Development of a Prototype Solution for Hearing Problems in Noise in People with Disabilities, Using an Acoustic Beamforming System with a FPGA Card

Fabián Sáenz[1,2](✉), Paúl Bernal[1], Carlos Romero[1,2], and Marcelo Zambrano Vizuete[3]

[1] Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador
{fgsaenz,cpbernal,cgromero}@espe.edu.ec
[2] Universidad Nacional de La Plata, La Plata, Argentina
[3] Instituto Superior Tecnológico Rumiñahui, Sangolqui, Ecuador
Marcelo.zambrano@ister.edu.ec

Abstract. The study of voice signals is an important issue, since a part of society has hearing impairments. This implementation aims to help people with hearing problems, through enhanced voice; using a microphone array with hemispherical methodology broadband beamforming, which can distinguish signals arriving from different directions. A semi-spherical microphone array adapts better to human anatomy because it captures finer sound field.

Keywords: Beamforming · Finite precision · Adaptative algorithm

1 Introduction

The voice is the main form of communication of the human being. The voice process begins with the thought of a message represented abstractly in the brain of the announcer and through the complex process of voice production, finally the information becomes an acoustic signal.

The voice is characterized for being the combination of several frequencies with their corresponding harmonics. However, compared to the full range of human hearing, which goes from 20 Hz to 20 kHz, the voice covers a relatively small range of frequencies between 100 Hz to 6 kHz.

The voice is a non-stationary signal, but it can be assumed through intervals (small sample blocks) as locally stationary. Intervals between 20 and 30 ms are the most suitable for nearly all applications.

The processing of voice signals, for numerous applications, is related to the analysis of their significant characteristics, for example we can mention the pitch (fundamental tone or frequency of the vocal cord's vibration) and its harmonics (peak spectrum of an audible sound). The male voice has a pitch between 100 and 200 Hz and the female voice is typically between 150 and 300 Hz.

M. Botto-Tobar et al. (Eds.): ICAT 2019, CCIS 1195, pp. 359–375, 2020.
https://doi.org/10.1007/978-3-030-42531-9_29

1.1 Initial Mathematical Analysis

An array of sensors is a set of isotropic elements distributed in each geometry, in order to obtain information of the wave fields in the mean in which they propagate. Sensor arrays allow each sensor to have greater directivity and sensitivity. The most commonly used sensor arrays are the Uniform Linear Array (ULA) and the Uniform Circular Array (UCA).

If you have a flat wave that reaches the point $\vec{r}$ at a time t you get Eq. 1. A flat wave is characterized because it has the same amplitude and phase.

$$s(\mathbf{r},t) = Ae^{j(\boldsymbol{\beta}.\mathbf{r}-\omega t)} con \left\{ \beta = \|\boldsymbol{\beta}\| = \frac{2\pi}{\lambda} = \frac{2\pi}{\vartheta/f} \right. \tag{1}$$

Where $\boldsymbol{A}$ is the wave amplitude, $\boldsymbol{\beta}$ the wave vector (propagation), β wave factor (number of waves), λ is the wavelength, ϑ wave propagation speed, f is the frequency.

The flat wave signal in Eq. 2 is considered to arrive at a ULA array, as shown in Fig. 1, at the position $\left(r_x, r_y, r_z\right)$ and up to the origin (0, 0, 0) you have:

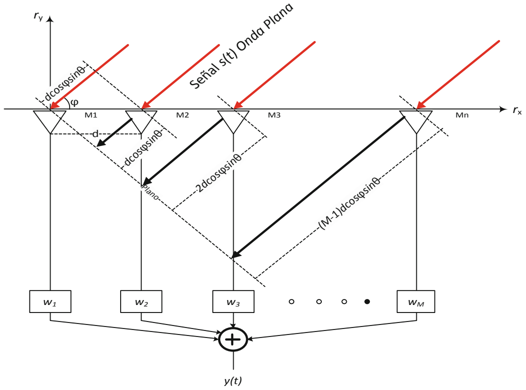

Fig. 1. Uniform Lineal Array ULA. *Source*: (Yang, Cho and Choo 2012)

$$s(\mathbf{r},t) = s(\mathbf{0},t)e^{j\beta\left(r_x \cos\varphi \sin\theta + r_y \cos\varphi \sin\theta + r_z \cos\varphi\right)} \tag{2}$$

Where φ y θ represent the azimuth angle and the elevation angle respectively.

