n+1} = d[n+1]x_{n+1}\,, \qquad (6)$$

where H stands for Hermitian.

We can find $\boldsymbol{p}_n$ using following relation:

$$\boldsymbol{p}_n \approx x_{n+1}\left[d[n+1] - x_{n+1}^{H}\boldsymbol{f}\right]. \qquad (7)$$

Afterwards, $\boldsymbol{f}_{n+1}$ can be found by applying expression

$$\boldsymbol{f}_{n+1} = \boldsymbol{f}_n + m x_{n+1}\left[d\left[n+1\right] - x_{n+1}^{H}\boldsymbol{f}_n\right]. \qquad (8)$$

If the statistics of the signal are changing slowly with respect to the sampling frequency, then we can have a better approximation for $\boldsymbol{p}_\text{n}$ and use it as a function of x_n and $d[n]$ instead of x_{n+1} and $d[n+1]$. The result for slow changing signal, $\boldsymbol{f}_{n+1}$, is

$$\boldsymbol{f}_{n+1} = \boldsymbol{f}_n + m x_n \left[d[n] - x_n^H \boldsymbol{f}_n \right] \tag{9}$$

and $\boldsymbol{f}_0$ is an initial guess.

3. Frequency Domain Equalization

Algorithms similar to LMS can also be implemented in the frequency domain. The equalizers of this type have programmable taps for several frequency bands that adjust magnitude and phase of a received signal to a desired one. An adaptive frequency equalization system consists of the following components: a hard decision circuit configured to select ideal values using equalized information; a frequency response circuit configured to determine frequency response update values using ideal values and the received signal; an adjust circuit configured to update stored frequency response information using frequency response update values during a transmitted frame, and to update programmable equalizer taps using stored frequency response information. The transition from the time domain to the frequency domain is usually realized with help of the Fourier transform.

Often it is desired that an audio signal be modified as a function of time and frequency. A way of doing this is to divide a signal into blocks and to apply a Discrete Fourier Transform (DFT) to each block. But the use of this method has a side effect, that is discontinuities at block boundaries. To alleviate this problem instead of hard boundaries, overlapping windowing is used to separate signal blocks. This type of frame-wise DFT processing is called Short Time Fourier Transform (STFT) (Vaidyanathan, 1993).

Like standard LMS, the equalizer proposed in this paper aims at minimizing the expected mean square error (MSE) between a desired output signal, $d[n]$, and the actual output, $y[n]$. The y signal is obtained from the input $x[n]$ by filtering with an adaptive filter, while d is x filtered with an unknown filter (which can model a room impulse response for example). Firstly, the time signal will be divided into overlapped frames (in order to compute the STFT). Fig. 2 presents the basic model of this adaptive filter implementation. The innovation compared to LMS is that the values used for error minimization are situated in frequency-domain (Fernandes *et al.*, 2004) (in the following, capital letters will denote frequency-domain values, while index n will be used for time and k for frequency). After computing the spectrum of each frame (consisting of 1024 values) we will divide the frequency values in 16 different sub-bands (Pinto *et al.*, 2006; Kahrs *et al.*, 1998), covering the range 0…$Fs/2$ (where Fs is the sampling frequency). Because the human audio system has different sensitivities for different frequency values, the number of samples will not be the same for all sub-bands. Table 1 presents the number of samples per sub-band, the minimum and maximum sub-band frequencies and the indexes of the sub-band samples (for *Fs* = 44,100 samples/s).

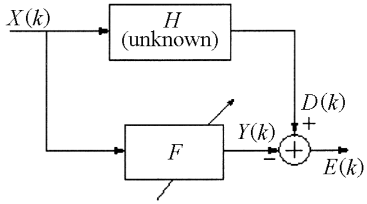


Fig. 2 – Adaptive filter model.

