# A Practical Forensic Method for Enhancing Speech Signals Drowned in Loud Music

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

***Abstract*—Recording audio or video is nowadays easier than ever. Almost every phone can do this task with high quality. This has some serious implications in forensic: almost every dialogue or event can be recorded and used as evidence in trials. The problem is that editing multimedia content has also become a very accessible operation. The advances of editing software make it possible with very convincing results for the untrained audience. Forged recordings could be used in trials. The need for multimedia forensic is imminent. There are two main directions of this field: probe authentication and noise reduction. This paper presents the research activities conducted to extract speech signal masked by loud music. The developed system is based on an adaptive system identification configuration. Various scenarios are studied showing the advantages and disadvantages of the adaptive algorithms that were tested. The influence of the acoustic environment over the performances of the proposed system is also studied and the results can help to determine if placing a microphone in a specific room could be used to intercept a speech.**

***Keywords-adaptive algorithms; system identification; noise reduction; multimedia forensic.***

## I. INTRODUCTION

The technological advances made the recording of high quality multimedia content available to almost everyone. Phones have rapidly turned into small pocket computers and they are more affordable as time passes. Since phones are mainly used for speech communication, it is self-understood why audio recording is an easy task, but most of them are also fitted with at least one video camera allowing the user to capture full HD video for a decent period, like tens of minutes. On high end terminals, even state of the art 4K video can be recorded.

From the security point of view, there are two sides of this situation, explained onwards. The first implication is: if anyone can store a clear multimedia recording of an event, it means that many trials should end very quickly. With clear evidence of the events, very little is, apparently, remaining to be evaluated. It is necessary to mention that along the evolution of the recording devices, the industry of multimedia editing software also grew, allowing one can edit the recordings before presenting them as evidence. This brings to light the second implication: the multimedia content can be edited and the verdict may not reflect the consequence of the real events. Special training to use these editing software is not needed, and some of them are available for free, so the malicious editing can be considered as easy as the recording. To the untrained audience, the forgeries could be very convincing. These two implications show the necessity of some authorities and technologies to counteract these illegal actions. This paper concentrates on the latter part.

Before allowing some multimedia content as evidence into a trial, it must be determined if it is the original, unaltered version. This process is called content authentication and it represents one large field of multimedia forensics. There are other situations in which the material is not forged, but greatly affected by noise in such way that the key element (some specific spoken phrase or a zone of an image) is heavily masked. This is another research direction called noise reduction. The work presented in this paper is part of this topic and it extends the results shown in [1].

In [2], power spectral subtraction based methods for speech enhancement are presented. These methods could give very good results if the noise is slowly varying in time and the speech signal is not drowned into it. Other methods based on Wiener filtering or which use singular values decomposition (SVD) are presented in [3][4]. The method presented in this paper has the advantages of simplicity and good performances in harsh signal-to-noise ratio conditions, but, unlike the other methods, it is specifically designed for one particular situation.

Besides this introduction, the rest of this paper is organized as follows. Section II describes a speech recovery method, Section III thoroughly describes the suitable adaptive algorithms [5][6], Section IV presents and discusses the results, and Section V concludes the paper.

## II. THE DESCRIPTION OF THE SPEECH RECOVERY APPROACH

The studied situation is the following: if a group of people would like to speak about something confidential, it is obvious that they will take some measures to avoid being intercepted. If they suspect that there is a high chance for a microphone to be placed in the room, the easiest way to avoid being recorded when talking is to turn very loud any nearby music player. This will make the speech signal (i.e., the secret discussion) to be heavily masked (or "drowned") by the loud music. The captured audio signal will be dominated by the music and could be considered useless. It is a very high chance that the source for the musical signal is a radio station or a labeled CD and so the melody has some

notoriety. Music identification software (like Shazam or SoundHound) very rarely fail to recognize even the most exotic tunes nowadays and they could be used to determine the masking melody. The original, studio quality, full length melody can be bought (or simply downloaded in many cases since an important part of artists give their music for free) and made available to the forensic engineer. The problem restates as: if the recorded signal is a mixture of the sought speech signal and a masking melody and if the melody is identified and available in studio quality, can the latter be processed in a way that could make it match the recorded melody so by subtracting these two signals, the speech would be recovered? This is a typical adaptive system identification situation.

The real situation has some specific elements like the acoustic properties of the room that were not discussed in the short description above. All the audio signals that propagate in a room will be affected by the acoustic impulse response of the room. The microphone will record the direct wave, but also all the waves that are reflected by the various objects present in the room. Since the recorded signal will be composed of multiple delayed replicas of the direct wave, the propagation of the sound waves between two points in a room can be modelled using a finite impulse response (FIR) filter. Taking this added element into consideration, a more accurate situation is illustrated in Figure 1. The properties of the impulse response (length, sparsity, etc.) have great impact on determining the solution that could be used to extract the speech from the masking music, as it is shown in the following sections.

Let us denote with $s_{speech}(n)$ the speech signal unaffected by the acoustic environment (i.e., that speech signal that would be recorded if the microphone and the speaker are in open space conditions) and with $n_{music}(n)$ the studio quality masking musical signal. The signal recorded using the microphone that is placed in the room [$r(n)$] is modelled as a mixture of the two aforementioned signals filtered with the acoustic impulse response, denoted with $h(n)$. Keeping in mind the speakers' intention to conceal their dialogue, the musical signal dominates the mixture. The recorded signal is analyzed using a music identification software, and the masking song is found and acquired. Furthermore, the louder the masking music is turned, the easier becomes the job of the music identification software. This means that in their try to conceal their secret, the speakers could unintentionally help the extraction of the masked dialogue. There are high chances that the music being played in the room is in the same format as the music that is acquired in studio quality, since radio stations use the commercially available version also. If the music is played from a CD, a CD can also be made available. In the event that the music is transcoded, the problem gets tougher because the CD version must be encoded using various codecs, various encoding settings, then decoded and processed by the forensic software. This scenario involving the estimation of the encoder is not considered in this paper. The final element that is required to recover the speech is a good estimate for the room's acoustic impulse response denoted with $h_{est}(n)$, which could be determined using an adaptive filter connected in the system

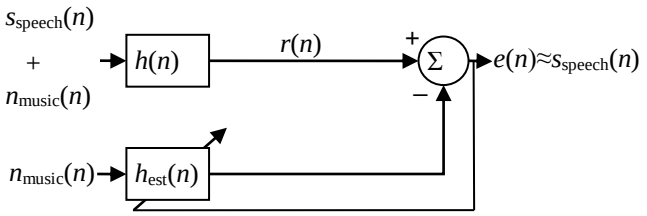


Figure 1. The adaptive noise reduction configuration in the proposed approach.

identification configuration. The result of filtering the acquired studio quality melody with $h_{est}(n)$ and then subtracting it from the recorded mixture will be called the error signal [$e(n)$] and it will represent a good estimate for the secret speech signal. In fact, in the ideal situation of perfect extraction (no trace of music can be identified in the extracted signal), the recovered speech will be the ideal speech (the direct sound wave) filtered with the room's acoustic impulse response. This is not a problem since this kind of signals are heard every day when speaking with somebody in a room. The presented method is practical in the considered scenarios.

## III. ADAPTIVE ALGORITHMS

The operation that is at the foundation of eliminating the masking music is the identification of the system that models the acoustic properties of the room. An adaptive filter will evolve in such way to match the filter that models the sound waves' propagation in the room. Generally, an adaptive algorithm's task is to minimize a cost function. Updating the impulse response of the adaptive filter can be done in multiple ways, using various adaptive algorithms.

