Volume 56, Number 1, 2015

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

OPTIMIZED SYNTHESIS AND FPGA IMPLEMENTATION OF A FIR FILTER FOR MULTIPLE POSITION EQUALIZATION OF A RECORDING STUDIO/CONCERT HALL

Adrian Călin FĂRCAȘ Ervin SZOPOS Ioana SĂRĂCUȚ Marius NEAG Marina Dana ȚOPA
Technical University from Cluj-Napoca,
Str. G.Bariţiu nr.26-28, Cluj-Napoca, tel.0264401243, e-mail calinf@radiocluj.ro

Abstract: This paper presents an improved approach to the synthesis and implementation of digital filters employed for multiple position equalization of a room transfer function. First, we propose a more effective way of deriving the desired frequency response of the equalizer than the Levinson-Durbin algorithm employed by well-known method used here as a reference to compare against. Second, the equalizer is implemented by a FIR filter synthesized by using an optimized procedure, based on a genetic algorithm that curtails the differences between the wanted and synthesized frequency responses while minimizing the filter length. These two features help us reduce significantly the hardware requirements for implementing the resulting equalizer. The effectiveness of the proposed method is demonstrated on a real-life example: the implementation of a FIR filter able to equalize the room transfer function of a recording studio used also as a concert hall. The optimized synthesis procedure yielded a FIR equalizer whose number of taps was reduced by an order of magnitude compared to the results reported by the reference method. The circuit implementation of the equalizer required only a small fraction of the resources provided by a mid-range FPGA development board. Objective and subjective measurements were carried out with and without the equalizer being inserted in the acoustic chain; they showed clearly that the proposed equalizer improved significantly the audition.

Keywords: *multiple position equalization, room transfer function, genetic algorithm, recording studio, concert hall, FPGA.*

I. INTRODUCTION

In general, an acoustic chain consists of a signal source (transmitter), a signal processing/mixing console, an audio amplifier, loudspeakers (source), the enclosed space (room) and a receiver (listener or test microphone). The same receiver will perceive sound differently in different location within the room, due to the multiple sound reflections from the walls, ceiling and the floor of the room. Each receiving point can be described by a Room Transfer Function (RTF), which depends on the room acoustical characteristics and the position on the loudspeakers relative to that point. In principle, the audition can be improved by inserting an equalizer into the acoustic chain – usually within the mixing console – able to compensate for the disturbing effects of multiple sound reflections.

The first studies towards equalization in an audition area were performed on single-point equalization [1]. However, the equalization of the acoustic path only for a particular receiving point may have adverse effects for other receiving points. Therefore, one has to design the equalizer considering an overall RTF, representative for the entire audition area. Such an RTF can be obtained by combining the responses from multiple receiving points.

In most of the cases the RTF is a mixed phase function and the inversion of the non-minimum phase part is impossible. The solution obtained with the method from [1] was improved by minimizing the least-square error or by homomorphic methods [2]. The multiple-input/multiple-output inverse theorem was used in [3] to obtain exact equalization for all the receiving points; note that this required that the number of loudspeakers exceeds the number of receiving points.

Other solutions proposed in the literature for achieving multiple-point equalization employ adaptive algorithms [4], common acoustical poles [5] and digital all-pole filters [6]. Most of these approaches are difficult to deploy in practice, as they require substantial computational effort, and thus important hardware resources, for actual implementation.

A notable exception is the multiple position equalization method presented by Carini: its starting point is the equalization technique based on fuzzy c-means clustering in the time domain introduced in [7] but achieves a significant reduction of the computational complexity by operating in the frequency domain [8]. Even so, the length of the resulting equalizer remains rather large: 512 taps [7], [8].

This paper presents a design method for multiple position equalizers focused on reducing the hardware resources required by its implementation. It is based on Carini's method described in [8] but with significant improvements. The proposed method is described in Section II, while Section III presents a real-life example: the implementation of a FIR filter able to equalize the RTF of a recording studio also used as a concert hall. Conclusions are drawn in the last Section.