1.2 Signals Processing

Digital processing of voice signals is an interdisciplinary topic that involves phonetics, physiology, acoustics, among other disciplines, in addition to the theory of digital signal processing. This last one has had a great advance thanks to the development of the digital signal processor's technology, being found innumerable applications in the modern life whether it is in the voice coding, recognition or synthesis. For the development of this implementations, it is used the adaptive algorithms of the Householder family, the unconstrained algorithms NLMS (Normalized Least Mean Squares) and the CG (Conjugate Gradient).

Spatial Filtering or Beamforming is a technique that uses an array of sensors; where the main signal is estimated using linear combinations of the different outputs of each of the array sensors, so that unwanted disturbances coming from different directions are attenuated by the phenomenon of spatial directivity or arrangement selectivity (Fig. 2).

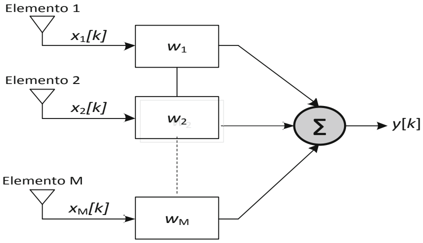

Fig. 2. Narrowband beamforming. Source: (Madisetti 2010)

The study of voice signals is an important issue, since a large part of society has some form of hearing impairment. Hearing disability is considered as those quantitative alterations in a correct perception of hearing. The present implementation seeks to help people with hearing problems, by enhancing the voice; using a semi-spherical arrangement of microphones with the beamforming broadband methodology, that can distinguish signals coming from different directions. It uses a semi-spherical arrangement of microphones that adapts better to human anatomy, and which will allow to better capture the sound field, as studied in the thesis project Acoustic Signal Optimization in a Semi-spherical Microphone Array Using the Beamforming Broadband Methodology.

LCMV adaptive algorithms are divided into three families: Linearly Constrained, such is the case of CLMS (Constrained Least Mean Square), among others. In addition, the GSC structure algorithms (Generalized Sidelobe Canceler) that allows any unconstrained algorithm to be used as constrained. And finally, we have those with Householder structure that through the matrix Q and its Householder's reflectors, they accomplish as the GSC but computationally more efficient.

2 Adaptive Algorithms

LCMV (Linearly Constrained Minimum Variance) adaptive algorithms aim to restrict beamformer in amplitude and output phase to signals coming from one or more directions of no interest. The filter weights are selected in such a way that it minimizes the variance or power of the error subject to a set of linear constraints, as shown in Eq. 3. In Fig. 3 it is shown an adaptive broadband beamforming structure with M sensors y N weights (number of filter coefficients).

$$\min_{\mathbf{w}} \xi[k] \quad subject\, to\, a \quad \mathbf{C}^H\mathbf{w} = \mathbf{f} \tag{3}$$

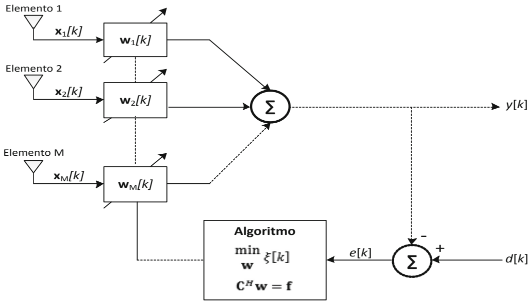

Fig. 3. Adaptive broadband beamforming structure. Source: (Apolinario and de Campos 2014)

Where signal error $e[k]$ is related with the target function $\xi[k] = E\left\{|e[k]|^2\right\} = E\{e[k]e^*[k]\}$, $\mathbf{C}$ is the dimension restriction matrix **MN x p**, being **p** the number of restrictions and **f** the dimension gainance vector **p x 1.**

By imposing linear restrictions, the need for a signal can be often dismissed $d[k] = 0$ LCMV adaptive algorithms are divided into three families: *Linearly Constrained*, with GSC (*Generalized Sidelobe Canceler*) structure, that allows that any *unconstrained* algorithmZZZ be used as constrained and finally there are those of *Householder* structure.

The *Householder* structure is the one with the best performance compared to the other two structures with *unconstrained* NLMS (*Normalized Least Mean Squares*) adaptive algorithms and CG (*Conjugate Gradient*) for the development of this implementation.

The matrix $\mathbf{Q}$ is an orthogonal rotational matrix that is used as the transformation that will generate a vector of modified coefficient $\overline{\mathbf{w}}[k]$ that relates to $\mathbf{w}[k]$.