Typically, an adaptive algorithm has two input signals denoted with $x(n)$ and $d(n)$. Usually $x(n)$ is called the input signal and $d(n)$ is known as the desired signal. In the described system identification problem, the signal $d(n)$ is the output of the unknown filter (i.e., the acoustic impulse response of the room). The vector containing the coefficients of the unknown filter is denoted onward with $\mathbf{w}_o$ and the one containing the coefficients of the adaptive filter is denoted with $\mathbf{w}$ because these are the common notations used in literature. The quantity that gives and characterizes the quality of the estimation is known as the misalignment and is evaluated as:

$$m(n) = \left\| \mathbf{w}(n) - \mathbf{w}_o(n) \right\|. \quad (1)$$

where $\|\cdot\|$ is the $l_2$ norm of a vector.

Another variant to evaluate the performance of the algorithm is the normalized misalignment, computed as:

$$m_{normalized}(n)=\frac{\left\|\mathbf{w}(n)-\mathbf{w}_o(n)\right\|}{\left\|\mathbf{w}_o(n)\right\|}. \quad (2)$$

A cost function based on the error signal [the difference between $d(n)$ and the output of the adaptive filter, denoted with $y(n)$] is considered and its minimization represents an optimization problem. Various approaches are used by different algorithms to give the solution. Only real signals are considered in this paper (the signal samples and filter coefficients are real numbers).

### A. The least-mean-squares and the normalized least-mean-squares algorithms

The cost function used in the case of the least-mean-squares (LMS) algorithm is the square error, hence the name of the algorithm. It is defined as:

$$C(n)=e^2(n) \quad (3)$$

where $e(n)$ denotes the aforementioned error signal.

The minimization of the cost function is done with respect to the **w** vector. The solution gives the impulse response of the adaptive filter at the $n$ sample time:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mu\mathbf{x}(n)\left[d(n)-\mathbf{w}^{\mathrm{T}}(n-1)\mathbf{x}(n)\right], \quad (4)$$

where $\mu$ is a parameter known as step-size and $\{\cdot\}^{\mathrm{T}}$ is the transposition operator. If the length of the adaptive filter is considered equal to $L$, the structure of the input data vector involved in (4) is:

$$\mathbf{x}(n)=\left[x(n),x(n-1),\ldots,x(n-L+1)\right]^{\mathrm{T}}, \quad (5)$$

The step-size will be chosen by making a compromise between a better estimation quality (given by a smaller step-size) and a faster, but coarser estimation. The $\mu$ parameter cannot take any value. For assuring the convergence of the algorithm, $\mu$ must respect the following relation:

$$0<\mu<\frac{2}{\mathrm{tr}\{\mathbf{R}\}}, \quad (6)$$

Where tr{·} is the trace of a matrix and **R** is the autocorrelation matrix of the input signal computed as:

$$\mathbf{R}=\mathrm{E}\left\{\mathbf{x}(n)\mathbf{x}^{\mathrm{T}}(n)\right\}, \quad (7)$$

where E{·} is the statistical expectation.

A great disadvantage of the LMS algorithm arises from equations (6) and (7): in practice, choosing a step-size that will guarantee convergence is a difficult task since the LMS depends on the scaling of the input signal. This important problem is solved in the normalized LMS (NLMS) algorithm by scaling the step-size with the power of the input signal. Equation (4) becomes:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\frac{\mu\mathbf{x}(n)\left[d(n)-\mathbf{w}^{\mathrm{T}}(n-1)\mathbf{x}(n)\right]}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n)}, \quad (8)$$

where $\mu$ must now respect only $0<\mu<2$. The greatest convergence speed is obtained when $\mu=1$. Since the behavior of the algorithm on the $0<\mu\le 1$ interval is similar with the behavior on the $1\le\mu<2$ interval, the first one is preferred in practice because it greatly reduces the risk of the algorithm going out of convergence.

Since in (8) a division to the power of the input signal is computed, this could generate problems if $\mathbf{x}(n)$ is almost zero. To avoid the situation, a small positive number named the regularization parameter (usually denoted with $\delta$) is introduced, and the final update equation for the NLMS algorithm becomes:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\frac{\mu\mathbf{x}(n)\left[d(n)-\mathbf{w}^{\mathrm{T}}(n-1)\mathbf{x}(n)\right]}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n)+\delta}. \quad (9)$$

The main advantages of the NLMS algorithm are its simplicity and reduced computational cost. One disadvantage could be considered the limited performance tweaking parameters (in this form, only the step-size can be adjusted by the user).

### B. The affine projection algorithm

One cause of the performance limitation in the case of the NLMS algorithm is the fact that it uses only one input signal vector [$\mathbf{x}(n)$]. The performance worsens for correlated input data. The affine projection algorithm (APA) increases the performance in the mentioned situation by using more than one input signal vector. The number of the input signal vectors used by the algorithm is controlled by a specific parameter named "projection order", denoted with $M$. The existence of this new tweakable parameter increases the flexibility of the algorithm in terms of the convergence speed/misalignment compromise. The obvious consequence of this operation is an increase in the computational complexity. The $M\times L$ matrix containing the $M$ input signal vectors, denoted with **A**, is constructed as:

$$\mathbf{A}^{\mathrm{T}}(n)=\left[\mathbf{x}(n),\mathbf{x}(n-1),\ldots,\mathbf{x}(n-M+1)\right]. \quad (10)$$

Using this new matrix approach, it can be shown that equation (9) becomes:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mu\mathbf{A}^{\mathrm{T}}(n)\left[\mathbf{A}(n)\mathbf{A}^{\mathrm{T}}(n)+\delta\mathbf{I}_M\right]^{-1}\mathbf{e}(n), \quad (11)$$

where $\mathbf{I}_M$ is the $M$ order identity matrix and now

$$\mathbf{e}(n)=\mathbf{d}(n)-\mathbf{y}(n), \quad (12)$$

$$\mathbf{d}(n)=\left[d(n),d(n-1),...,d(n-M+1)\right]^{\mathrm{T}}, \quad (13)$$

$$\mathbf{y}(n)=\mathbf{A}(n)\mathbf{w}(n-1). \quad (14)$$

A major computational load represented by the inverse of a matrix can be observed in (11). Larger projection orders lead to an increase in the convergence speed, but also to a worse system identification. Another observation is that the NLMS algorithm is a particular case of APA, obtained when $M=1$. An actual topic of interest is the convergence of the APA. If the evolution of the misalignment can be computed in some sufficiently general situations, $M$ and $\mu$ can be chosen to obtain the desired performances. The quality of the estimation can be evaluated by:

$$\mathrm{E}\left\{\left\|\mathbf{c}(n)\right\|^2\right\}=\mathrm{E}\left\{\left\|\mathbf{w}_{\mathrm{o}}(n)-\mathbf{w}(n)\right\|^2\right\}, \quad (15)$$