Manuscript received February 14, 2015; revised February 24, 2015

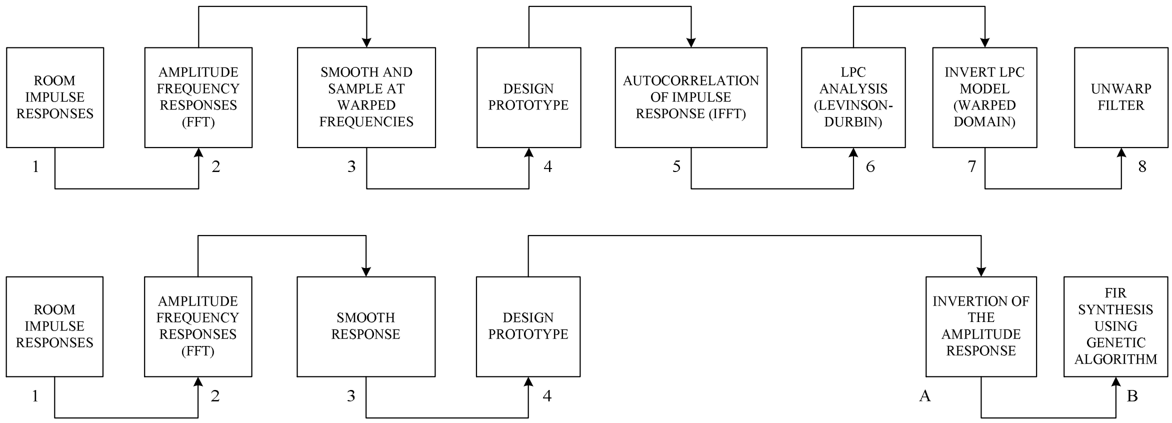

Figure 1. Multiple position room response equalization block diagram using the method from [8] (top) and the proposed method (bottom)

II. THE PROPOSED METHOD FOR DESIGNING MULTIPLE-POSITION EQUALIZERS

Figure 1 top shows the block diagram of the method for designing a digital acoustic equalizer introduced in[8].

For convenience, the block diagram of the method proposed here is placed immediately underneath it, at the bottom of Figure 1.

One notices that the first four steps of these methods are similar: first, the RTFs associated with all the receiving points considered are obtained through measurements; second, the amplitude frequency responses are derived, which are smoothed in the third step; the filter prototype is designed in the fourth step. However, there are several important differences:
- the proposed method employs a Maximum Length Sequence (MLS)-type measurement signal, whereas [8] uses a Logarithmic Sine Sweep (LSS)-type signal [9];
- in our approach the smoothing of the amplitude frequency response is achieved simply by root mean square (RMS) averaging, without using window functions;
- we do not perform the warp/unwarp operations.

Warping is used to increase the accuracy of the amplitude response at low frequencies (<100 Hz), which is often impaired by nonlinear distortions caused by the audio amplifiers and processing equipment. But most people cannot perceive sounds clearly (if at all) at frequencies below 100 Hz. Moreover, the accuracy of measurements performed in the low frequency bands is inherently poor. Considering also the computational effort and the hardware resources required to implement the warping/unwarping filters we decided to not use them.

The fourth step of both the method proposed here and the one presented in [8] consists in deriving the magnitude response of the overall RTF for the entire audition area, called the design prototype. It was proven in [8] that the design prototypes obtained by using the mean, RMS, median or min-max approaches are as good as the solutions provided by the fuzzy c-means, but the later requires far more computational effort.

The remaining steps of the proposed method are different from those used in [8]. There, the Levinson–Durbin algorithm is employed to obtain an all-pole filter model of the design prototype, which can then be inverted to obtain the equalizer. Note that the Levinson–Durbin algorithm needs the autocorrelation of the impulse response of the design prototype. These operations are shown in Figure 1 up as steps 5-7.