The matrix $\mathbf{Q}$ should be chosen in such a way that $\mathbf{Q}\mathbf{Q}^H = \mathbf{I}$.

$$\overline{\mathbf{w}}[k] = \mathbf{Q}\mathbf{w}[k] \tag{4}$$

$$\overline{\mathbf{C}}\left\{\overline{\mathbf{C}}^H\overline{\mathbf{C}}\right\}^{-1}\overline{\mathbf{C}}^H = \begin{bmatrix}\mathbf{I}_{\mathrm{p}\times\mathrm{p}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} \tag{5}$$

Then $\overline{\mathbf{C}} = \mathbf{Q}\mathbf{C}$ satisfies $\mathbf{f} = \overline{\mathbf{C}}^H\overline{\mathbf{w}}[k+1]$ and the projection matrix transformation is given by:

$$\overline{\mathbf{P}} = \mathbf{Q}\mathbf{P}\mathbf{Q}^H = \mathbf{I} - \overline{\mathbf{C}}\left\{\overline{\mathbf{C}}^H\overline{\mathbf{C}}\right\}^{-1}\overline{\mathbf{C}}^H = \begin{bmatrix}\mathbf{0}_{\mathrm{p}\times\mathrm{p}} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix} \tag{6}$$

If $\overline{\mathbf{w}}[0]$ is initialized according to

$$\overline{\mathbf{w}}[0] = \overline{\mathbf{C}}\left\{\overline{\mathbf{C}}^H\overline{\mathbf{C}}\right\}^{-1}\mathbf{f} = \mathbf{Q}\mathbf{F} \tag{7}$$

The first p elements of $\overline{\mathbf{w}}[0]$ don't need to be updated. The solution $\overline{\mathbf{w}}[k]$ is based on the transformation $\mathbf{Q}$ to the output signal and therefore the output error is not modified by the transformation.

The HNCLMS algorithm has a slow speed of convergence and its computational cost is low (2p + 3)N − (p^2 + p − 1), so the algorithm becomes desirable for the implementation on devices with limited hardware capabilities.

The HCCG algorithm has a fast convergence and its computational cost (3N^2 + (10 − 4p)N + p^2 − 8p + 2) is higher compared to the HNCLMS algorithm one, so it is desirable when its implementation on a hardware has abundant resources.

2.1 Calculations in Finite Precision

In a digital implementation of an adaptive filtering algorithm, the input data and internal calculations of the algorithm are performed with finite precision, and the cost of digital implementation of an algorithm is influenced by the number of bits (precision) available to perform numerical calculations associated with the algorithm.

The simulation will be done with the Matlab® tool that emulates the operation of finite precision; working with the number of bits with which the hardware works, in this case of 16 bits. The simulation will be performed for 8, 12 and 16 bits.

The theory of adaptive filtering considers that all variables involved in calculations and input signals can be represented in finite precision, thus facilitating mathematical analysis, but the practical implementation in digital signal processors is limited by the number of bits that can be used in internal mathematical calculations and the accuracy with which the values used are stored.

In a digital implementation of an adaptive filter there are essentially two sources of quantization errors to be considered:

- Analogue Conversion – digital
- Finite length of word arithmetic

Analogue Conversion/Digital

The analogue-digital conversion (A/D) can be visualized by an ideal continuous-discrete converter (C/D), followed by a quantizer. The quantization operation will be represented by:

$$\bar{x}[k] = Q\{x[k]\} \tag{8}$$

Where the operator $Q\{.\}$ performs a non-linear rounding operation of the value of $x[k]$ for the nearest quantization level. The spacing of quantization levels defines the step of quantization, uniform or not uniform. For example, a 3-bit uniform quantization is seen in Fig. 4. It is noted that the 3 bits correspond to $2^3 = 8$ levels of quantization. In addition, it is shown on the figure that $x(k) > \frac{7\Delta}{2}$ or $x(k) = \frac{-9\Delta}{2}$, is going to have a saturation. In digital signal processing, we have an output to the A/D converter, value belonging to one of the possible quantization levels, represented according to a numbering system.

The quantization introduces an error (quantization error) defined by:

$$e[k] = x[k] - \bar{x}[k] \tag{9}$$

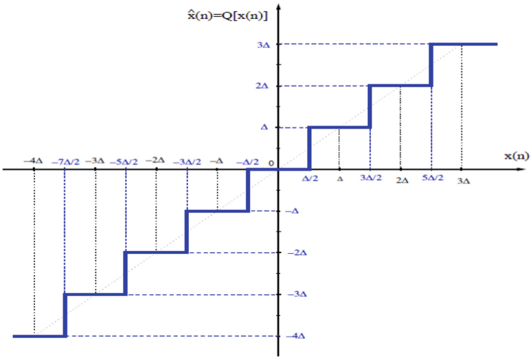

Fig. 4. 3 bits quantification examples

So that, in the case of an example of uniform quantization, there is no saturation (overflow) $-\Delta/2 \leq e(k) \leq \Delta/2$.