Table 1

Sample Distribution for Each Frequency Sub-Band (*Fs* = 44,100 Hz)

Sub-band	F_{min} Hz	F_{max} Hz	Bandwidth Hz	Number of samples	Lowest bin	Highest bin
1	0	172	172.3	4	1	4
2	172	345	172.3	4	5	8
3	345	517	172.3	4	9	12
4	517	689	172.3	4	13	16
5	689	947	258.4	6	17	22
6	947	1,292	344.5	8	23	30
7	1,292	1,723	430.7	10	31	40
8	1,723	2,326	602.9	14	41	54
9	2,326	3,101	775.2	18	55	72
10	3,101	4,048	947.5	22	73	94
11	4,048	5,599	1,550.4	36	95	130
12	5,599	7,494	1,894.9	44	131	174
13	7,494	9,905	2,411.7	56	175	230
14	9,905	13,006	3,100.8	72	231	302
15	13,006	16,968	3,962.1	92	303	394
16	16,968	22,050	5,081.8	118	395	512

The 512 samples corresponding to the negative part of the spectrum are similarly grouped.

LMS aims at minimizing a cost function, *J*, which in our case will be

$$J = E\left\{\left|E(k)\right|^2\right\};\ E(k) = D(k) - Y(k)\,. \tag{10}$$

An equivalent expression will be obtained by splitting the value among the 16 sub-bands

$$J = \sum_{l=1}^{16} E\left\{\left|D(k_l) - Y(k_l)\right|^2\right\},\ k_l = \left\{k \left| k_{l_{\min}} \le k \le k_{l_{\max}} \right.\right\}. \tag{11}$$

$k_{l_{\min}}$ and $k_{l_{\max}}$ are the minimum and the maximum values of the sample indexes (last two columns in Table 1).

For each sub-band, l (with l = 1…16), we will try to find out the coefficient $w(l)$ which minimizes the total cost function. If index k belongs to sub-band l, then

$$Y(k)=w(l)X(k) \tag{12}$$

and the cost function becomes

$$J=\sum_{l=1}^{16}E\left\{\left|D\left(k_l\right)-w(l)X\left(k_l\right)\right|^2\right\},\ k_l=\left\{k\left|k_{l_{\min}}\le k\le k_{l_{\max}}\right.\right\}. \tag{13}$$

3.1. Method 1: Real Coefficients, Using the Complex Values of X and D (M1)

The values of $w(l)$ will be determined to minimize J

$$\begin{aligned}\frac{\partial J}{\partial w(l)}&=2w(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}-E_{k_l}\left\{X^*(k)D(k)\right\}-E_{k_l}\left\{X(k)D^*(k)\right\}=\\&=2w(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}-2\Re e\left(E_{k_l}\left\{X^*(k)D(k)\right\}\right).\end{aligned} \tag{14}$$

Because the derivative of J with respect to $w(l)$ does not depend on other sub-band coefficients, we will compute $w(l)$ for each sub-band independently.

To each coefficient will be assigned an initial value, and will be successively updated. Given its value, $w_n(l)$, the following value will be determined:

$$w_{n+1}(l)=w_n(l)-\frac{1}{2}m\frac{\partial J}{\partial w_n(l)}, \tag{15}$$

where $\partial J/\partial w_n(l)$ represents the update direction and μ the step size. The value of μ will be computed such as the error at iteration $n+1$ is lower than at iteration n for each sub-band, $E_{n+1}<E_n$, and consequently

$$\begin{aligned}&E_{k_l}\left\{\left|D(k)\right|^2\right\}+w_{n+1}^2(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}-w_{n+1}(l)E_{k_l}\left\{X^*(k)D(k)+X(k)D^*(k)\right\}<\\&<E_{k_l}\left\{\left|D(k)\right|^2\right\}+w_n^2(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}-w_n(l)E_{k_l}\left\{X^*(k)D(k)+X(k)D^*(k)\right\}\end{aligned}$$

i.e.

$$\begin{aligned}&w_{n+1}^2(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}-w_{n+1}(l)E_{k_l}\left\{2\Re e\left(X^*(k)D(k)\right)\right\}<\\&<w_n^2(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}-w_n(l)E_{k_l}\left\{2\Re e\left(X^*(k)D(k)\right)\right\}\end{aligned} \tag{16}$$

or

$$\left[w_{n+1}(l)+w_n(l)\right]E_{k_l}\left\{\left|X(k)\right|^2\right\}<2E_{k_l}\left\{\Re e\left[X^*(k)D(k)\right]\right\}.$$

By substituting (15) into (16) the maximum step size

$$m_{\max} = \frac{\dfrac{2E_{k_l}\left\{\Re e\left(X^*(k)D(k)\right)\right\}}{E_{k_l}\left\{\left|X(k)\right|^2\right\}} - 2w_n(l)}{E_{k_l}\left\{\Re e\left(X^*(k)D(k)\right)\right\} - w_n(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}} \tag{17}$$

is obtained.

