Adaptative Echo Cancellation scheme for hands-free systems based on Fast Block Least Mean Square Algorithm

Mestanza Joaquín Matías, Romarís Juan Manuel

Abstract—**In this paper an echo cancellation scheme based on Fast Block Least Mean Square (Fast Block LMS) algorithm is implemented. The paper discusses convergence, stability, parameter selection and coefficient tracking of the mentioned echo cancellation scheme. The performance of the designed scheme is evaluated with different in-room movement simulations so as to guarantee the scheme's functionality and response in diverse acoustic conditions. The performance of the designed scheme is found to be satisfactory.**

Index Terms*—*Adaptive algorithm, Adaptive Filter, Echo Cancellation, Least Mean Square, Fast Block Least Mean Square

I. Introduction

In hands free telecommunication systems talking through a loudspeaker and a receiving microphone, the acoustic echo greatly deteriorates speech quality. This acoustic echo is formed by both the direct path Echo and all secondary path echoes. These secondary path echoes are generated by acoustic wave reflections in different surfaces present inside the room (walls, furniture, people). Moreover, in a real hands free communication situation, furniture can be moved, people can enter the room or even the speaker can wander inside the room making secondary paths variable with respect to time.

In order to solve this problem a system or scheme has to be found that given an input communication signal and an output room filtered signal can estimate main and secondary echo paths in a dynamic or adaptive fashion. In this paper a scheme to solve this pressing problem is proposed and its performance is evaluated taking into account different room impulse simulations.

II. Proposed Solution Scheme

A solution for this previously mentioned issue is the implementation of an adaptive echo cancellation scheme to suppress the acoustic echo signal's magnitude. The principle of this scheme is to estimate the impulse response of interfering acoustic paths using an adaptive filter, generating a pseudo echo and substracting it from the original echo.

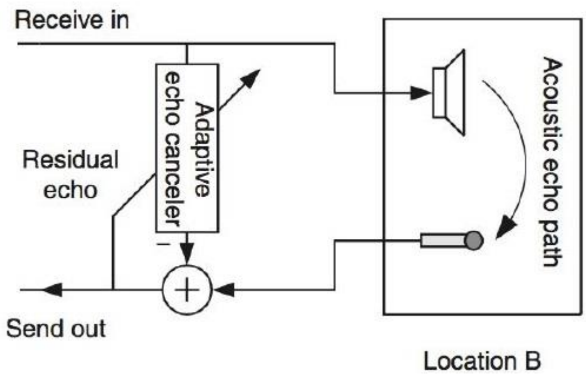

Figure 1. Classic adaptive Echo Cancellation scheme implemented with an adaptive filter

Let $x(n)$ be the transmitted and received signal of a telecommunication system and $d(n)$ the same signal filtered by a given room impulse response, in this adaptive scheme an M-order adaptive filter H is obtained so as to obtain an estimation of the room impulse response $\widehat{d(n)}$. This estimated signal is then subtracted from the original filtered signal to obtain an acoustic echo-free signal as an error $e(n)$ signal. This scheme's error signal can be represented as follows:

$$e(n) = d(n) - \widehat{d(n)} = d(n) - \sum_{i=0}^{M-1} h_i . x(n-i)$$

Once the scheme's output form is known, what is left to determine are the H adaptive filter coefficients. The proposed scheme uses the error signal to update the filter parameters in order to improve its performance. There exist many algorithms that can be used to update adaptive filter coefficients, some of them are discussed in the following section, highlighting the algorithm considered to be more suitable related to the desired functionality.

III. Adaptive Filter update algorithm

Various adaptive filter update algorithms exist, some of which will presented and discussed in the current section of the paper.