The proposed method takes a different approach, that implied only two steps, denoted A and B in Figure 1 bottom. First, one derives the required magnitude response of the equalizer by processing the design prototype as follows: the mean value of the RFT is computed, then it is subtracted from each of the RTF values measured at the N test frequency points, effectively shifting the RTF along the vertical axis until it gets centered on the 0dB axis; the wanted values of the equalizer magnitude response at the test frequency points, $A^{dB}_{want}(f_k)$ result by simply changing the signs of the corresponding points of the centered-on-0dB version of the RTF. Thus, the magnitude response of the equalizer is effectively symmetrical with respect to the 0dB axis with the centered-on-0dB RTF:

$$A^{dB}_{want}(f_k) = -(RTF^{dB}(f_k) - mean(RTF^{dB})); \; k = \overline{1,N} \quad (1)$$

Second, a FIR filter that implements the wanted magnitude response is synthesized by using a design tool developed by the authors for this purpose. This tool approaches the synthesis of the equalizer as a multicriteria optimization problem, targeting the minimization of both the deviation of the resulting frequency response from the wanted one and the filter length [9]. The optimization loop is driven by a genetic algorithm (GA). The filter synthesis is iterative: for each design iteration the GA generates a number of possible solutions; each of them is then evaluated, that is the distance between the wanted, A_{want}, and the synthesized, A_{synth}, amplitude (in dBs) responses is computed at each of the M frequency points set by the user for monitoring the magnitude response, yielding the *peak magnitude error* (PME) [10]:

$$PME(f_k) = max\left\{\left|A^{dB}_{want}(f_k) - A^{dB}_{synth}(f_k)\right|\right\}; \; k = \overline{1,M} \quad (2)$$

Usually there are substantially more monitoring points than frequency test points for measuring the RTF, that is in

general M >> N.
The filters for which the PME value is lower than the maximum acceptable value, set by the user, are saved in the solution library and the synthesis process starts afresh, targeting a filter with a smaller number of taps. The process ends when suitable filters can no longer be found, despite running the algorithm for a number of iterations, determined by the maximum number of generations set by the user.

The method described here is conceptually simpler than the method reported in [8]; also, the GA-based optimized synthesis yields several FIR equalizers that meet the application requirements. The user can choose from the set of available solutions the one best suited for implementation, for example the filter with the smallest length.

III. FIR EQUALIZER SYNTHESIS USING GENETIC ALGORITHM

The block marked B in Figure 1 (bottom) represents the synthesis of the FIR equalizing filter based on GA. This Section presents in some detail the filter synthesis tool developed by the authors for this application.

The block diagram of the proposed synthesis method is presented in Figure 2. Its synthesis engine – the block called Frequency Sampling FIR Design in Fig. 2 – is based on the Frequency Sampling method (FS method). The FS method provides a straightforward way of deriving the coefficients of a FIR filter whose magnitude response is determined by a set of frequency-magnitude points, called here "the reference points".

The standard form of the FS method has a major shortcoming: the resulting filters exhibit large approximation errors at intermediary points (between the reference points), particularly when the distances between successive reference points vary considerably. In order to avoid this drawback the proposed method employs a dedicated GA for optimizing the set of samples used by the synthesis engine.

For the GA optimization each filter is represented by a chromosome, with the structure shown in Figure 3: it comprises only the position in frequency of the samples used by the FS method to synthesize the filter. Note that the GA does not change the position of the first and last samples – they remain always f_{min} respectively f_{max}, so that all resulting filters cover the entire band of interest, (f_{min}, f_{max}). The magnitude values corresponding to these frequency positions are derived through interpolation, targeting the wanted magnitude response of the filter, itself defined by interpolation of the reference points. The resulting frequency-magnitude points are called here the "design set of samples".

The proposed synthesis method is organized as an iterative optimization loop driven by a GA developed specifically for this task: for each iteration the GA provides a new design set of samples to the Frequency Sampling FIR Design block; the PME values for the resulting filters are calculated in the Eval Filter block; if none of these values are smaller than the maximum allowable PME, *PMEmax*, set by the user, a new iteration starts. If at least one filter if found which has the PME value smaller than *PMEmax*, the length of the synthesized filters is decreased and the synthesis procedure starts again. The initial set of design samples for the shorter-length optimization cycle is derived