Because the sets used for coefficient updating ($X(k)$ and $D(k)$) are finite, we will use the following estimates for the expectations:

$$E_{k_l}\left\{X\left(k_l\right)\right\} = \frac{1}{N}\sum_{k_l} X\left(k_l\right);\; N = k_{l_{\max}} - k_{l_{\min}} + 1. \tag{18}$$

Having in view that the signal is correlated, we will only do one iteration per frame. Moreover, the initial value for each frame will be the value obtained for the previous frame. As a result, the coefficients will be updated only once for each 1,024 samples (similar results are obtained if we update the coefficients every two or three frames). This implementation allows for few computations and is well suited for real-time implementation.

After updating $w(l)$ we will compute $Y(k)$ and perform an inverse STFT to return in time domain. The obtained samples will be stored in a buffer from which they will be played.

The main drawback of the above mentioned method is that the w coefficients are real and they cannot compensate for phase difference between X and D (complex signals). As a result, there will be high differences between the reconstructed signal, Y, and the target one in the frequency bands where phase shifts occur in the unknown filter (several figures describing this will be shown in the section **4** of this paper). One way to compensate this discrepancy is to use complex values for w.

3.2. Method 2: Complex Coefficients, Using the Complex Values of *X* and *D* (M2)

Considering (Haykin, 2002)

$$w(l) = a(l) + \mathrm{j}b(l), \tag{19}$$

It results that

$$\frac{\partial J}{\partial w_n(l)} = \frac{\partial J}{\partial a_n(l)} + \mathrm{j}\frac{\partial J}{\partial b_n(l)} = -2E_{k_l}\left\{X^*(k)D(k)\right\} + 2w(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}; \tag{20}$$

the max-step-size is

$$m_{\max} = \frac{\dfrac{2E_{k_l}\left\{X^*(k)D(k) + X(k)D^*(k)\right\}}{E_{k_l}\left\{\left|X(k)\right|^2\right\}} - 2w_n(l)}{E_{k_l}\left\{X^*(k)D(k)\right\} - w_n(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}}. \quad (21)$$

The price that has to be paid for the more accurate adaptive filter response is an increase in complexity, because real additions and multiplications are replaced with complex ones.

3.3. Method 3: Real Coefficients, Using the Absolute Values of *X* and *D* (M3)

Another way to determine the adaptive filter is to use only the absolute values of *X* and *D* (therefore the phase difference will not influence the adaptive filter coefficients). In this case the update direction will become

$$\frac{\partial J}{\partial w_n(l)} = 2w(l)E_{k_l}\left\{\left|X(k)\right|^2\right\} - 2E_{k_l}\left\{\left|X(k)\right|\left|D(k)\right|\right\} \quad (22)$$

and the maximum step size is

$$m_{\max} = \frac{\dfrac{2E_{k_l}\left\{\left|X(k)\right|\left|D(k)\right|\right\}}{E_{k_l}\left\{\left|X(k)\right|^2\right\}} - 2w_n(l)}{E_{k_l}\left\{\left|X(k)\right|\left|D(k)\right|\right\} - w_n(l)E_{k_l}\left\{\left|X(k)\right|^2\right\}}. \quad (23)$$

In all cases the step size will be chosen as $t\mu_{\max}$, where $t \in (0, 1)$.