In a more realistic situation, a zero mean white noise (named system noise) denoted with $v(n)$, having a variance equal to where $\sigma_v^2$ is intervening at the output of the unknown system, transforming the desired signal in:

$$\mathbf{d}(n)=\mathbf{A}(n)\mathbf{w}_{\mathrm{o}}+\mathbf{v}(n), \quad (16)$$

with $\mathbf{v}(n)$ respecting the structure in (13). In these conditions, by denoting:

$$\mathbf{C}(n)\triangleq\mathbf{A}^{\mathrm{T}}(n)\left[\mathbf{A}(n)\mathbf{A}^{\mathrm{T}}(n)+\delta\mathbf{I}_M\right]^{-1}, \quad (17)$$

equation (15) becomes:

$$\begin{aligned}&\mathrm{E}\left\{\left\|\mathbf{c}(n)\right\|^2\right\}=\\&\mathrm{tr}\left\{\mathrm{E}\left\{\mathbf{c}(n-1)\mathbf{c}^{\mathrm{T}}(n-1)\left[\mathbf{I}_L-\mu\mathbf{C}(n)\mathbf{A}(n)\right]^2\right\}\right\}\\&+\mu^2\mathrm{tr}\left\{\mathrm{E}\left\{\mathbf{v}(n)\mathbf{v}^{\mathrm{T}}(n)\mathrm{E}\left\{\mathbf{C}^{\mathrm{T}}(n)\mathbf{C}(n)\right\}\right\}\right\}+T_M,\end{aligned} \quad (18)$$

where

$$\begin{aligned}&T_M=\\&-2\mu\mathrm{tr}\left\{\mathrm{E}\left\{\mathbf{v}(n)\mathbf{c}^{\mathrm{T}}(n-1)\left[\mathbf{I}_L-\mu\mathbf{C}(n)\mathbf{A}(n)\right]\mathbf{C}(n)\right\}\right\}.\end{aligned} \quad (19)$$

The general solution in the case of a first level of approximation [7] shows that:

$$\begin{aligned}\mathrm{E}\left\{\left\|\mathbf{c}(n)\right\|^2\right\}&=\mathrm{E}\left\{\left\|\mathbf{c}(0)\right\|^2\right\}a^n\left(\beta,\sigma_x^2,M,L\right)\\&+\frac{b\left(\beta,M,L,\sigma_x^2,\sigma_v^2\right)\left(1\text{-}a^n\left(\beta,\sigma_x^2,M,L\right)\right)}{1\text{-}a\left(\beta,\sigma_x^2,M,L\right)},\end{aligned} \quad (20)$$

where $\sigma_x^2$ is the variance of the input signal and

$$a\left(\beta,\sigma_x^2,M,L\right)=1-2\beta M\sigma_x^2+\beta^2LM\sigma_x^4, \quad (21)$$

$$b\left(\beta,M,L,\sigma_x^2,\sigma_v^2\right)=\beta^2\sigma_x^2\sigma_v^2LM+T_M, \quad (22)$$

$$\beta\triangleq\frac{\mu}{L\sigma_x^2+\delta}, \quad (23)$$

Equation (21) gives the convergence speed, while the residual misalignment can be computed using (22). Under the convergence condition, in this first level of approximation the residual misalignment is found as:

$$\lim_{n\to\infty}\mathrm{E}\left\{\left\|\mathbf{c}(n)^2\right\|\right\}=\frac{\beta L\sigma_v^2}{\left(2-\beta L\sigma_x^2\right)}+\frac{T_M}{1-a\left(\beta,\sigma_x^2,M,L\right)}, \quad (24)$$

with $T_M=0$, which would mean that the residual misalignment is independent of $M$. Experimental results contradict this statement.

The analysis done using a second order approximation shows that:

$$T_M\cong 2\beta^2\left(1-\beta L\sigma_x^2\right)L\sigma_x^2\sigma_v^2\sum_{m=1}^{M-1}(M-m)\left(1-\beta L\sigma_x^2\right)^{m-1}. \quad (25)$$

This analysis can be used to decide on choosing the APA working parameters to satisfy the necessities of a specific situation. Further details based on less restrictive conditions are given in [8].

### C. Proportionate variants of APA

The aforementioned adaptive algorithms do not make use of any information about the filter to be estimated. In particular situations, some properties of the unknown filter can be known *a priori*. In the context of the application presented in this paper, the unknown system is represented by acoustic impulse responses, which are usually sparse (a small part of coefficients is significant and the rest are almost equal to zero). The residual misalignment in a situation when only the significant coefficients are estimated (and the others are considered equal to zero) will be almost equal with the residual misalignment when the filter evolves to estimate the

whole impulse response. It would clearly be an advantage to prioritize the estimation of the significant coefficients because it would lead to increase the convergence speed. The proportionate variants of adaptive algorithms exploit these properties. The update equation of APA in this case becomes:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mu\mathbf{G}(n-1)\mathbf{A}^{\mathrm{T}}(n)\mathbf{Z}^{-1}\mathbf{e}(n), \quad (26)$$

$$\mathbf{Z}=\mathbf{A}(n)\mathbf{G}(n-1)\mathbf{A}^{\mathrm{T}}(n)+\delta\mathbf{I}_M, \quad (27)$$

where $\mathbf{G}(n-1)$ is a $L\times L$ diagonal matrix of gains representing the way of exploiting the properties of sparse impulse responses. Each element of the $\mathbf{G}(n-1)$ matrix is computed using [9]:

$$g_l(n-1)=\frac{1-\alpha}{2L}+(1+\alpha)\frac{|w_l(n-1)|}{2\sum_{k=0}^{L-1}|w_k(n-1)|+\varepsilon}, \quad (28)$$

with $l=\overline{0,L-1}$, $-1\le\alpha<1$ and $w_l$ representing the elements of the **w** vector. Typically, the **w** vector is initially filled with zeros, which would lead to similar problems as the ones discussed about equation (8). The problem is solved in a similar manner, by introducing a small positive constant denoted with $\varepsilon$. This version of APA is named improved proportionate APA (IPAPA). In the particular case of $M$=1, a proportionate NLMS algorithm is obtained.

The operations presented in (26) can be simplified by exploiting the structure of the **A** matrix. A more efficient version of IPAPA was proposed in [10]. The update equation for this algorithm is:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mu\mathbf{P}(n)\left[\mathbf{A}(n)\mathbf{P}(n)+\delta\mathbf{I}\right]^{-1}\mathbf{e}(n), \quad (29)$$

where the structure of **P** matrix is

$$\mathbf{P}(n)=\left[\mathbf{p}_1(n),\mathbf{p}_2(n),...,\mathbf{p}_M(n)\right]. \quad (30)$$

The elements of **P** can be computed recursively as:

$$\mathbf{P}(n)=\left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\ \ \mathbf{P}_{2\cdots M}(n-1)\right], \quad (31)$$

where $\mathbf{P}_{2\cdots M}(n-1)$ is a $L\times M-1$ matrix containing the last $M-1$ columns of $\mathbf{P}(n-1)$:

$$\mathbf{P}_{2\cdots M}(n-1)=\left[\mathbf{p}_2(n-1),\mathbf{p}_3(n-1),...,\mathbf{p}_M(n-1)\right], \quad (32)$$

and $\odot$ is the Hadamard product. The structure of $\mathbf{g}(n-1)$ can be found by knowing:

$$\mathbf{g}(n-1)\odot\mathbf{x}(n)=\mathbf{G}(n-1)\mathbf{x}(n). \quad (33)$$

This variant of IPAPA is called memory IPAPA (MIPAPA).