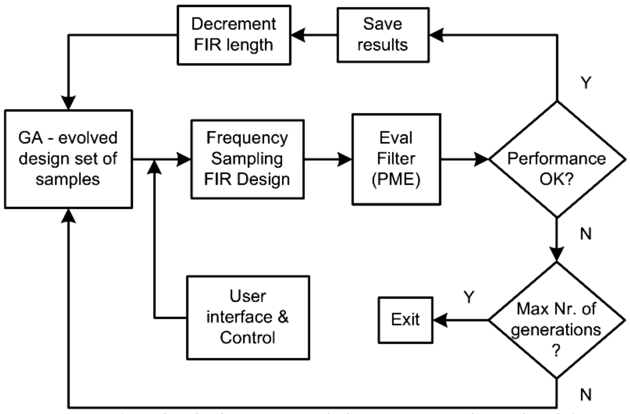

Figure 2. Block diagram of the proposed method for designing FIR filters with the magnitude responses that match the frequency characteristic defined by the user

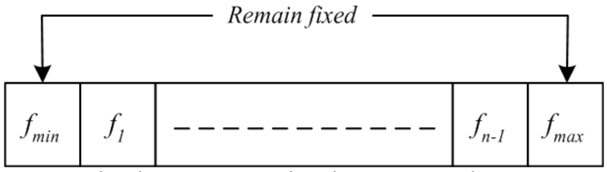

Figure 3. The structure of a chromosome that represents a filter: it comprises only the position in frequency of the samples used to synthesize the corresponding FIR filter

from the one used in the previous run, by simply eliminating one of the samples from each set. The procedure stops after a set number of iterations, set by the user, are run without finding a valid solution. All synthesized filters which meet the main design requirement, PME < *PMEmax*, are saved and delivered to the user; if no valid solution is found the user receives the best two filters, i.e. those with the lowest PME values.

Figure 4 presents the LabWIEW diagram of the design tool presented here. The LabVIEW environment was preferred to the more popular Matlab due to its higher computational speed and better support for on-the-fly adjustment of parameters.

At the beginning of the synthesis procedure the user inputs the reference points that define the desired magnitude characteristic, the design requirements (*PMEmax* and the maximum filter length) and the parameters of the GA algorithm which will be detailed in the followings. This step is enabled by the User Interface & Control block in Fig. 4.

The Initial Population block in Fig.4 generates PS random chromosomes, where PS is the population size set by the user. The GA core placed immediately on its right in Fig. 4 implements the filter synthesis engine and the evaluation blocks from Fig.2: it yields PS FIR filters, each being characterized by its "fitness score", equal here with the PME metric defined by eq. (2). If none of the filters in the current population meets the design requirement, that is PME < *PMEmax*, the next iteration starts. First, "offspring" of the current generation is derived by using crossover and mutation operators that mimic the natural reproduction. Standard roulette-wheel selection of parents was implemented, along with standard single-point crossover and single-bit mutation operators [11]. The user sets the mutation probability considering that too small a value makes mutation irrelevant while a high mutation rate could turn the generation-by-generation evolution driven by the GA into a random search.

The Clone Elimination" block in Fig. 4 searches for clones (identical individuals) within both the initial population and its offspring; the clones are replaced with randomly generated individuals. The process is repeated until PS new, distinct chromosomes are obtained, forming a complete new generation. The FIR filters corresponding to these new, clone-free chromosomes are synthesized and evaluated within the GA core, as for the previous generation. Then, an operation that mimics the natural selection is performed on the entire "genetic pool" available: the Sort-after-score block in Fig. 4 selects the best PS filters within the previous population and its clone-free offspring, considering their fitness scores. Thus, a new generation is obtained. If none of its individuals meets the design requirements a new iteration starts.

The GA described here is somewhat similar to the one reported in [9] but the key difference is the fact that the optimization objectives are different: the fitness function in [9] is determined by the total squared error normalized to the wanted magnitude values. Also, the GA in [9] aims at reducing the spread of the resulting filter coefficients. The parent selection algorithms are different, as well as the implementation of the crossover and mutation operators.

