The Impact of the Acoustic Environment on Recovering Speech Signals Drowned in Loud Music

Robert Alexandru Dobre, Radu-Mihnea Udrea, Cristian Negrescu, Dumitru Stanomir
Telecommunications Department
Politehnica University of Bucharest
Bucharest, Romania
email: rdobre@elcom.pub.ro, mihnea@comm.pub.ro, negrescu@elcom.pub.ro, dumitru.stanomir@elcom.pub.ro

***Abstract*—In many trials, multimedia materials, video or audio, brought as evidence could make the difference between "guilty" or "not guilty" verdicts. Most multimedia content is stored in a digital form nowadays, therefore, with so many free editing software at anyone's disposal, it is very easy to be forged. In other situations the critical evidence, even if recorded, it can be heavily masked by other signals and declared inappropriate. This paper is a contribution to the multimedia forensic domain presenting the impact of the acoustic environment on a computer software based on adaptive filtering, which can be used to recover a speech signal drowned in loud music. The results help to decide if placing a microphone in a certain room could be useful or not given the proposed solution for recovering the speech is to be used afterwards.**

Keywords-multimedia forensic; noise reduction; adaptive filtering.

I. INTRODUCTION

There are many ways in which criminals could act in order to turn the tables on their side in a trial. For example, they could do basic editing of audio recordings in order to change the meaning of the message and present the forged material as evidence. This audio signal must be authenticated before taken into account. The direction of multimedia forensics that study these problems is called multimedia authentication. In other situations, the suspects can be tapped. If some people would like to discuss something of great importance and they are afraid that a microphone can be placed in the room where the dialogue is about to take place, the simplest solution that would come into mind to make the conversation private is to turn very loud any nearby audio system. This way the speech signal would be drowned in the loud music and the recording, at a first glance, could be considered useless. There are very high chances that the musical material would be represented by a radio station program or the studio versions of some songs recorded on a CD or any other storage form. With all the advances in musical material identification, the melody can be precisely determined and a studio quality version of it can be acquired. The problem in this stage is as follows: given the speech and loud music mixture recorded using the microphone placed in the tapped room and the studio quality of the song that masks the dialogue in the recording, can these signals be processed in such way that the speech signal can be recovered? This is a typical adaptive noise reduction problem and its configuration is depicted in Figure 1.

In Figure 1 $s_{speech}(t)$ represents the ideal speech signal, or the speech that would be recorded in open space conditions and $n_{music}(t)$ is the masking melody in studio quality. $h(t)$ is a finite impulse response (FIR) filter that models the acoustic environment in which the recording took place and $r(t)$ is the actually recorded signal, the sum of the aforementioned signals affected by the room's acoustics. In the recorded mixture, given the intention of the speakers to hide their conversation, the musical signal dominates. The recorded signal is fed into a music identification software like Shazam or SoundHound and the masking song is identified. Furthermore, the louder the music is turned in the room (with the purpose to achieve better masking), the easier is the job of the identification software, so the speakers may even help the forensic engineer without knowing it. After the successful identification, the studio quality version of the song is acquired. In order to be able to remove the musical signal from the recorded mixture, an estimate for the impulse response of the room [$h_{est}(t)$] is needed. In the mentioned conditions this can be found using an adaptive algorithm [recursive least squares (RLS)[1][2] and variable forgetting factor RLS (VFF-RLS) are used]. In the end, the error signal of the adaptive algorithm denoted $e(t)$ will be a good estimate for $s_{speech}(t)$. More precisely, it will represent the speech signal affected by the room's acoustics, but that is what everyone hears when talking to other persons in a closed acoustic environment every day, so it is clearly intelligible. The problem that arises is how large can be the room [which translates into how long the $h(t)$ impulse response can be] for the proposed solutions

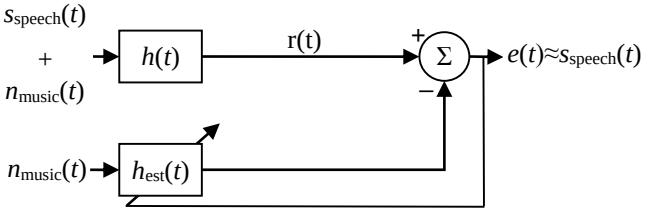

Figure 1. The adaptive noise reduction configuration.