### D. The recursive least-squares algorithm

The algorithms presented above have difficulties in situations in which the input signals are highly correlated. The recursive least-squares (RLS) algorithm offers a higher convergence rate in such situations, but its drawback is its high computational complexity. This algorithm is part of the Kalman filters family. Unlike the LMS and the NLMS algorithms, the RLS uses more than one sample of the error signal in its coefficients update equation. The cost function that is used by the RLS is:

$$C_L(\mathbf{w}(n))=\sum_{l=1}^{n}\lambda^{n-l}|e(l,n)|^2, \quad (34)$$

where λ is the RLS specific parameter called "forgetting factor". For real signals:

$$e(l,n)=d(l)-\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(l), \quad (35)$$

The coefficients of the adaptive filter are found by minimizing the cost function with respect to the **w** vector. The solution is found as:

$$\mathbf{R}_L(n)\mathbf{w}(n)=\mathbf{D}_L(n), \quad (36)$$

where $\mathbf{R}_L$ is the correlation matrix and $\mathbf{D}_L$ is the cross-correlation vector. These two quantities are computed as:

$$\mathbf{R}_L(n)=\sum_{l=1}^{n}\lambda^{n-l}\mathbf{x}(l)\mathbf{x}^{\mathrm{T}}(l), \quad (37)$$

$$\mathbf{D}_L(n)=\sum_{l=1}^{n}\lambda^{n-l}\mathbf{x}(l)d(l), \quad (38)$$

Keeping in mind that **x** is a vector with the length equal to $L$, solving the above equations would require more and more memory as the time index $n$ grows. Fortunately, as the name implies, the **w** vector can be computed recursively.

The relations that define the RLS algorithm are:

$$e(n)=d(n)-\mathbf{w}^{\mathrm{T}}(n-1)\mathbf{x}(n), \quad (39)$$

$$\mathbf{k}(n)=\frac{\mathbf{R}_L^{-1}(n-1)\mathbf{x}(n)}{\lambda+\mathbf{x}^{\mathrm{T}}(n)\mathbf{R}_L^{-1}(n-1)\mathbf{x}(n)}, \quad (40)$$

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mathbf{k}(n)e(n), \quad (41)$$

$$\mathbf{R}_L^{-1}(n)=\frac{1}{\lambda}\left[\mathbf{R}_L^{-1}(n-1)-\mathbf{k}(n)\mathbf{x}^{\mathrm{T}}(n)\mathbf{R}_L^{-1}(n-1)\right], \quad (42)$$

where $\mathbf{k}(n)$ is called the Kalman gain vector.

The forgetting factor, in the classical approach, is a positive constant ($0<\lambda\le 1$) that affects the convergence speed, the residual misalignment, the stability and, very important in the stated problem, the tracking capabilities in the case in which the unknown system changes over time. Unfortunately, a compromise must be made between the previous performance elements [11]. A forgetting factor very close to 1 will make the RLS algorithm to function with good stability and low residual misalignment, but the tracking capabilities are affected.

Typically, in a system identification configuration, the output of the unknown filter is summed with another signal called system noise, as shown in (16). In the context of extracting a speech signal from loud music, the speech signal plays the role of the system noise. The main objective is to make the error signal equal to the speech signal, not to make it equal to zero. It is shown in [12] that a low forgetting factor would determine $y(n)\cong\mathbf{x}^{\mathrm{T}}(n)\mathbf{w}_{\mathrm{o}}(n)+v(n)$ which means $y(n)\cong d(n)$ and $e(n)\cong 0$, while in the case of $\lambda\cong 1$ the output of the adaptive filter would be $y(n)\cong\mathbf{x}^{\mathrm{T}}(n)\mathbf{w}_{\mathrm{o}}(n)$ and consequently $e(n)\cong v(n)$. It can be concluded that in the system identification configuration, the RLS algorithm should work with a forgetting factor very close to 1. While the initial convergence speed would be satisfactory, the algorithm would lack tracking capabilities. A smaller $\lambda$ would improve the tracking, but will determine $e(n)\cong 0$, so a compromise must be made, which led to the development of the variable forgetting factor RLS (VFF-RLS) algorithms.

### *E. The variable forgetting factor recursive least-squares algorithm*

The $e(n)$ signal in (39) uses $\mathbf{w}^{\mathrm{T}}(n-1)$, hence its name could be considered *a priori* error, with its power being $\mathrm{E}\{e^2(n)\}=\sigma_e^2(n)$. Starting from it, an *a posteriori* error can also be defined as:

$$\varepsilon(n)=d(n)-\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)=e(n)\left[1-\mathbf{x}^{\mathrm{T}}(n)\mathbf{k}(n)\right]. \quad (43)$$

In the stated problem, the aim is to recover the speech signal which is, at this stage, modeled by the system noise leading to imposing $\mathrm{E}\{\varepsilon^2(n)\}=\sigma_v^2$. Using this new condition in (43), if the input signal is not correlated with the error signal, the result is:

$$\mathrm{E}\left\{\left[1-\frac{p(n)}{\lambda(n)+p(n)}\right]^2\right\}=\frac{\sigma_v^2(n)}{\sigma_e^2(n)}, \quad (44)$$

where $p(n)=\mathbf{x}^{\mathrm{T}}(n)\mathbf{R}_L^{-1}(n-1)\mathbf{x}(n)$. Another assumption is that the forgetting factor is time dependent and deterministic. The quadratic equation (44) has the following solution:

$$\lambda(n)=\frac{\sigma_p(n)\sigma_v}{\sigma_e(n)-\sigma_v}, \quad (45)$$

where $\mathrm{E}\{p^2(n)\}=\sigma_p^2(n)$. Statistical expectation is avoided in practice, so another method is used to estimate the power of the $e(n)$, $p(n)$ and $v(n)$ signals. By using exponential windows:

$$\hat{\sigma}_e^2(n)=\psi\hat{\sigma}_e^2(n-1)+(1-\psi)e^2(n), \quad (46)$$

$$\hat{\sigma}_p^2(n)=\psi\hat{\sigma}_p^2(n-1)+(1-\psi)p^2(n), \quad (47)$$

where the weighting factor is $\psi=1-1/(K_\psi\cdot L)$, with $K_\psi\ge 2$. The initial values of the two power estimates are $\hat{\sigma}_e^2(0)=\hat{\sigma}_p^2(0)=0$. If a longer exponential window is used, the power of $v(n)$ can be estimated from $e(n)$, from practical reasons, resulting:

$$\hat{\sigma}_v^2(n)=\theta\hat{\sigma}_v^2(n-1)+(1-\theta)e^2(n), \quad (48)$$

with $\theta=1-1/(K_\theta\cdot L)$, $K_\theta>K_\psi$.

Care must be taken in practice when evaluating (45) because it is constructed using power estimates. A solution is to impose $\lambda(n)=\lambda_{\max}$ in the case of:

$$\hat{\sigma}_e(n)\le\varphi\hat{\sigma}_v(n), \text{ with } 1<\varphi\le 2. \quad (49)$$

The forgetting factor can now be evaluated using [13]:

$$\lambda(n)=\begin{cases}\lambda_{computed}(n), & \hat{\sigma}_e(n)\le\varphi\hat{\sigma}_v(n)\\ \lambda_{\max}, & \hat{\sigma}_e(n)>\varphi\hat{\sigma}_v(n)\end{cases}, \quad (50)$$

$$\lambda_{computed}(n) = \min\left(\frac{\hat{\sigma}_p(n)\hat{\sigma}_v(n)}{\xi + \left|\hat{\sigma}_e(n) - \hat{\sigma}_v(n)\right|}, \lambda_{\max}\right), \quad (51)$$

where $\xi$ is a small positive constant to prevent problems that could occur when $\hat{\sigma}_e(n) \cong \hat{\sigma}_v(n)$. Before the algorithm converges (i.e., the adaptive filter is not yet a very good estimation of the unknown system), $\hat{\sigma}_e(n)$ is larger than $\hat{\sigma}_v(n)$ and the forgetting factor will have lower values, determining fast convergence. This situation occurs when there is a change in the unknown system. The lower value of $\lambda(n)$ will offer also good tracking capabilities. In the other case, when the algorithm reaches the steady-state, $\hat{\sigma}_e(n) \cong \hat{\sigma}_v(n)$ and $\lambda(n) = \lambda_{\max}$, which assures low residual misalignment.