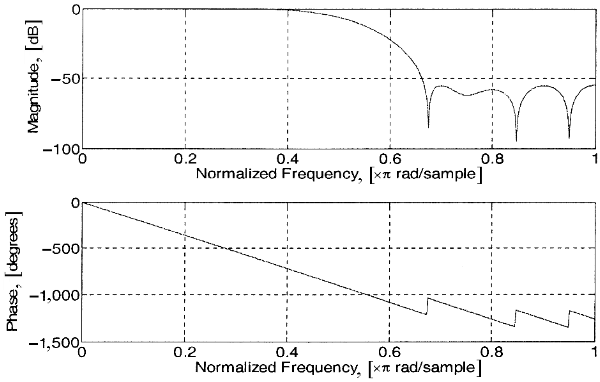
4. Simulation Results

4.1. Simulation Setup

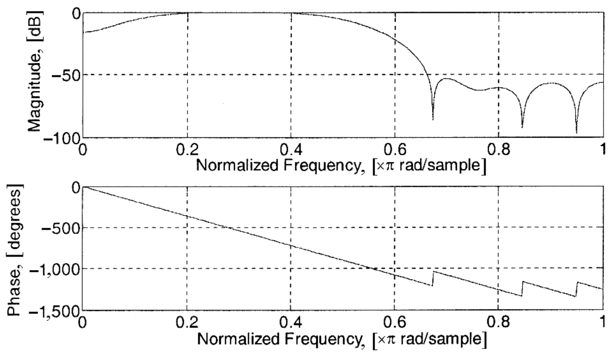
The proposed frequency equalizer was implemented in Matlab. The original signal was read from a ".wav" file. The target signal was obtained from the original filtered by a FIR filter. Three filter types were used, with the frequency response (magnitude and phase) given in Fig. 3.

A windowed overlapped STFT was applied to the original and the target signals. The used window is a Kaiser-Bessel window (Oppenheim *et al.*, 1999), and its shape depends on an input parameter (the higher the parameter value is, the larger the flat area in the center of the window is). The synthesis window is identical to the analysis one.

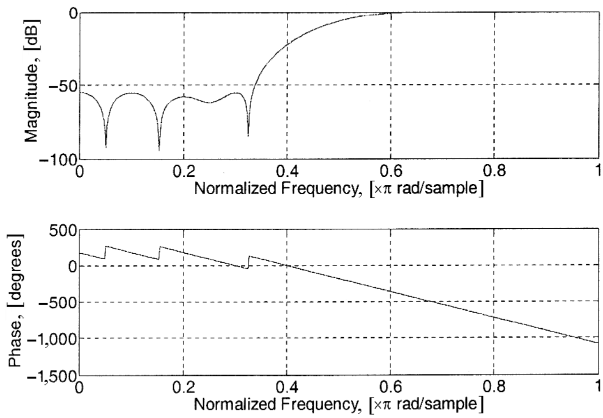
Afterwards, the equalization was done in frequency domain by the three methods presented in section **3**. The equalization will be done in real-time, ***i.e.*** while a frame is being equalized, the previous ones are already being played.



a



b



c

Fig. 3 – Low pass filter (*a*); band pass filter (*b*); high pass filter (*c*).

4.2. Results

The Figs. 4,…,6 present the adaptive filter response for the low-pass, band-pass and high-pass approximations, respectively, for each of the approximation methods (M1: real coefficients, using the complex values of *X* and *D*; M2: complex coefficients, using the complex values of *X* and *D*; M3: real coefficients, using the absolute values of *X* and *D*). The responses are shown after the adaptive filters have converged. The quantities in the plots are: *x*-axis: Normalized Frequency, *y*-axis: Magnitude [dB].