 ISBN: 978-1-61208-546-3

to still work with good performances (the recovered signal is clearly intelligible).

Besides this brief introduction, the paper consists of another four sections: Section II presents the algorithms that were used in the speech recovering software, Section III details the actually speech recovery solution along with some experimental results, in Section IV the impact of the room size on the proposed solution is described and the results of the conducted experiments are presented and Section V concludes the paper.

II. RLS AND VFF-RLS ALGORITHMS

The speech recovery software is mainly built around a system identification problem. The chosen algorithms were RLS and VFF-RLS because of their very fast convergence speed, property which is important in the presented application. The recovered signal would not be intelligible when the adaptive algorithm is not in steady state (the room's impulse response is not accurately determined) so, if the convergence speed would be low, the unusable part could be too large and corrupt the meaning of the message. In the following equations the largely accepted adaptive filtering notations will be used: x – the input signal, d – the desired signal, $\mathbf{w}$ – the adaptive filter's coefficients vector, e – the error signal.

A. The RLS algorithm

The RLS algorithm uses a totally different approach compared to the classical least mean squares (LMS) or the normalized LMS (NLMS) detailed in [1] and [2]. It uses more than just one sample of the error signal to update the adaptive filter's coefficients. Its cost function is described by (1):

$$c_N(\mathbf{w}_N) = \sum_{k=0}^{n} \lambda^{n-k} \left| e_N(k,n) \right|^2 \quad (1)$$

where λ is a constant called forgetting factor, N is the length of the adaptive filter and

$$0 < \lambda \le 1, \quad (2)$$

$$e_N(k,n) = d(k) - \mathbf{w}_N^{\mathrm{T}}(n)\mathbf{x}_N(k), \quad (3)$$

$$\mathbf{x}_N(k) = \left[x(k), x(k-1), ..., x(k-N+1) \right]^{\mathrm{T}} \quad (4)$$

where $(\cdot)^{\mathrm{T}}$ is the transposition operator (only real signals are taken into consideration). The solution to the minimization of the cost function with respect to $\mathbf{w}$ is:

$$\mathbf{R}_N(n)\mathbf{w}_N(n) = \mathbf{D}_N(n), \quad (5)$$

where $\mathbf{R}_N$ is a correlation matrix computed using:

$$\mathbf{R}_N(n) = \sum_{k=0}^{n} \lambda^{n-k} \mathbf{x}(k)\mathbf{x}^{\mathrm{T}}(k) \quad (6)$$

and the cross-correlation vector $\mathbf{D}_N$ can be computed using:

$$\mathbf{D}_N(n) = \sum_{k=0}^{n} \lambda^{n-k} \mathbf{x}(k) d(k) \quad (7)$$

The name of the RLS algorithm comes from its property that the vector $\mathbf{w}$ can be determined recursively. It is clearly more computationally complex that the aforementioned classical adaptive algorithms, but its convergence speed is much greater.

B. The VFF-RLS algorithm

The performance of an adaptive algorithm in estimating an unknown filter is given mainly by two indicators: the misalignment and the convergence speed. The misalignment is defined as the norm of the difference vector between the vector containing the coefficients of the filter to be estimated and the vector containing the estimated coefficients. Using the notations introduced in Figure 1 this translates as:

$$m(n) = \left| h(n) - h_{\mathrm{est}}(n) \right|^2 \quad (8)$$

The convergence speed gives an information about the amount of time the adaptive algorithm needs to reach its minimum misalignment.

The choice of the forgetting factor parameter in RLS algorithm is done by making a compromise: a small λ will give very fast convergence speed, but the convergence will not be very strong (the misalignment will have large values) while a larger λ will give better misalignment performances, but will decrease the convergence speed [3]. An algorithm with variable forgetting factor is desirable, which could detect large misalignment and decrease the λ parameter in order to speed up the convergence and progressively increase it as the misalignment decreases.