A. *Least Mean Square Algorithm*

Least mean squares (LMS) algorithms are a class of adaptive filters which update filter coefficients so as to minimize the mean square error between the desired signal $d(n)$ and the filter's output $\widehat{d(n)}$. This algorithm seeks to converge to the same final value as Wiener Filters without having to solve the Wiener-Hopf equations with matrix inversion. This algorithm equation takes the form of:

$$h_k(n+1) = \widehat{h_k}(n) - \frac{1}{2}\mu.\nabla J_{s,k}(n)$$

Where h_k is the k-th adaptive filter coefficient, μ is the learning rate and $\nabla J_{s,k}(n)$ is the loss function gradient which is reduced to $\nabla J_{s,k}(n) = -2x(n-k)\overline{e(n)}$. Applying this new definition, the algorithm's update equation resumes to:

$$h_k(n+1) = \widehat{h_k}(n) + \mu.x(n-k).\overline{e(n)}$$

Taking into account that this algorithm emerges as a solution for the Wiener-Hopf equation it can be proven that this algorithm's convergence depends on the relation between the learning rate and the input signal autocorrelation matrix. When μ is small, this relationship related to filter coefficient convergence can be shown with the following inequality:

$$0 < \mu < \frac{2}{\lambda_{max}}$$

Where λ_{max} is the greatest eigenvalue of the input autocorrelation matrix. If the algorithm's learning rate meets this inequality, algorithm convergence in the mean is guaranteed.

This algorithm's misadjustment $\boldsymbol{M}$ can be defined as the relation between the stationary minimum loss and the stationary excess mean quadratic error. This can be represented with the following equation:

$$\boldsymbol{M} = \frac{\mu}{2}\mathrm{M}\sigma_x^2$$

It can be seen that this algorithm's stationary quadratic error is related to the filter's length, the learning rate and the input signal power. Given that in the case of hands free communications, the input signal is human speech and its instantaneous power can vary, the algorithm's misadjustment $\boldsymbol{M}$ can also vary accordingly and that may not be desired.

So as to solve this issue, modifications of the LMS algorithm exist that take into account input signal instantaneous power. One of this modifications is the Normalized Least Mean Square Algorithm that will be explained below.

B. Normalized Least Mean Square Algorithm

This LMS variation intends to reduce dependence between input instantaneous power and filter coefficient's update. So as to solve this issue the Lagrange multiplier method was used resulting in the following coefficient update equation:

$$h_k(n+1) = h_k(n) + \frac{\tilde{\mu}}{\|x(n)\|^2+\delta}.x(n-k).\overline{e(n)}$$

It can be seen that in the coefficient update equation input signal power is taken into account. So as to prevent drastic and unexpected changes to occur when input signal power is null or weak another parameter is δ is added. This parameter ensures that the denominator is never too small, causing undesirable big changes in coefficient's update.

Since input signal power is taken into account for filter coefficient update, it is expected that filter convergence and misadjustment will depend less from this power.

Despite this variation solving the dependence of filter convergence speed and input signal power, this LMS variation does not solve another problem that the algorithm presents; large computational complexity for large adaptive filters. So as to solve this issue another LMS variation is explored: Fast Block LMS with Convergence Optimization, that will be explained and discussed below.

C. Fast Block Least Mean Square with Convergence Optimization Algorithm

This basis of this algorithm lies in dividing the input signal in L length blocks and applying the M length adaptive filter to each block. h_k filter coefficients are updated block by block. Relating to computational complexity it can be shown that optimum block length is the adaptive filter's length M.

So as to improve computational complexity this algorithm uses the Fast Fourier Transform to implement the adaptive filtering equation and the correlation between the input signal and the error signal. This algorithm, can be shown with the following diagram:

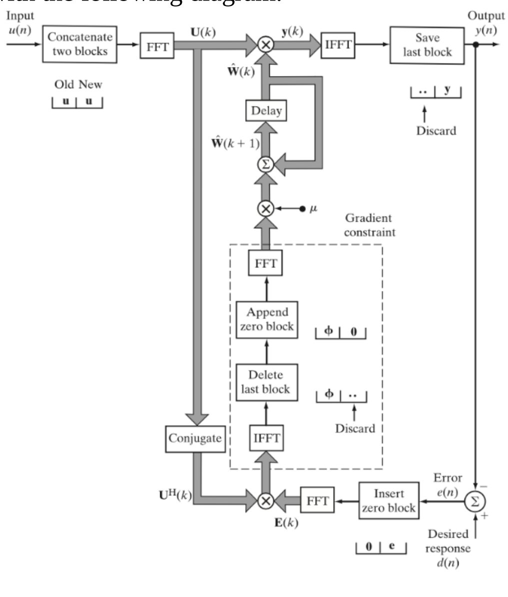

Figure 2. Fast Block Least Mean Square blocks diagram.