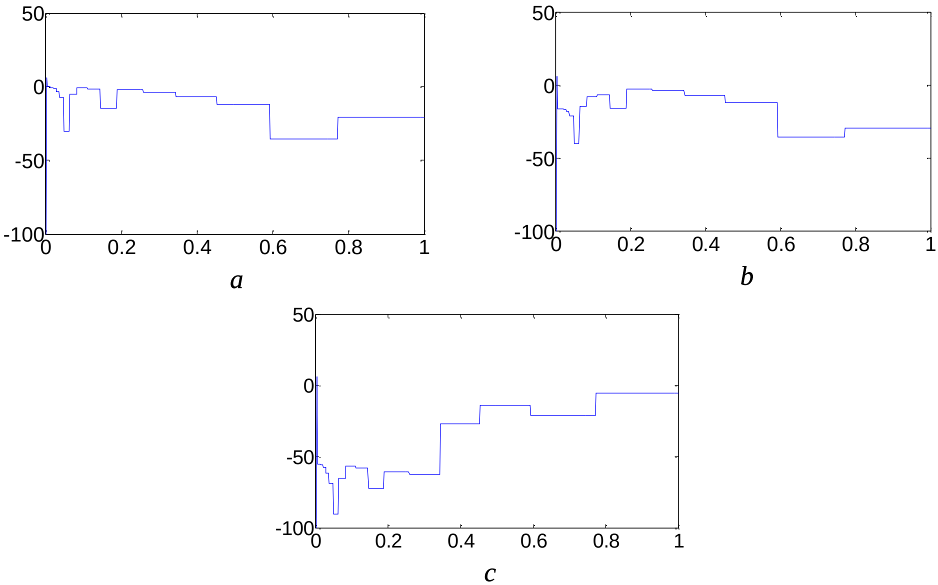


Fig. 4 – Adaptive filter responses using M1: low pass filter approximation (*a*); band pass filter approximation (*b*); high pass filter approximation (*c*).

In the low-pass and band-pass approximations we observe some important differences with respect to the target filters (from Fig. 4) in the bins containing the normalized frequencies of 0.05 and 0.15. A possible explanation for this phenomenon is that real values for the coefficients cannot compensate the phase differences between *X* and *D*. Therefore, even if the *X* and *D* values have similar amplitudes, if their phases are different, the coefficient value will be low.

Considering the Figs. 4,…,6, we can say that if we use the absolute values of *X* and *D* to compute the coefficient values, we obtain a better result than using the first method. The results are comparable to those obtained by M2, but without any increase in the required computations. We did not perceive any auditory differences in the equalized signal between the three methods.

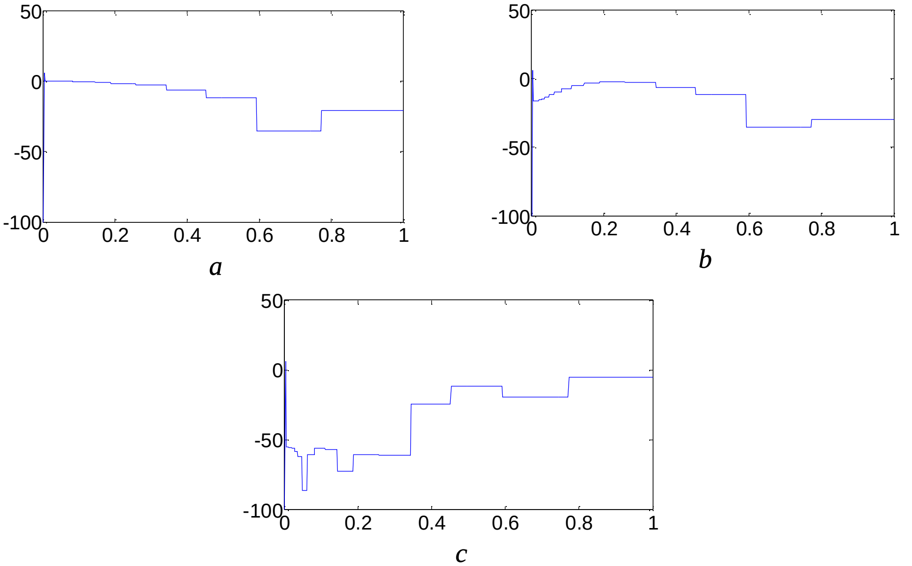


Fig. 5 – Adaptive filter responses using M2: low pass filter approximation (*a*); band pass filter approximation (*b*); high pass filter approximation (*c*).