In [4] and [5] new ways in which λ can be computed are shown:

$$\lambda(n) = \begin{cases} \min\left(\dfrac{\sigma_q(n)\sigma_v(n)}{\xi + \left| \sigma_e(n) - \sigma_v(n) \right|}, \lambda_{\max} \right), & \sigma_e(n) \le \gamma\sigma_v(n) \\ \lambda_{\max}, & \sigma_e(n) > \gamma\sigma_v(n) \end{cases}, \quad (9)$$

$$0 < \gamma \le 1, \quad (10)$$

where ξ is a small positive constant to avoid division by zero, $\lambda_{\max}$ is a preset maximum value for the forgetting factor and

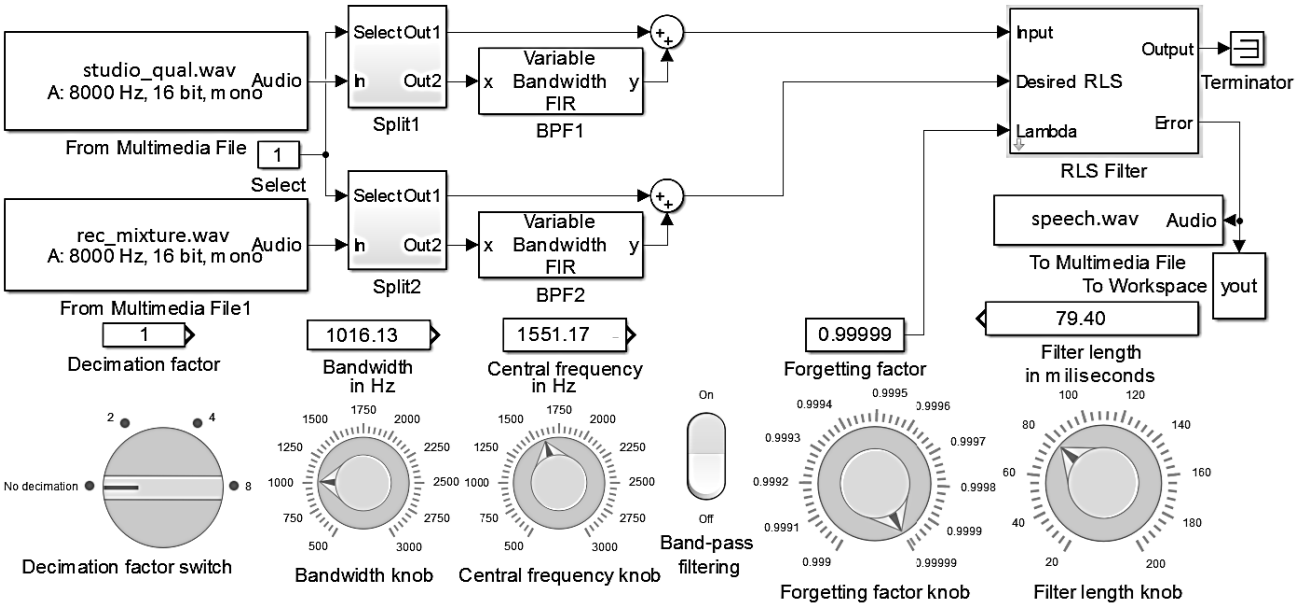

Figure 2. The forensic software for speech recovering based on the RLS algorithm.

$$\sigma_e^2(n) = \left(1 - \frac{1}{KN}\right)\sigma_e^2(n-1) + \left(\frac{1}{KN}\right)e^2(n), \quad (11)$$

$$\sigma_q^2(n) = \left(1 - \frac{1}{KN}\right)\sigma_q^2(n-1) + \left(\frac{1}{KN}\right)q^2(n), \quad (12)$$

$$\sigma_v^2(n) = \left(1 - \frac{1}{K_\beta N}\right)\sigma_v^2(n-1) + \left(\frac{1}{K_\beta N}\right)v^2(n), \quad (13)$$

$$K_\beta > K \geq 1, \quad (14)$$

$$q(n) = \mathbf{x}^T(n)\mathbf{R}_N^{-1}(n)\mathbf{x}(n), \quad (15)$$

where K=2 and K_β=5·K for noise input or K=6 and K_β=3·K for speech input.