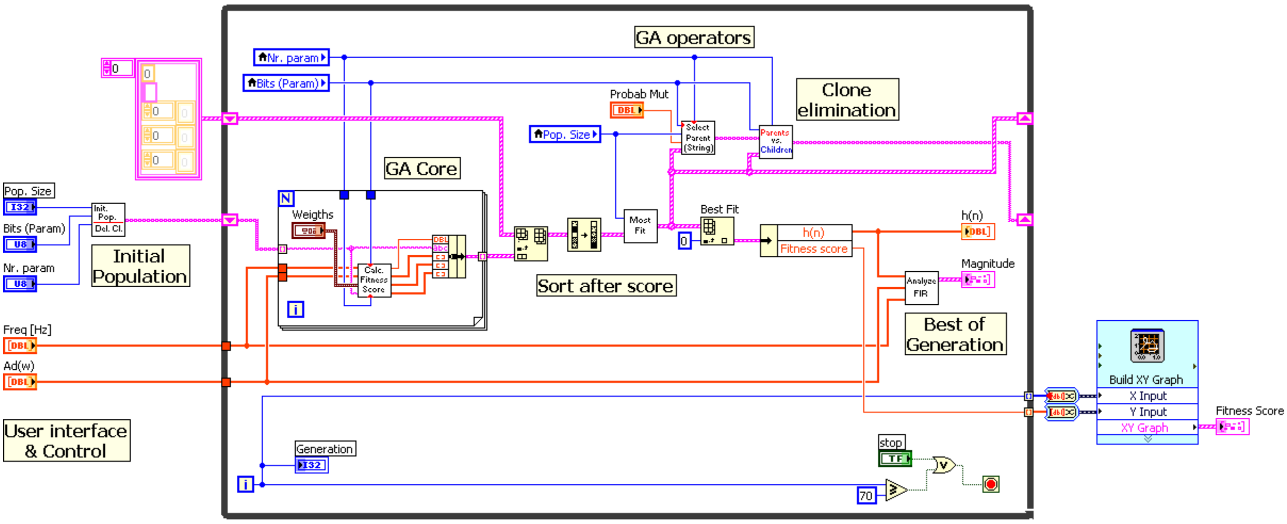

Figure 4. The LabVIEW diagram for implementing the proposed method for FIR filter synthesis

IV. DESIGN EXAMPLE: OPTIMIZED FIR EQUALIZER FOR EFFECTIVE HARDWARE IMPLEMENTATION

A real-life application is presented in this Section in order to demonstrate the effectiveness of the proposed method.

A. Brief description of the application

We implemented our multiple-point sound equalization approach in a recording studio with a volume of 3.400 m3 and a reverberation time of 0.7 s at 1 kHz. Figure 5 presents the floor plan of the studio, which is also used occasionally as a concert hall. The locations of the two loud-speakers (S1, S2) and of the receiving points (P1 to P20) are highlighted. The placement of points P1-P20 was chosen so that they meet the requirements of the ISO 3382-1 standard [12]. The goal of the work presented here is to equalize the RTF of this enclosure.

First, we measured the RTF of the studio. An MLS type signal at 48 kHz sampling frequency was chosen for excitation, with a level of 75 dB above the background noise [13], in accordance with ISO 3382-1 [13], [14], [15].

In our experiments we used an external sound card MOTU 896HD (8 channels), PCB130 omnidirectional microphones operating within the frequency range of 20 Hz - 20 kHz and two loudspeakers (JBL-VRX932LAP, 75 Hz - 20 kHz, 136 dB SPL at 1 m) with incorporated audio amplifier. The Adobe Audition software was used for processing the signals measured during tests.

The room responses to the MLS test signal corresponding to the 20 receiving points, P1-P20, were measured at 25 points placed at frequencies between 100 Hz and 16 KHz with a third of an octave between them. Then the corresponding impulse responses were determined through deconvolution.

Next, the resulting amplitude responses for the receiving points have been smoothed, yielding the plots shown in Figure 8.a. Finally, the overall RTF for the entire audition area was derived, as shown in Fig. 8.b.