## IV. RESULTS

### A. The forensic speech recovery software based on the recursive least-squares algorithm

A forensic application for recovering speech signals drowned in loud music, based on the principles described in Section II and which uses the RLS algorithm for identifying was initially implemented using Simulink. Its interface is presented in Figure 2. All the parameters can be controlled very ergonomically by turning knobs. Its functioning is detailed onwards.

Before using this software, the user must have at his disposal the two input signals, the mixture recorded in the room and the studio quality masking melody, identified with a music identification software. The studio quality melody is recommended to be processed before loading it into the system from two points of view. First, its sample rate must match the sample rate of the recorded mixture, which is typically 8 kHz since the targeted signal is a speech and speech signals, thanks to their spectral properties, are sampled with 8 kHz in most general-purpose applications. Second, the masking signal in the recorded mixture, in most of the situations, is not temporarily aligned with the studio quality masking melody (i.e., the recorded mixture does not start at the very beginning of the masking melody) and the two input signals should be pre-aligned. This aspect can be handled by the adaptive filter if its length is sufficiently large, but the length of the filter increases the computational complexity. A very important aspect is the fact that the adaptive filter can only delay the input signal to align it with the studio quality melody. The user must take into account this very important necessity. The pre-alignment operation can be done in multiple ways [14] and itself it represents an independent field of research.

After these operations are done, the two signals are ready to be processed by the forensic software. The signals must be available in PCM (Pulse Code Modulation) Wave format. The multimedia file reading blocks, named "From Multimedia File" load the input signals. The next block in the way of the signals is a splitter with the structure presented in Figure 3, which will determine if the signals are routed directly into the adaptive algorithm or each of them will pass through a band pass filter. The splitter is controlled by the rocking switch labeled "Band-pass filtering". If it is set to "On", the signals are routed through the band-pass filter. In the other case, the signals are fed straight into the adaptive algorithm. The parameters of the band-pass filter (i.e., the central frequency and the bandwidth) can be set using two knobs: "Central frequency knob" and "Bandwidth knob". The role of the band-pass filters is to pre-select the spectral band of interest (the band in which the speech signal

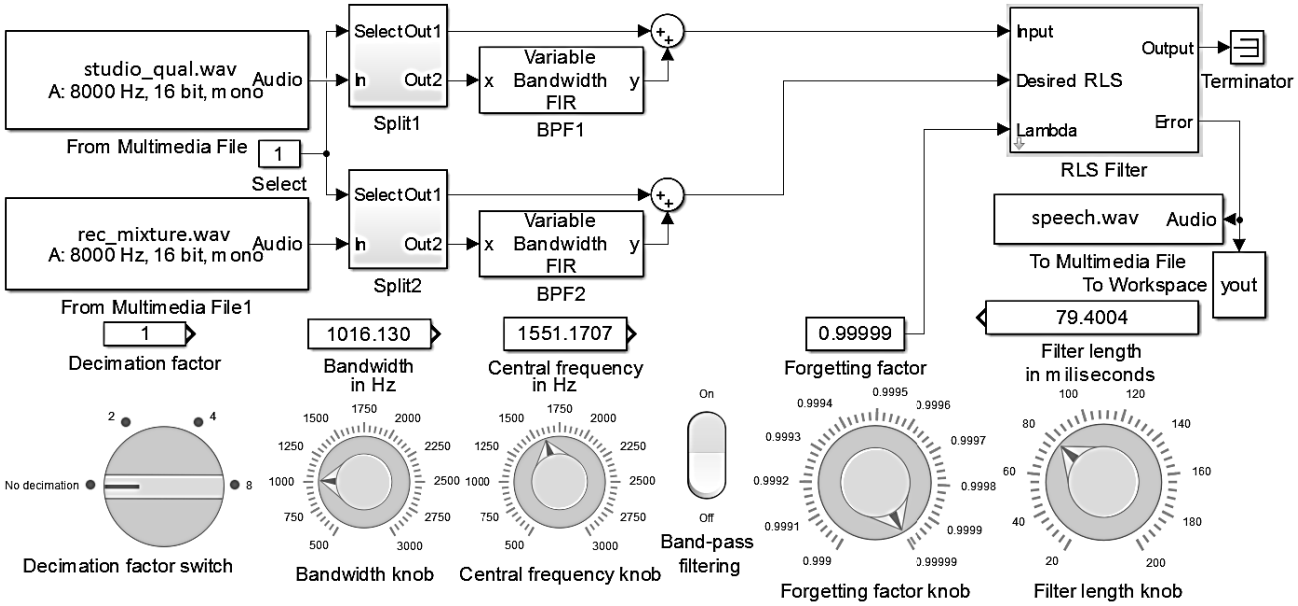


Figure 2. The forensic software for speech recovering based on the RLS algorithm.

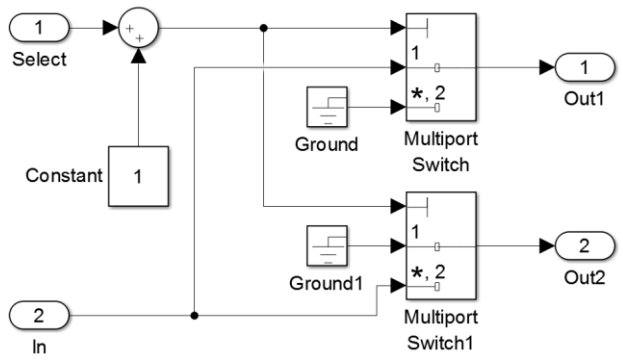


Figure 3. The contents of the Split block that was created to permit the selection of the signals to be fed in the adaptive filter (original signals or band-pass filtered signals).

is concentrated). This way some efforts of the adaptive filter are removed, increasing the efficiency. It is obvious that the two band-pass filters must be identical, or else the effect of an unbalance must be countered by the adaptive filter, increasing the computational effort.

The "RLS Filter" block implements the RLS algorithm described in the previous section. The forgetting factor of the algorithm can be tuned in real time using the "Forgetting factor knob". The recorded mixture will represent the desired signal and the studio quality melody (after pre-processing) will represent the input signal. The speech signal will be recovered as the error signal.

The last very important parameter is the length of the adaptive filter, which can be set using the "Filter length knob". The theory states that the adaptive algorithm will work if its length is equal or greater that the length of the unknown system. It is also intuitively true: if a filter with a length equal to $L_1$ is estimated using an adaptive filter with a length equal to $L_2>L_1$, in the ideal case, the first $L_1$ coefficients of the adaptive filter will be equal to the coefficients of the filter that is estimated and the remaining $L_2-L_1$ coefficients of the adaptive filter will be equal to zero. If $L_2<L_1$, then only $L_2$ coefficients of the unknown filter can be estimated. Depending on the difference of the two lengths and the properties of the unknown filter, a good enough estimation can be obtained, but clearly it cannot be guaranteed. Because the system can work using various sample rates, the length of the adaptive filter is set in milliseconds, to simplify the user's task to compute it in samples for each sampling frequency that is used. The length of the adaptive filter greatly affects the computational complexity. If information about the physical properties of the room (volume, furniture etc.) is known, the length of the filter which will represent the acoustic impulse response of the room can be roughly determined *a priori* using acoustic notions like reverberation time.

The software features a decimation knob named "Decimation factor switch" which, as the name suggests, will decimate both input signals before processing. It is useful when the recorded mixture has a higher sample rate or when a quick test run is desired, to reduce the computational complexity and the processing time consequently.