III. DESCRIPTION OF THE FORENSIC SOFTWARE

The forensic software for recovering speech signals drowned in loud music was briefly described in the introduction. The details are presented onwards.

A sample rate equal to 8 kHz is considered sufficient for the acquisition of a speech signal while musical signals are sampled using much higher rates ranging from 44.1 kHz to 192 kHz in very rare cases. Because the speech signal is the sought one, there is no need to assume that the recording sample rate would be larger than 8 kHz. The first step is to equalize the sample rates of the two available signals: the recorded mixture and the identified studio quality musical signal. This is achieved by decimating the latter.

Since there is a high chance that the source of the musical signal is a radio station, it is known that the songs are usually crossfaded, so parts (especially the beginning and the end) of the melody will miss from the recording. Since the available studio quality melody contains also these parts, some preprocessing must be done to the signals before applying the adaptive filtering. Particularly, the musical signals must be aligned. In theory, the adaptive filter can handle this aspect by itself, but it will greatly increase the computational effort and since both RLS and VFF-RLS are not very computationally light, avoiding this additional task from the main processing becomes a necessity. A method for aligning the signals based on the cross-correlation function can be imagined and it was detailed in [6]. Since the adaptive filter can intervene in this aspect, the alignment does not need to be perfect.

The RLS based forensic software that was developed using Simulink is presented in Figure 2. The signals to be processed are loaded in two multimedia files readers named "From Multimedia File" and "From Multimedia File1". For uncommon situations or for speeding the processing especially useful in initial testing a decimation control was provided. The user can select if the signals are decimated before processing or not and the value of the decimation factor.

Two tunable band-pass filters can be switched on or off as needed. The role of these filters is to preselect the spectral band occupied by the speech signals having the effect of reducing the work of the adaptive filter. Their parameters, i.e., the central frequency and the bandwidth, can be set using the corresponding knobs named suggestively "Central frequency

 ISBN: 978-1-61208-546-3

knob" and "Bandwidth knob". The on and off switching of the filters is done using the rocker switch named "Band-pass filtering". Finally, the parameters of the adaptive filter (the forgetting factor and the filter length) are set using the "Forgetting factor knob" and the "Filter length knob". The recovered speech signal is saved using a dedicated block named "To Multimedia File".

In order to test the implemented forensic software, it was proceeded as follows: a speech signal was mixed with a musical signal (in the role of the masking noise) in a very harsh signal to noise ratio, –40 dB. Then the mixture was processed using an impulse response that models an acoustic environment illustrated in Figure 3. The variation of the misalignment for the RLS algorithm can be observed in Figure 4, confirming its very fast convergence. The forgetting factor was set at 0.999999 and the length of the adaptive filter matched the length of the impulse response t models the acoustic environment. The recovered signal is a very good estimate for the initial speech signal as it can be seen in Figure 5.

The RLS algorithm gives very good results in this situation as shows the absolute recovery error (the absolute value of the difference between the normalized initial speech signal and the normalized recovered speech signal), which is negligible. However, this particular solution can be applied only if the acoustic properties do not change over time, which is not very often in real situations.

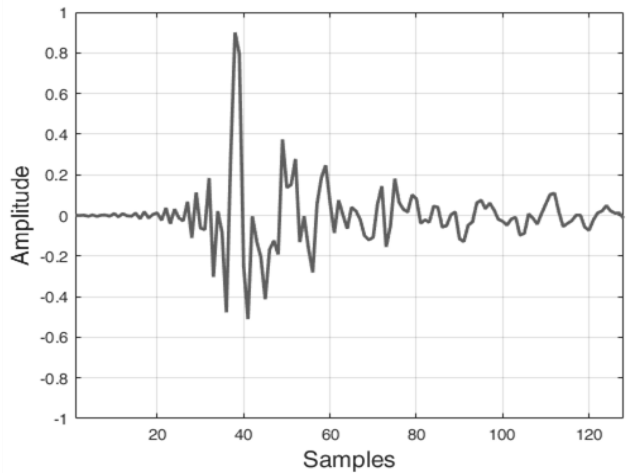

Figure 3. The impulse response used to model the acoustic environment.