The average error is zero, and its variance is given by:

$$\sigma_e^2 = \frac{\Delta^2}{12} = \frac{4X_m^2 2^{-2\#}}{12} = \frac{X_m^2 2^{-2(\#-1)}}{12} \tag{10}$$

Where # is the bits numbers.

2.2 Programming

The Matlab® simulation tool works internally with a high numerical accuracy of 32 or 64 bits. In this project the implementation hardware a FPGA NI myRIO works with 16 bits, so to carry out finite precision analysis it is used the qround function, which emulates the nearest integer.

The ground function will allow to quantify the decimal part of each variable by rounding b bits, as can be seen below.

```
function Vq = qround(V,b)

   Vq = 2^(-b)*round(V*2^b);

end
```

3 Results Analysis

By varying the number of bits between 8, 12 and 16 for the HNCLMS algorithm, the MSE starts to suffer variations, so as the number of bits increases the HNCLMS algorithm presents a better performance, decreasing its MSE, as can be seen in Fig. 5.

Figure 6 shows the behavior of the HCCG algorithm by varying the size of bits from 8, 12 and 16. In the HCCG algorithm, its performance varies with the increasing of bit size, as its MSE starts to decrease, the number of bits increases.

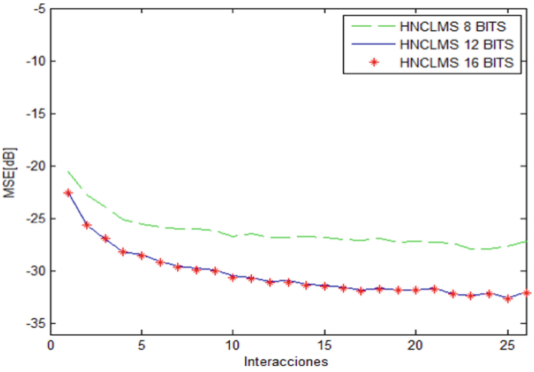

Fig. 5. HNCLMS algorithm for different bits sizes

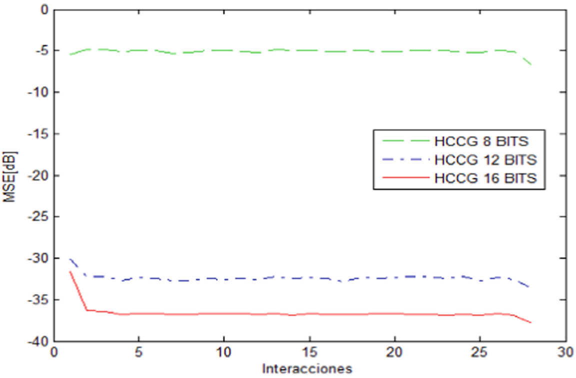

Fig. 6. HNCLMS algorithm for different bits sizes

3.1 Comparison of Results

As mentioned, the cost of digital implementation is influenced by the number of bits available to perform numerical calculations associated with the algorithm. For the algorithms to be implemented, the computational complexity of the algorithms is given by the number of multiplications detailed in Table 1.

Table 1. Computational cost of LCMV algorithms

Algorithm	
NLMS	$(2p + 3)N - (p^2 + p - 1)$
CG	$3N^2 + (10 - 4p)N + p^2 - 8p + 2$

Where N is determined to represent the number of taps and p the number of restrictions. In Table 2, it can be analyzed that the algorithms of lower computational cost are NLMS and the one of higher cost is CG. The structure that has the lowest computational cost is the Householder.

Table 2. MSE in dB for HNCLMS and HCCG algorithms

Algorithm	8 bits	12 bits	16 bits
HNCLMS	−26.0071	−29.6076	−29.6826
HCCG	−5.1213	−32.3649	−36.4347

Table 2 presents a summary of the MSE for the different bit numbers used for the HNCLMS and HCCG algorithms, as can be observed when the number of bits increases the MSE is smaller for each of the algorithms, and because the number of bits is smaller there is an increase in operations and therefore a propagation of the error.

3.2 Lattice Structure

The RLS (Recursive Least Squares) algorithm which, compared to other adaptive algorithms, presents a faster convergence rate and a better response in the presence of disturbance, but with the disadvantage that its hardware implementation is quite difficult due to its high computational complexity and problems with numerical stability.