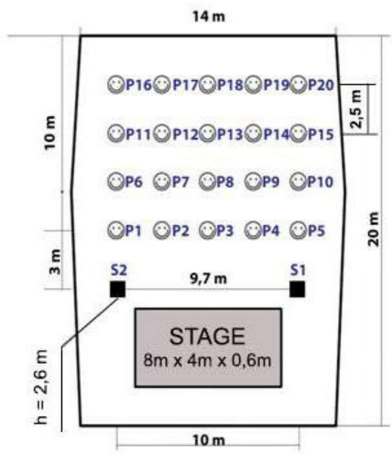

Figure 5. The floor plan of the recording studio

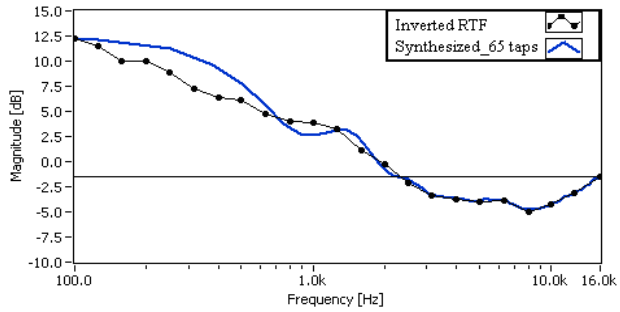

Figure 6. The desired magnitude response of the equalizer (dotted line) and the response of the synthesized FIR filter (continuous line)

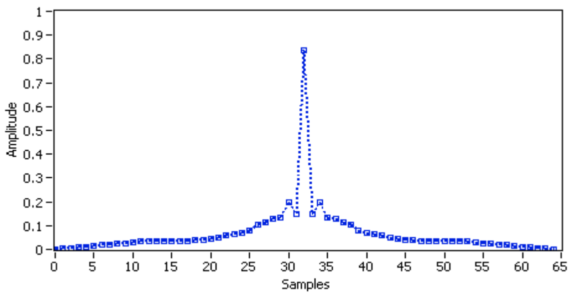

Figure 7. The impulse response of the synthesized 65-taps equalizer

These operations correspond to steps 1-4 in Figure 1 (bottom); they were described in detail in [13].

It should be noted that the application envisaged here is more demanding than the ones considered in [8]:

- three rooms were considered there, all far smaller than the one considered here: the largest size was 7.5m x 5m. The distances between the receiving points were correspondingly shorter in [8] than in our case.
- all measurements performed in [8] employed only one loud-speaker; also, only five receiving points were used in two of the three cases considered there.

B. Synthesis of the FIR Equalizer

The dotted line plot shown in Figure 6 presents the wanted magnitude response of the equalizer; the black squares there represents the wanted magnitude values at the 25 frequency test points, obtained from the overall RTF by applying eq. (1).

The synthesis tool described in Section II was used to find type-I FIR filters with magnitude responses that approximate the wanted characteristic. The maximum acceptable PME value was set to 3 dB and M=128 monitoring points were chosen, equally spaced within the 100 Hz-16 KHz range. The shortest filter found that met the requirements had only 65 taps – remarkably less that the 512 taps of the equalizer found in [8].

The continuous-line plots shown in Figure 6 presents the magnitude response of our optimized filter; it is reasonably close to the wanted characteristic, with the distance between characteristics measured at the 128 observation points smaller than PMEmax=3dB. The impulse response of the 65-taps FIR equalizer is presented in Figure 7; it has the symmetry typical of type I FIR filters, with perfectly linear phase response, which does not introduce phase distortions. There are no oscillations in this impulse response, so the filter is stable.

C. FPGA Implementation of the Proposed Equalizer

The synthesized equalizer was implemented on the cRIO-9104 FPGA board from National Instruments. This chassis contains the Virtex-II FPGA chip with 3M gates and 8 slots for external hardware modules. Sound acquisition was performed by using the external analog input and output modules NI 9215 and NI 9263 connected directly to the FPGA module. To ensure sufficient accuracy for the signal processing the acquired signal was converted into digital with 20 bits resolution. All the signal processing was performed in LabVIEW.

The report generated by the Xilinx FPGA compiler provides the following information: Total Slices: 21.7% (3105 out of 14336), Slice Registers: 12.1% (3458 out of 28672), Slice LUTs: 15.5% (4435 out of 28672), Block RAMs: 5.2% (5 out of 96). Obviously, the resources provided by the cRIO-9104 board are under-used, so a smaller, lower-cost FPGA board can be used instead. This proves the effectivenes of the proposed approach, from the simpler equalization procedure (which does not require warping/unwarping filters) to the optimized synthesis of the FIR equalizer (which includes reducing the filter length).

It should be noted that [8] reported software-only implementations of the equalizers described there; by comparison, the complete circuit-level implementation was performed.