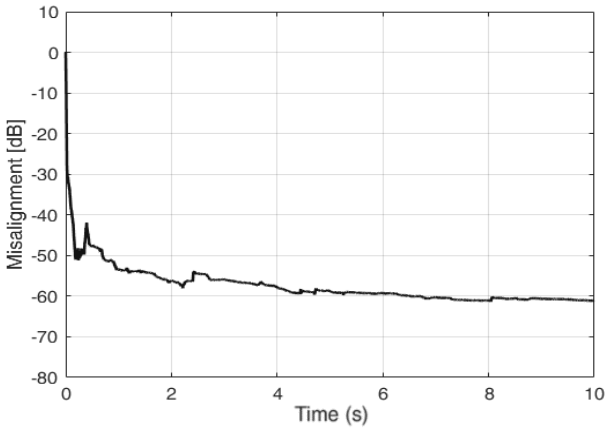

Figure 4. The variation of the misalignment for the RLS algorithm.

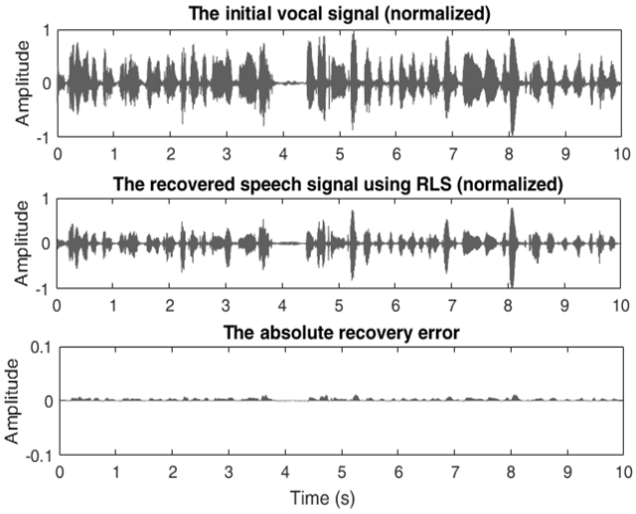

Figure 5. The performances of the RLS algorithm in the given situation.

The situation in which the acoustic properties change over time was studied. After 5 seconds the impulse response of the room was modified (shifted with 8 samples). The results presented in Figu re 6 show that the RLS algorithm, because of its large and constant forgetting factor, cannot recover from this sudden change (the absolute recovery error is much greater than the reference signal after the moment of the change), while the VFF-RLS algorithm recovers very quickly (in about 10 ms). The reason why the VFF-RLS behaves this way is its ability to modify the forgetting factor. The variation of the VFF-RLS parameters can be observed in Figure 7, and,

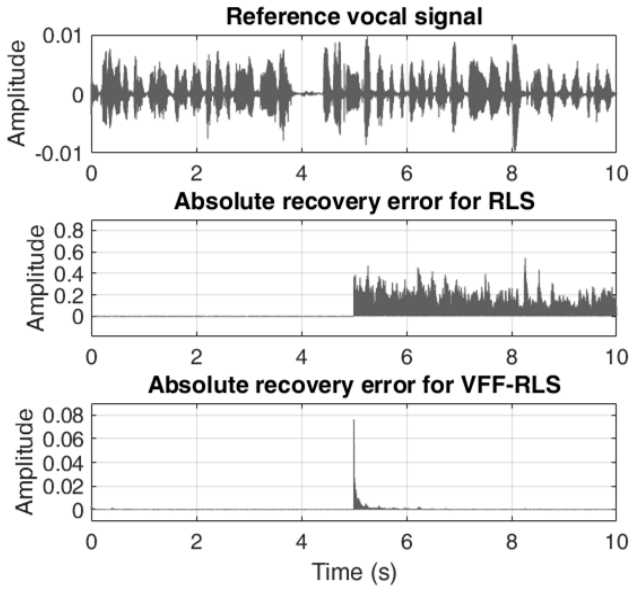

Figure 6. The performances of the RLS and VFF-RLS algorithms in the case in which a change in the acoustic parameters occurs.