This algorithm's complexity ratio in relation to the basic LMS algorithm can be shown below:

$$\frac{FB\ LMS\ complexity}{LMS\ complexity} = \frac{10M\log_2 M + 26M}{2M^2}$$

For example, for an M=4000 filter this equation shows that Fast Block LMS is 307 times faster in relation to the normal LMS algorithm.

So as to improve convergence speed, a μ_i learning rate can be assigned to each i-th frequency bin of the FFT used.

This parameter can be defined as follows:

$$\mu_i = \frac{\alpha}{P_i}$$

Where α is a constant and P_i is an estimation of the instantaneous power of the i-th frequency bin. This power can be estimated with an autoregressive equation as follows:

$$P_i(k) = \gamma P_i(k-1) + (1-\gamma)|X_i(k)|^2 \quad i = 0,1,..,2M-1$$

γ is a forgetting factor whose range of value lies between 0 and 1. This equation implements a 1 order low-pass filter, where the forgetting factor controls the cut-off frequency of the filter.

This algorithm improves computational complexity in relation to the LMS algorithm while also taking into account input signal power in the coefficient update equation, maintaining convergence speed improvements seen in the NLMS algorithm.

Given the needed application and this algorithm's benefits, Fast Block Least Mean Squares with Convergence Optimization was chosen.

D. *Algorithm Implementation*

Given the chosen algorithm (Fast Block LMS with convergence optimization) an existing python package was chosen to be used, this package being adafilt[1]. This package provided and implementation of a Fast Block LMS filter with parameters; M (filter length), L (block length), δ (numerical problem prevention constant), α (step size constant) and γ (forgetting factor).

Parameter selection for this given filter will be explained in the Simulation, Parameter Selection and Results section of this work.

IV. Test Conditions and Room Selection

Performance and audio quality of hands free communications should independent of room conditions, allowing flexibility and user comfort. So as to ensure this it would be desirable to find algorithm parameters and test, algorithm's performance for rooms with different dimensions and impulse responses. Below this selection process is shown.

A. *Test Room Selection*

Before defining room size and other room parameters it was chosen that this scheme's performance was to be tested with simulated room impulse responses. These impulse responses where generated with the python library gpuRIR[2]. This package allowed GPU acceleration calculus and also enabled microphone and speaker movement.

Given this utilized python package room, parameters for the scheme's testing had to be chosen. Firstly, simulations were realized in rectangular rooms. Also, so as to simplify simulations, numbers of audio receivers and sources was kept to 1 (gpuRIR provides usage of arrays of receivers or sources).

In relation to room's surfaces absorption coefficients the following characteristics were chosen. A coefficient of 0.5 was chosen for floors, since it was assumed that the floor was carpeted. For walls and ceiling an absorption coefficient of 0.8 was chosen since it was assumed that these were made with perforated plywood.

Given these absorption coefficients and a defined T60 time the package also had a function that realized the Sabine estimation of the reflection coefficients inside the room. A T60 time of 0.7 seconds was chosen for calculation of these coefficients.

Finally, room size was allowed to vary, testing scheme's performance for different room sizes, and adjusting it so as to minimize the error signal.

So as to visualize the impulse response of these testing rooms, given a room size RIRs for different room positions where plotted together as shown below for a 3x4.5x2.5 m room.