For testing the software, a speech signal was mixed with a musical signal (which played the role of the masking noise) in −40 dB signal-to-noise ratio. Afterwards, this mixture was filtered using an acoustic impulse response illustrated in Figure 4. The filtered mixture and the original musical signal were used as input signals in the presented software. The RLS algorithm provide very fast convergence rate and good misalignment, visible in Figure 5, which can be observed in the very accurate recovery of the speech signal in Figure 6. In this case changes in the unknown system were not considered.

### *B. The forensic speech recovery software based on the variable forgetting factor recursive least-squares algorithm*

In a real situation, the people that are having the confidential conversation that they try to conceal will not remain perfectly still. Instead, naturally, they can move around affecting the acoustic impulse response of the room. The impulse response of interest is the one that characterizes the propagation of the masking signal. To subtract the musical signal, this impulse response must be accurately

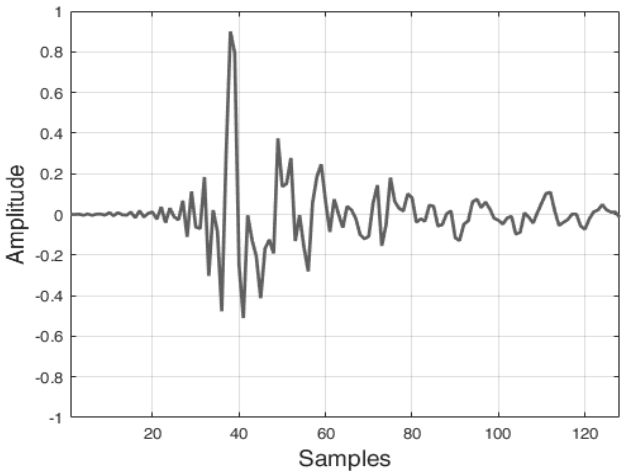


Figure 4. The impulse response used to model the acoustic environment.

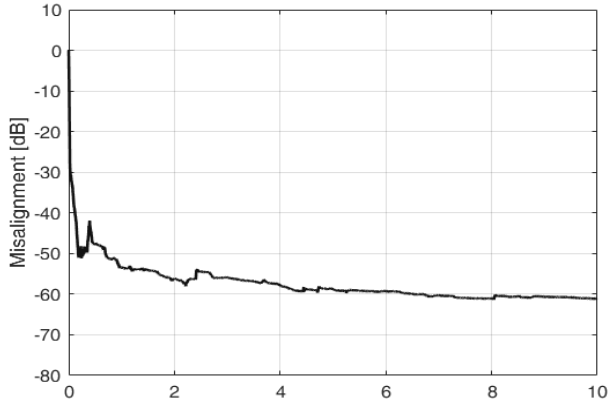


Figure 5. The variation of the misalignment for the RLS algorithm.

estimated. Other events could happen, which also will lead to the modification of the discussed impulse response like the opening or closing of a door, the entrance or exit of a person in/from the room, the opening of a window, etc. In conclusion, a real-world unknown system has a high chance of changing over time.

Testing the RLS based software in such situations confirmed the poor tracking capabilities of the algorithm when the forgetting factor is close to 1, as it can be observed in Figure 7. After 5 seconds, the unknown system changes (the impulse response was shifted with 8 samples). The absolute error means the absolute values of the signal obtained by subtracting the recovered signal from the original (reference) signal. In practice, the reference signal would not be available. It is used here to highlight the performances of the proposed software. The RLS algorithm

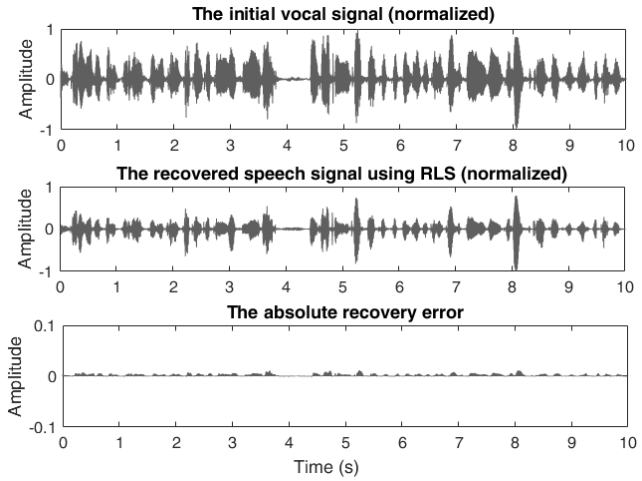


Figure 6. The performances of the RLS algorithm in the given situation.

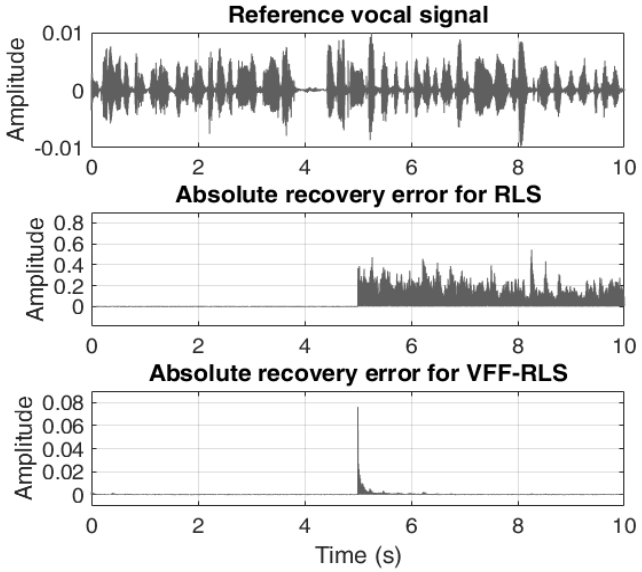


Figure 7. The performances of the RLS and VFF-RLS algorithms in the case in which a change in the acoustic parameters occurs.

gives a very high recovery error after the change, which decreases very slowly. The VFF-RLS algorithm tracks the system change very quickly [15]. The largest absolute value of the recovery error for the VFF-RLS is still much smaller than the error given by the classical RLS algorithm. The

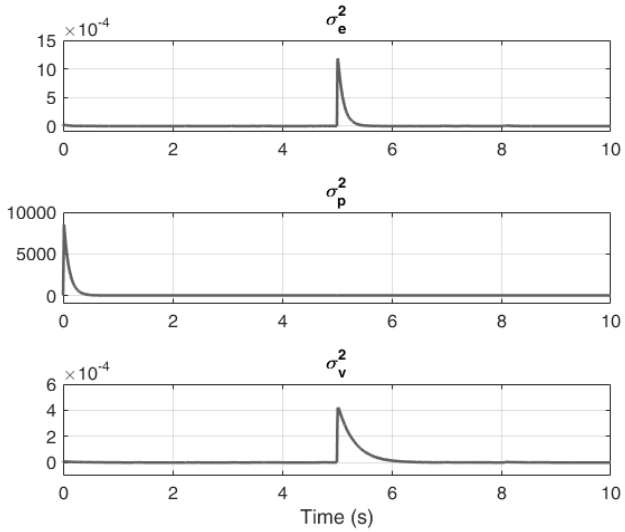


Figure 8. The variation of the VFF-RLS parameters (see the title of each graph for identifying the parameters).

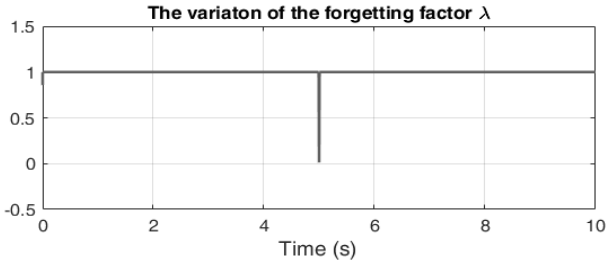


Figure 9. The variation of the forgetting factor.