in Figure 8, the evolution of the forgetting factor is depicted. It can be observed that λ is equal to the maximum value (0.999999) for most of the time and a very sudden change occurs at the moment when the impulse response is modified. The forgetting factor λ drops to very small values in order to grant the algorithm a fast convergence speed, then its value grows to assure the sought low misalignment. Figure 9 shows the variation of the misalignment for the two algorithms and confirms the conclusions drawn above. Other parameters are K=6 and K_β=3·K.

IV. ACOUSTIC ENVIRONMENT IMPACT ON THE PROPOSED SOLUTION

From the previous experiments it resulted that the preferred algorithm to be used for recovering a speech signal drowned in loud music is VFF-RLS. Its disadvantage of being more computationally complex than the RLS (which is already demanding from this point of view compared to the classical algorithms) is compensated by its ability to follow changes in the acoustic parameters. Changes in the acoustic parameters could mean the moving of the speakers through the room, opening doors, people entering or leaving etc.

The length of the chosen impulse response for the tests conducted above was 128 samples, which at a sample rate of 8 kHz would mean 16 ms. A room that can be characterized by such a small impulse response is either very small or it is acoustically treated (typically only studios and professional audio spaces are treated, so it is not a very common situation) or both.

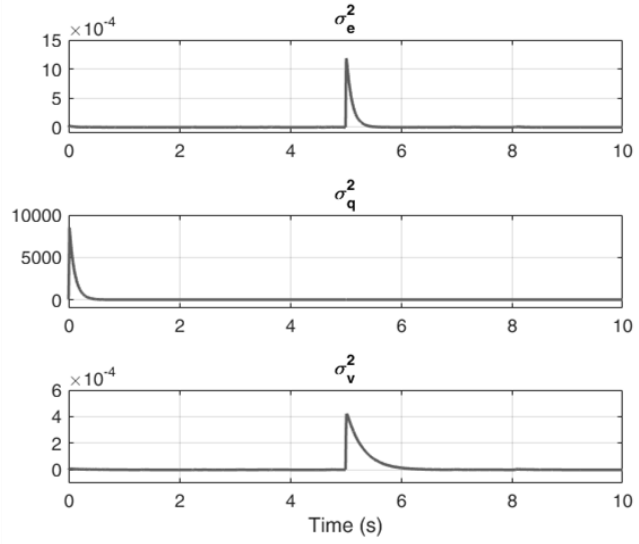

Figure 7. The variation of the VFF-RLS parameters.

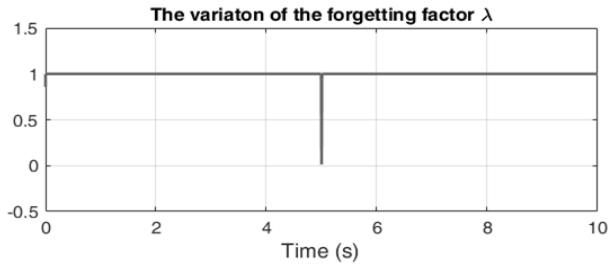

Figure 8. The variation of the forgetting factor.