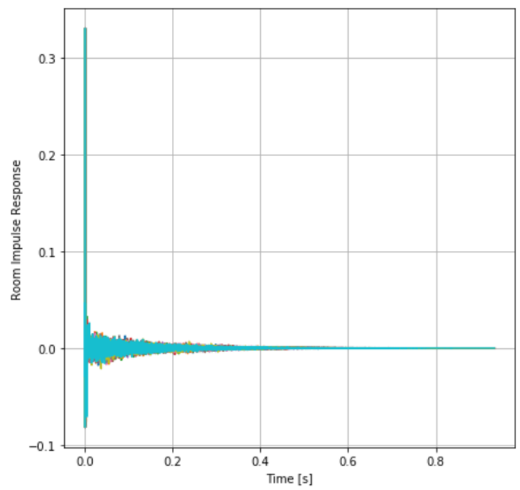

Figure 3. Different position RIRs for 3x4x2.5 m testing room

Finally regarding movement speed inside the room it was considered to realize simulations with a movement speed of 0.1m/s. Also the distance between source and receiver was set to 0.1m similar to a cellular phone length.

V. Simulation, Parameter Selection and Results

After defining testing room characteristics, the scheme was tested utilizing a sample speech audio, sampled at 16kHz. Taking into account the sampling frequency and the T60 time it was firstly considered to utilize a filter with length M=11000 since this filter length would be long enough to capture the complete RIR. The other values chosen for the other parameters of the scheme were; $\delta = 0.0001$, $\alpha = 0.7$ (step size constant) , $\gamma = 0.6$ (forgetting factor). This scheme was tested in different rooms doing 100 simulations changing simulation starting and ending positions.

However, after simulating this scheme results were worse than expected. While the error signal was desired to be as close to zero as possible and uncorrelated to the desired signal, the

[1] Fhchl, 2019, adafilt, https://github.com/fhchl/adafilt

2 Guerra Aparicio, D. 2019, gpuRIR, https://github.com/DavidDiazGuerra/gpuRIR

error signal obtained did not meet these criteria. Below, mean and variance of error signal of 100 simulations using these parameters for a 3x4.5x2.5 m room are shown on par with the scheme's desired and estimated signal for one of the simulations.

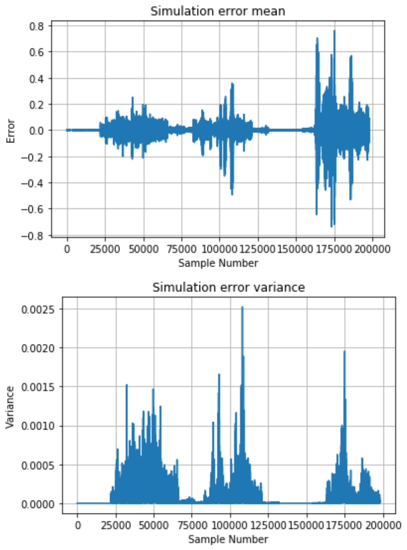

Figure 3. Mean and Variance of total simulation's error signal

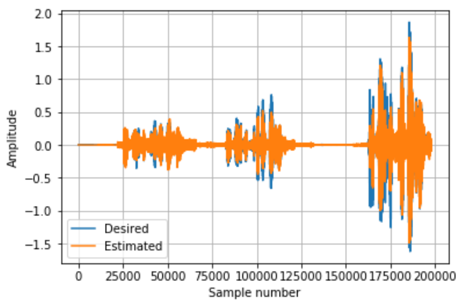

Figure 4. Estimated and desired signals for a particular simulation.

While filter length was maintained the other parameters were changed so as to see if better results could be achieved. If the step size was decreased the algorithm response could not adapt adequately to variations in the input signal. If the step size was increased misadjustment of the algorithm increased and in extreme cases the algorithm diverged.

Taking this into account it was considered that the previous assumption of filter's length was erroneous. Recalling misadjustment equation, a long filter caused an increase in misadjustment, resulting in the scheme's bad performance.

To choose the new filter length it was taken into account that this scheme was to be used for human speech in hands free communications, recalling that human speech had a stationary period of approximately 20ms. Recalling sampling frequency of 16kHz a new filter length of 300 samples was considered. It was considered that reducing filter length would help with misadjustment, while this new filter would only work with stationary input signals.