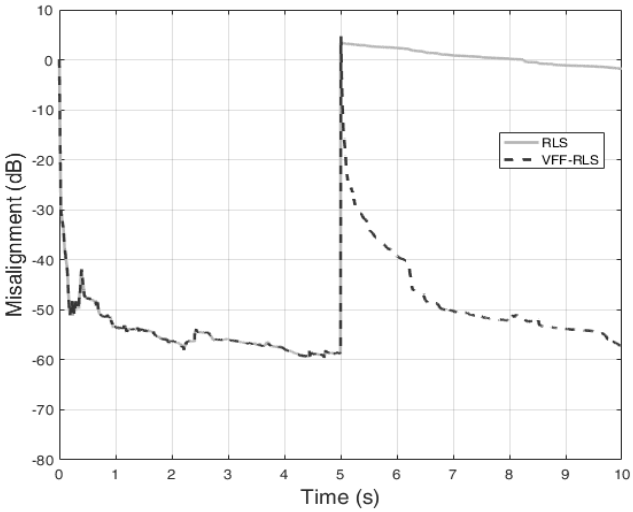


Figure 10. The variation of the misalignment for the two adaptive algorithms.

variation of the VFF-RLS specific parameters and of the forgetting factor can be observed in Figure 8 and Figure 9, respectively. The variation of the misalignment is presented in Figure 10.

### C. The impact of the acoustic environment on the proposed forensic software

The acoustic impulse response greatly affects the performances of the proposed system. One key parameter is its length. Since the unknown system can be considered that it changes frequently, it becomes of importance to determine the length of the unknown system for which the performances are acceptable.

For these experiments, a longer impulse response was used (512 samples long), depicted in Figure 11. The length of the impulse response used in the experiments was progressively increased, starting with 128 samples and incrementing it with 64 samples. It was determined that acceptable quality of the recovered speech signal is achieved when the misalignment reaches –20 dB. The results are illustrated in Figure 12.

The RLS algorithm failed to track the change even in the shortest case and it was not tested for longer impulse responses. In the case of 512 samples, the VFF-RLS would take around one second to achieve the desired misalignment, meaning that the same duration of the recovered signal would be unintelligible. The results in this section could help in taking the decision if placing a microphone in a specific room is worth it or not. The length of the impulse response of the room can be coarsely determined by a trained person if he/she enters the room, by studying the volume of the room and the materials that are placed there. After the inspection, knowledge about acoustics can be used, like Sabine's reverberation time formula detailed in (52), for determining the approximate length of the impulse response:

$$RT_{60} \cong 0.161 \frac{V}{S \cdot a}, \quad (52)$$

where $V$ is the volume of the room, $S$ is the total surface area of the room and $a$ is the average absorption coefficient of the surfaces present in the room.

### D. The performances of the affine projection algorithm in the given situation

The RLS gives very good results in recovering the speech signal if the unknown system does not change in time. It was shown that the VFF-RLS can handle the situations in which there are changes in the system to be estimated if its length is reasonably short. It is very important to remember that RLS and VFF-RLS have a great computational complexity and consequently the processing times can be very long. The affine projection algorithm is a good candidate for decreasing the computational complexity. The software was implemented with this algorithm and similar tests were

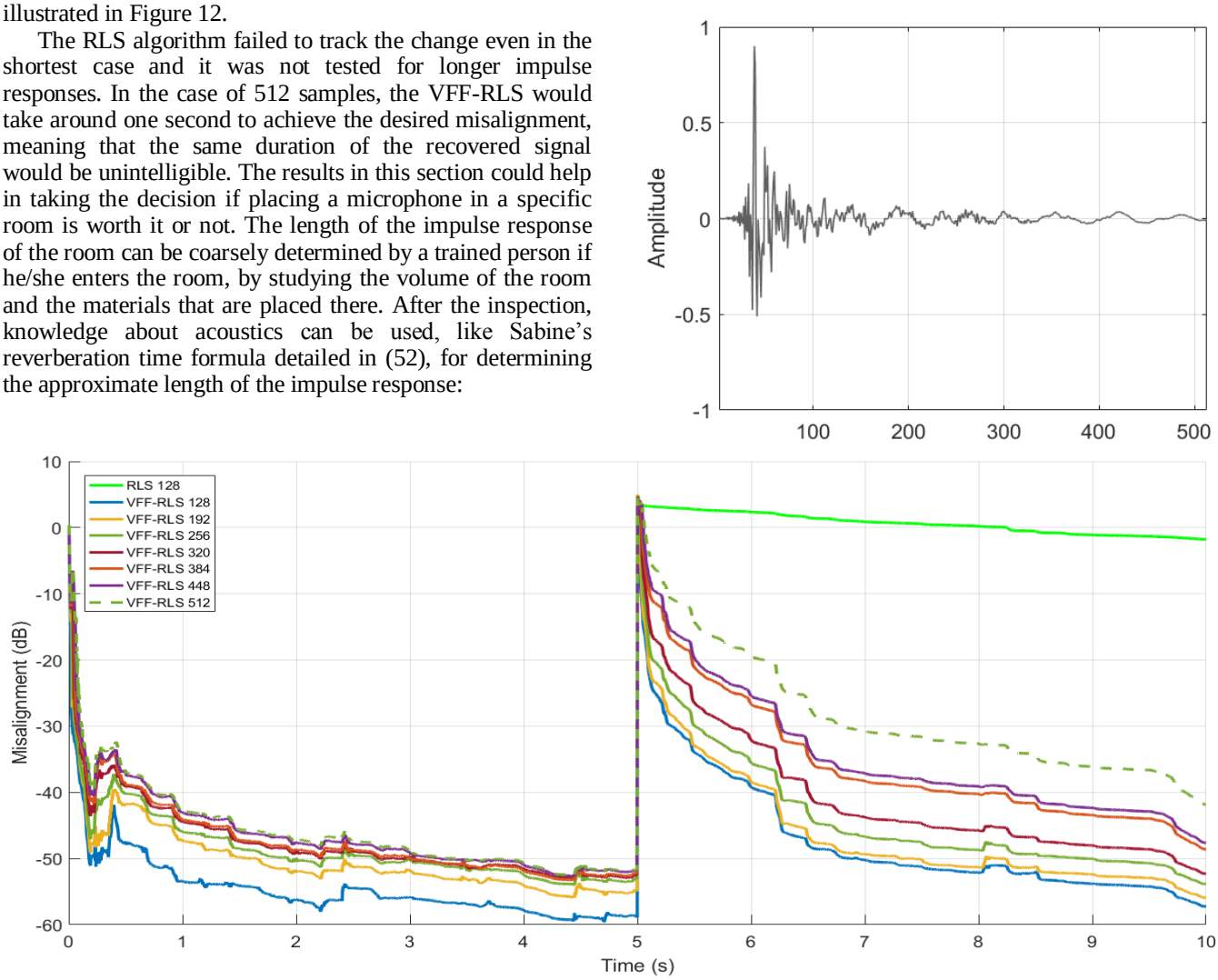


Figure 12. The variation of the misalignment for the two adaptive algorithms with respect to the length of the impulse response.

performed. For the 512 samples long impulse response, the results are illustrated in Figure 13 and Figure 14.

The performances that were obtained qualify APA for solving the investigated situation, but they are lower than in the case of VFF-RLS. It is very important to observe the initial convergence in the case of the two algorithms. The RLS based solutions have a very fast initial convergence, while the convergence of APA is almost the same in the beginning as it is after a system change.