D. Measurement Results

The implemented equalizer was inserted in the acoustic chain used to measure the RTFs shown in Figure 8 and a new series of measurements were carried out, for exactly the same conditions. Figure 9.a shows the resulting smoothed magnitude responses for the P1-P20 receiving points, while Figure 9.b shows the overall RTF of the audition area after equalization.

By comparing the later with the initial RTF shown in Figure 8.b the effect of the equalizer becomes apparent:

- the distance between two succesive plots is significantly smaller in Fig. 9.a. than in Fig. 8.a. Even the shapes of the equalized RTFs shown in Figure 9.a are fairly similar to each other, far more so than their non-equalized counterparts shown in Figure 8.a. This indicates that the sound will have similar quality albeit different intensity at the P1-P20 receiving points.
- the RTF values vary over a narrower range when equalization is performed: -69dB to -53dB against -66dB to -48dB without equalization.

Subjective assessments made by human listeners confirmed the positive impact made by the equalizer. An important feature noticed by all listeners was the metallic sensation of the sound before equalization (due to the large value of high frequency components of the initial RTF, shown in Figure 8.b) which was much attenuated after equalization (in line with the shape of the equalized RTF shown in Figure 9.b).

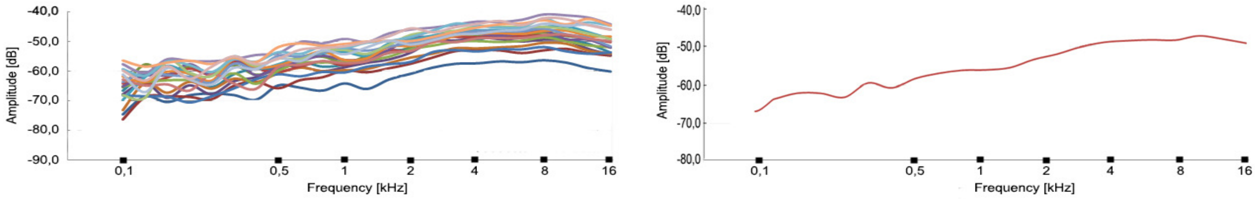

Figure 8. a) The smoothed magnitude responses for the P1-P20 points; b)The RTF for the entire audition area (the design prototype)

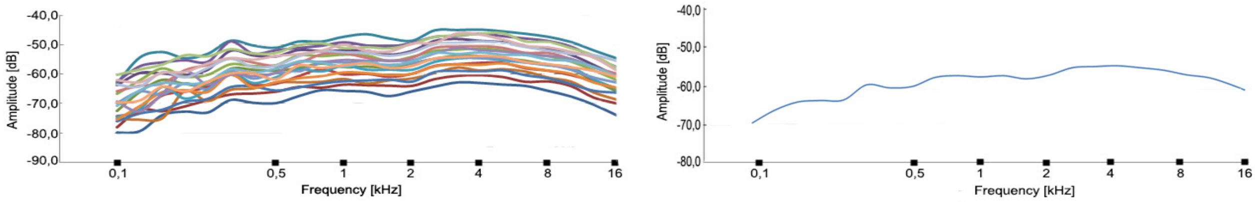

Figure 9. Measured magnitude responses with the equalizer inserted in the signal chain. a) the smoothed RTFs corresponding to the P1-P20 receiving points; b) the overall RTF for the audition area

V. CONCLUSIONS

A method for multiple-position equalization of a room transfer function was presented. It is based on the design method reported in [8] but significant improvements were made in order to reduce the computational effort and hardware resources required by its implementation. First, we proposed a way for deriving the desired frequency response of the equalizer directly for the measured RTF, much easier to apply than the Levinson-Durbin algorithm employed in [8]: the wanted values of the equalizer magnitude response at the frequency test points are the opposite (equal module but different sign) of the measured RTF values at those points, from which the RTF mean value was subtracted. The equalizer response is symmetrical with the centered-on-0dB RTF. Second, a FIR filter with the wanted magnitude response was obtained by using a novel synthesis tool based on a genetic algorithm, able to curtail the differences between the wanted and synthesized frequency responses while minimizing the filter length. The proposed method was used to solve a real-life design problem: the implementation of a digital equalizer for a recording studio/concert hall, by using an off-the-shelf FPGA board. The synthesized FIR filter was able to approximate the wanted magnitude response with a maximum error of 3dB, although it had only 65 taps - whereas the equalizer proposed in [8] had 512 taps. It should be noted that in [8] multiple-point equalization was performed for less demanding cases: smaller-size rooms (maximum 7.5m x 5m compared to 20m x 14m in our case), only one signal source and a smaller number of receiving points. Objective and subjective measurements were carried out with and without the equalizer being inserted in the acoustic chain; they showed clearly that the proposed equalizer improved significantly the audition.