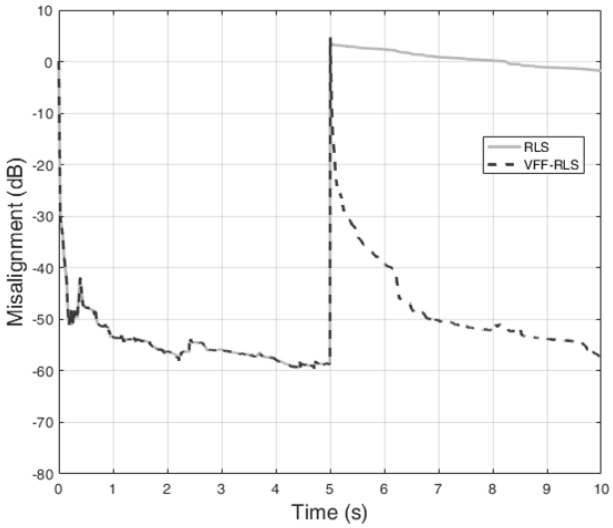

Figure 9. The variation of the misalignment for the two adaptive algorithms.

The purpose of this paper is to investigate the maximum length of the impulse response that characterizes a room for which the presented software gives good results. It is clear that the computational complexity will increase with the length of the impulse response.

For this, a longer (i.e., 512 samples) impulse response was considered, depicted in Figure 10. The length of the impulse response used in the experiment was progressively increased starting from 128 samples in order to determine the length at which the performances of the adaptive algorithm are not

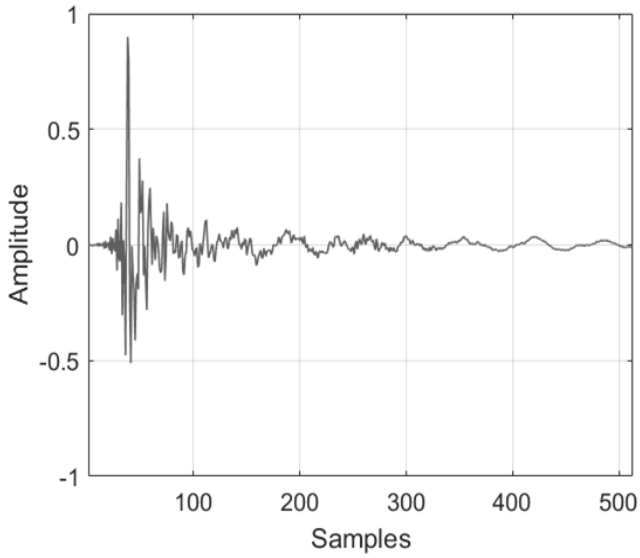

Figure 10. The impulse response used for studying the impact of acoustic parameters on the proposed solution.