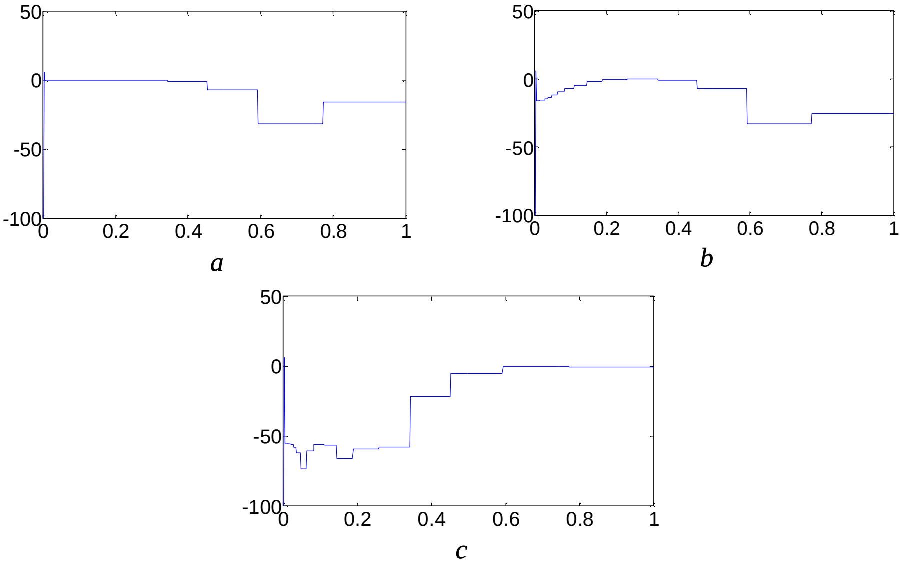


Fig. 6 – Adaptive filter responses using M3: low pass filter approximation (*a*); band pass filter approximation (*b*); high pass filter approximation (*c*).

The filter convergence time is the same for all methods, and depends on the chosen step-size. A larger value for the step-size allows for shorter convergence time, but there is a higher coefficient fluctuation when convergence is reached. Smaller step values will require more iterations to reach the convergence, but the values will be more stable. A comparison between the three equalization methods can be observed in Table 2.

Table 2
Complexity Comparison between the Three Proposed Equalization Methods

Coefficient update (operations)			$Y = wX$ (operations)		Subjective reconstruction accuracy
Case	Real	Complex	Real	Complex	
M1	$4N + 10$	0	1,024	0	AVERAGE
M2	$2N + 5$	$N + 5$	0	1,204	GOOD
M3	$4N + 10$	0	1,024	0	GOOD

The second and third columns show the number of operations required for the update of one coefficient (N is the number of samples in the frequency bins corresponding to that coefficient). Columns 4 and 5 show the number of operations required for determining the adaptive filtered signal for one frame (1,024 samples). The averaging computations were not considered.

From Table 2 we can see that the best results (in terms of both quality and complexity) are obtained when we only consider the absolute values of X and D.

5. Conclusions

A derivation of the least mean squares algorithm in the frequency domain is proposed. The used techniques allow the implementation of an efficient equalizer that takes advantage of the fact that filtering or convolution in time is equivalent to multiplication in frequency. Because the Fourier Transform of a signal can be easily computed by means of FFT, there is no difficulty to translate to the frequency domain. Based on these principles, we proposed a method that allows, for good equalization, results without requiring a large amount of computations.

REFERENCES

Fernandes G., Leitao J., Ferreira A., *Adaptive Room Equalization in the Frequency Domain*. Convention Paper, 116th AES Convention Paper, Berlin, Germany, May 2004.

Haykin S., *Adaptive Filter Theory*. 4th Ed., Prentice Hall, Upper Saddle River, NJ, USA, 2002.

Kahrs M., *Applications of Digital Signal Processing to Audio and Acoustics*. Kluwer Academic Publisher, Norwell MA, USA, 1998.

Manolakis D.G., Ingle V.K., Kogan S.M., *Statistical and Adaptive Signal Processing*. Artech House, Boston - London, 2005.

Oppenheim A.V., Schafer R.W., *Discrete-Time Signal Processing*. Prentice-Hall International, London, UK, 1999.

Pinto F., Rocha A.F., Leite A., Ferreira A.J.S., *Adaptive Audio Equalization of Rooms Based on a Technique of Transparent Insertion of Acoustic Probe Signals.* Audio Engng. Soc. Convention Paper, May 2006.
Vaidyanathan P., *Multirate Systems and Filter Banks*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1993.

EGALIZATOR ADAPTIV ÎN DOMENIUL FRECVENŢĂ

(Rezumat)

Se propune un egalizor în domeniul frecvenţă, în timp real. Sistemul realizează filtrarea adaptivă a semnalului original, pe baza unei idei ce derivă din algoritmul LMS, diferenţa faţă de acesta constând în faptul că adaptarea se obţine în domeniul frecvenţă. Sunt propuse trei tehnici de egalizare, prezentându-se avantajele şi dezavantajele fiecăreia.