ACKNOWLEDGMENT

This paper was supported by the Post-Doctoral Programme POSDRU/159/1.5/S/137516, co-funded by the European Social Fund through the Human Resources Sectorial Operational Program 2007-2013. The authors also thank National Instruments Romania and the Cluj Territorial Studio of Radio Romania for their support.

REFERENCES

[1] S. T. Neely and J.B. Allen, "Invertibility of a Room Impulse Response", *J. Acoust. Soc. Am.*, vol. 66, no. 1, pp. 165-169, 1979.
[2] J. Mourjopoulos, P. M. Clarkson and J.K. Hammond, "A Comparative Study of Least-Squares and Homomorphic Techniques for the Inversion of Mixed Phase Signals", *IEEE Int. Conf. ASSP*, pp. 1858-1861, 1992.
[3] M. Miyoshi and Y. Kaneda, "Inverse Filtering of Room Acoustics", *IEEE Trans. on Audio, Speech and Lang.*, vol.36, no. 2, pp. 145-152, 1988.
[4] S. J. Elliott and P. A. Nelson, "Multiple-Point Equalization in a Room Using Adaptive Digital Filters", *J. Audio Eng. Soc.*, vol.37, no. 11, pp. 899-907, 1989.
[5] Y. Haneda, S. Makino and Y. Kaneda, "Multiple-Point Equalization of Room Transfer Functions by Using Common Acoustical Poles", *IEEE Trans. on Speech and Audio Proc.*, vol. 5, no. 4, pp. 325-333, 1997.
[6] J.N. Mourjopoulos, "Digital Equalization of Room Acoustics", *J. Audio Eng. Soc.*, vol. 42, no. 11, pp. 183-186, 1994.
[7] S. Bharitkar and C. Kyriakakis, "A cluster centroid method for room response equalization at multiple locations", *Workshop on Applic. of Signal Proc. to Audio and Acoustics*, pp. 55-58, 2001.
[8] A. Carini, S. Cecchi, F. Piazza, "Multiple position room response equalization in frequency domain ", *IEEE Trans. on Audio, Speech & Lang Proc*, vol.20, no.1, pp. 122-135, 2012.
[9] E.Szopos, C.Fărcaş, M.Topa, M.Neag and I.Sărăcuţ, "LabView Tool for Designing FIR Filters for Acoustic Equalization", *Acta Technica Napocensis*, vol.55, no.1, pp.26-32, 2014.
[10] E.Szopos, M.Neag, I.Sărăcuţ, H.Hedeşiu and L.Feştilă, "A method for designing FIR filters with arbitrary amplitude characteristic used for modeling human audiogram", *Advances in Electrical and Comp. Eng.*, vol. 12, no. 2, pp. 51-56, 2012.
[11] Haupt, R.L., Haupt, S.E., *Practical Genetic Algorithms*, John Wiley & Sons Inc., 2004.
[12] International Standard ISO 3382–1, Acoustics — Measurement of room acoustic parameters, Part 1: Performance spaces, ISO Geneva, 2009.
[13] A. C. Fărcaş and M. Ţopa, "The adaptation of a recording studio for a performance room, using multiple position room response equalization", *Int. Conf. on Acoustics AIA-DAGA*, Merano, Italy, pp. 2094-2097, 2013.
[14] N. G. Kanev, A. Ya. Livshits, and H. Möller, "Acoustics of the Great Hall of the Moscow State Conservatory after Reconstruction in 2010-2011", *Acoustical Physics* , vol. 59, no. 3, pp. 361-368, 2013.
[15] N. G. Kanev and A. Ya. Livshits, "Changes in the Acoustic of the Moscow Conservatory Great Hall Two Years after Reconstruction", *Acoustical Phys.*, vol.60, no.2, pp.197-199, 2014.