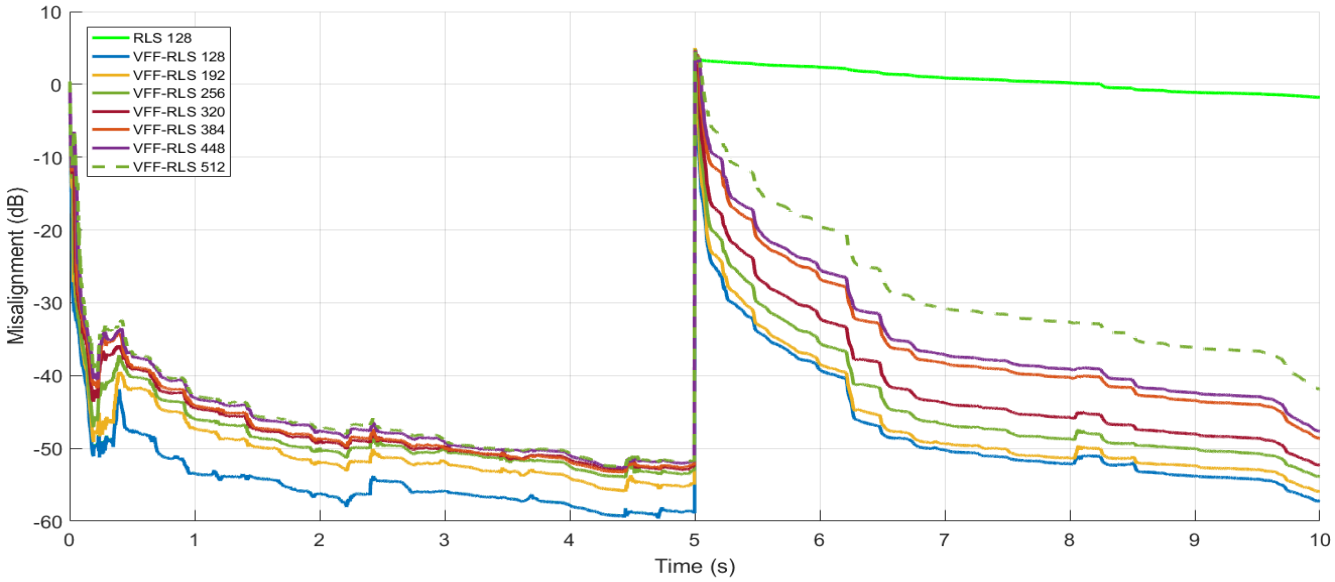

Figure 11. The variation of the misalignment for the two adaptive algorithms with respect to the length of the impulse response.

satisfactory anymore. The impulse response from Figure 10 was truncated from the start to various lengths and the previous experiments were rerun.

In Figure 11, the misalignment of the two adaptive algorithms with respect to the length of the impulse response can be observed. The RLS algorithm works very well until the acoustic parameters are changed, then it loses convergence and, because the forgetting factor is large and fixed, its convergence speed is slow. The VFF-RLS shows promising results for following the impulse response change for low filter lengths, then, for longer filters, its performance slightly degrades, but it is still usable for the whole impulse response of 512 samples.

V. Conclusion and Future Work

In this paper, the problem of recovering a speech signal drowned in loud music was described and its importance in the field of multimedia forensic was highlighted.

It was shown how adaptive filters can be used in order to solve the stated problem. A software developed using Simulink, which uses adaptive algorithms for recovering speech drowned in loud music, was presented and characterized from the performance point of view.

The RLS algorithm gives very good results if the acoustic properties of the room in which the recording takes place does not vary over time. Since this is a rare situation, a change in these parameters was simulated and the performances fell drastically. Another algorithm, VFF-RLS, was tested in these more realistic conditions and its results are promising.

The length of the impulse response is in close relation with the size of the room or the quality of the acoustic treatment that could exist in it, but very few rooms are treated. Tests were conducted in order to determine the longest impulse response for which the VFF-RLS algorithm is still usable.

The maximum length of the impulse response for which the VFF-RLS gives usable results was concluded to be 512 samples. This information along with basic knowledge of acoustics (Sabine's reverberation time formula) can help to decide if placing a microphone in a certain room it's worth it or not.

Future work will include testing the algorithms using other types of impulse response changes than time shifting.

Acknowledgment

This work was supported by UEFISCDI Romania under Grant PN-II-RU-TE-2014-4-1880, and under Grant PN-III-P2-2.1-PED-2016-1465/No. 32PED/2017.

References

[1] S. Haykin, Adaptive Filter Theory. Fourth Edition, Upper Saddle River, NJ:Prentice-Hall, 2002.

[2] A. H. Sayed, Adaptive Filters. New York, NY: Wiley, 2008.

[3] S. Ciochina, C. Paleologu, J. Benesty, and A. A. Enescu, "On the influence of the forgetting factor of the RLS adaptive filter in system identification," in Proc. IEEE ISSCS, 2009, pp. 205–208.

[4] C. Paleologu, J. Benesty, and S. Ciochină, "A robust variable forgetting factor recursive least-squares algorithm for system identification," IEEE Signal Processing Letters, vol. 15, pp. 597–600, 2008.

[5] C. Paleologu, J. Benesty, and S. Ciochină, "A practical variable forgetting factor recursive least-squares algorithm," in Proc. ISETC, 2014, pp. 1–4.

[6] R. A. Dobre, C. Negrescu, and D. Stanomir, "Development and testing of an audio forensic software for enhancing speech signals masked by loud music," Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies 2016, pp. 100103A-100103A-7, 2016.

 ISBN: 978-1-61208-546-3