Since the acoustic impulse responses are usually sparse, a proportionate version of APA could show better performances that the classical APA. The MIPAPA was tested and it achieved a better convergence speed than the classical APA, as it can be observed in Figure 15. For comparison, the NLMS algorithm and its proportionate version IPNLMS obtained by using a projection order equal to 1 in MIPAPA, were also tested and their misalignment

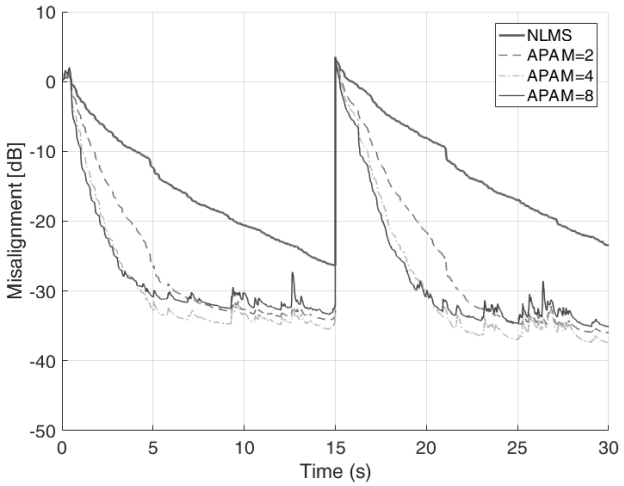


Figure 13. The variation of the misalignment for the estimation of an impulse response with $L$ = 512, $\mu$ = 0.5 and various projection orders $M$, from 1 to 8.

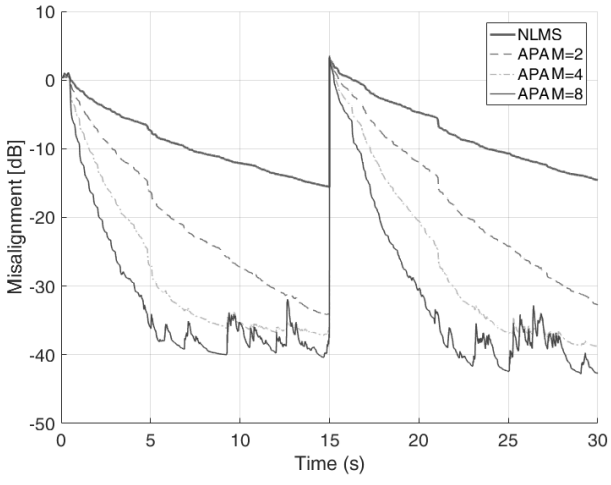


Figure 14. The variation of the misalignment for the estimation of an impulse response with $L$ = 512, $\mu$ = 0.2 and various projection orders $M$, from 1 to 8.

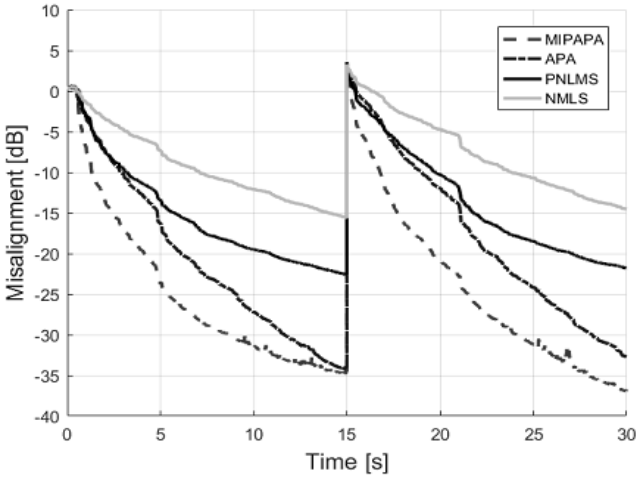


Figure 15. The variation of the misalignment of various adaptive algorithms when estimating the impulse response illustrated in Figure11 with change after 15 seconds ($M$=2, $L$=512, $\mu$=0.2).

was illustrated in the same figure.

## V. CONCLUSION AND FUTURE WORK

This paper describes the importance of multimedia forensic field and presents a practical method for extracting a speech signal drowned in loud music. The core of a forensic software capable of succeeding at such task is a system identification problem, which is a typical adaptive systems application.

Various adaptive algorithms were presented in detail to clearly observe their behavior and understand their suitability to be used in developing the desired forensic software. The unknown system in the stated problem is an acoustic impulse response which is usually sparse. A proportionate variant of the affine projection algorithm (MIPAPA) was also presented because this class could perform very well in such conditions. The importance of system tracking was highlighted and a variable forgetting factor recursive least-squares algorithm was described. It combines the great performances of the RLS algorithm with a good capacity of tracking, without drastically increasing the computational complexity.

A forensic software based on the RLS algorithm was implemented in Simulink to make its interface very easy to use. All the details about the implementation were given and the obtained performances were presented and discussed. The second variant was implemented based on the VFF-RLS algorithm and noticeable performance improvements were observed.

The impact of the acoustic environment on the software's performance was studied. Using the results in this paper, it can be determined if a microphone is worth to be placed in a certain room based on its acoustic properties.

To decrease the computational complexity, the RLS based algorithms were replaced by APA, with an expected (but not dramatic) decrease in performance. Since most acoustic impulse responses are sparse, the MIPAPA was investigated and it was shown that it behaves better than

APA. Since the computational complexity of the two algorithms is similar, the MIPAPA is preferred for solving this problem.

Future work will include the investigation of the method when other types of impulse response changes are considered.

## ACKNOWLEDGMENT

This work was supported by UEFISCDI Romania under Grant PN-II-RU-TE-2014-4-1880.

## REFERENCES

[1] R. A. Dobre, R. M. Udrea, C. Negrescu, and D. Stanomir, "The impact of the acoustic environment on recovering speech signals drowned in loud music," The Sixteenth International Conference on Networks (ICN), Venice, pp. 92-97, April 2017.

[2] A. Ghule and P. Benakop, “A review of LPC methods for enhancement of speech signals,” International Journal of Innovations in Engineering Research and Technology, vol. 2, pp. 1–6, 2015.

[3] P. C. Loizou, Speech Enhancement: Theory and Practice. Second Edition, Boca Raton, CRC Press, 2013.

[4] J. Benesty, S. Makino, and J. Chen, Speech Enhancement. First Edition, Berlin, Springer-Verlag, 2005.

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

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

[7] R. A. Dobre, V. A. Niţă, S. Ciochină, and C. Paleologu, "New insights on the convergence analysis of the affine projection algorithm for system identification," 2015 International Symposium on Signals, Circuits and Systems (ISSCS), Iași, pp. 1-4, July 2015.

[8] V. A. Niţă, R. A. Dobre, S. Ciochină, and C. Paleologu, "Improved convergence model of the affine projection algorithm for system identification," 2017 International Symposium on Signals, Circuits and Systems (ISSCS), Iași, pp. 1-4, July 2017.

[9] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in Proc. IEEE ICASSP, 2002, pp. II-1881–II-1884.

[10] C. Paleologu, S. Ciochină, and J. Benesty, “An efficient proportionate affine projection algorithm for echo cancellation,” IEEE Signal Processing Letters, vol. 17, pp. 165–168, 2010.

[11] 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.

[12] 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.

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

[14] 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.

[15] R. A. Dobre, C. Elisei-Iliescu, C. Paleologu, C. Negrescu, and D. Stanomir, "Robust audio forensic software for recovering speech signals drowned in loud music," 22nd IEEE International Symposium for Design and Technology in Electronic Packaging (SIITME), Oradea, pp. 232-235, October 2016.