Regarding the rest of this scheme's parameters the starting values were considered as a starting point; $\delta = 0.0001$, $\alpha = 0.7$ (step size constant) and $\gamma = 0.6$ (forgetting factor). Results for the same room size as before (3x4.5x2.5 m) are shown below:

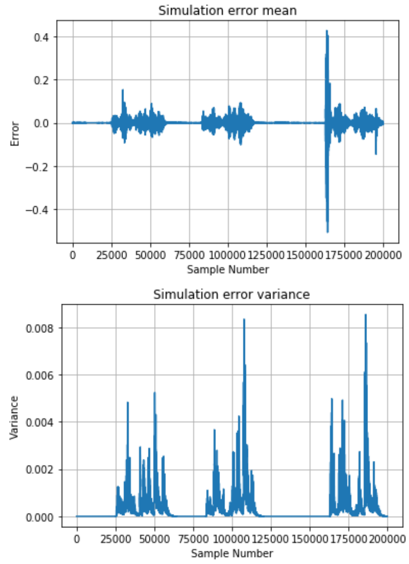

Figure 5. Mean and Variance of total simulation's error signal

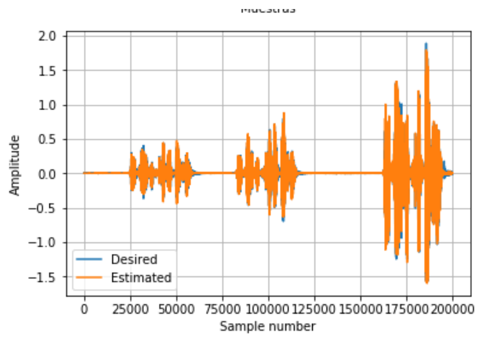

Figure 6. Estimated and desired signals for a particular simulation.

It can be seen that the filter's performance improved, reducing the error signal magnitude while maintaining a small variance. It was considered appropriate to plot filter coefficients for each block of processed input signal as shown below:

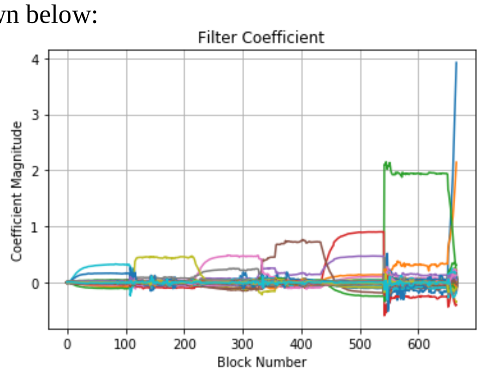

Figure 7. Filter coefficients per Block

Given that a desirable set of parameters for the echo cancellation scheme was found, the scheme was tested in another room simulation (size of 3.3x4x2.5 m) with a different speech sample as shown below:

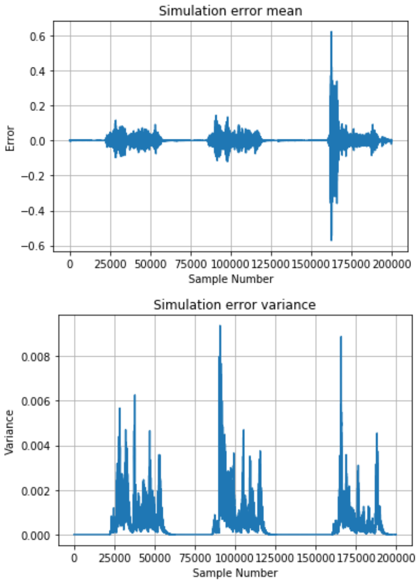

Figure 8. Mean and Variance of total simulation's error signal

Where it can be seen that scheme's performance is maintained.

VI. Conclusions

An echo cancellation scheme utilizing adaptive filters was successfully implemented and tested with adequate simulations. The algorithm utilized for filter coefficient update was Fast Block Least Mean Square with Convergence Optimization with the following parameters; filter length M=300, block length L=300, $\delta = 0.0001$, step size constant $\alpha = 0.7$ and forgetting factor $\gamma = 0.6$.

Given this parameters, scheme's performance was tested through different simulations with different input audios and room conditions, proving to be sturdy and maintaining acceptable performances in these test conditions.