Theoretically there are several algorithms that solve the problem of least squares recursively. In particular, the Lattice structure allows to reduce computational complexity (in the order of N). Therefore, the LRLS (Lattice Recursive Least Squares) structure is considered a quick implementation for the RLS algorithm problem.

The LRLS algorithm is a cascade structure based on forward and backward linear prediction filters that allows to describe the properties of the input signal, based on the development of least squares to reduce the quadratic error of the mentioned filters, it also updates the reflection coefficients as a function of time.

The performance of LRLS algorithms when implemented with infinite precision arithmetic is identical to those of any RLS algorithm, but with finite precision arithmetic, each algorithm has a different response.

3.3 Recursive Least-Squares Prediction

Forward and backward RLS predictions are essential to derive the order-updating equations inherent in LRLS algorithms. For both cases the results are obtained by following the derivation procedure of the conventional RLS algorithm, since the only distinguishing feature of the prediction problems is the definition of the reference signal $d[k]$. In the case of forward prediction, you have that $d[k] = x[k]$, while the input signal vector has the sample $x[k-1]$ as the most recent data. For backward prediction $d[k] = x[k-i-1]$, where the index i defines the sample of the past to be predicted and the input vector signal $x[k]$ has as the most recent data.

The goal of forward prediction is to predict a future sample given an input sequence, using the current information of the available sequence. For example, you can try to predict the value $x[k]$ using samples passed as $x[k-1]x[k-2]$ through a FIR prediction filter with i + 1 coefficient as shown in the following equation:

$$y_f[k, i+1] = \mathbf{w}_f^T[k, i+1]\mathbf{x}[k-1, i+1] \tag{11}$$

Where $y_f[k, i+1]$ is the output prediction signal

$$\mathbf{w}_f[k, i+1] = \left\{\mathbf{w}_{fo}[k]\mathbf{w}_{f1}[k]\ldots\mathbf{w}_{fi}[k]\right\}^T \tag{12}$$

This is the vector of forward FIR prediction coefficients

$$\mathbf{x}[k-1, i+1] = \{x[k-1]x[k-2]\ldots x[k-i-1]\}^T \tag{13}$$

$\mathbf{x}[k-1, i+1]$ is the available input signal vector. The second variable included in the vector of Eq. (11) shows the dimension of the vector and is required in the order-updating equations of the LRLS algorithm.

The subsequent instant error of the forward prediction is given by:

$$\boldsymbol{\varepsilon}_f[k, i+1] = x[k] - \mathbf{w}_f^T[k, i+1]\mathbf{x}[k-1, i+1] \tag{14}$$

For the RLS formulation of the forward prediction problem, the next prediction weighted error vector is defined as:

$$\boldsymbol{\varepsilon}_f[k, i+1] = \hat{\mathbf{x}}[k] - \mathbf{X}^T[k-1, i+1]\mathbf{w}_f[k, i+1] \tag{15}$$

Where:

$$\hat{\mathbf{x}}[k] = \left\{x[k]\lambda^{1/2}x[k-1]\lambda x[k-2]\ldots\lambda^{k/2}x[0]\right\}^T \tag{16}$$

$$\boldsymbol{\varepsilon}_f[k, i+1] = \left\{\varepsilon_f[k, i+1]\lambda^{1/2}\varepsilon_f[k-1, i+1]\lambda\varepsilon_f[k-2, i+1]\ldots\lambda^{k/2}\varepsilon_f[0, i+1]\right\}^T \tag{17}$$

$$\begin{aligned}&[k-1, i+1]\\ &= \begin{bmatrix} x[k-1] & \lambda^{1/2}x[k-2] & \ldots & \lambda^{[k-2]/2}x[1] & \lambda^{[k-1]/2}x[0] & 0\ 0 \\ x[k-2] & \lambda^{1/2}x[k-3] & \ldots & -\lambda^{[k-2]/2}x[0] & 0 & 0\ 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\ \vdots \\ x[k-i-1] & \lambda^{1/2}[k-i-2] & \cdots & 0 & 0 & 0\ 0 \end{bmatrix}\end{aligned} \tag{18}$$

The error vector $\boldsymbol{\varepsilon}_f[k, i+1]$ can be defined as:

$$\boldsymbol{\varepsilon}_f[k, i+1] = \mathbf{x}^T[k, i+2]\begin{Bmatrix} 1 \\ -\mathbf{w}_f[k, i+1] \end{Bmatrix} \tag{19}$$

The objective function to be minimized in the forward prediction problem is given by:

$$\xi_f^d[k, i+1] = \boldsymbol{\varepsilon}_f^T[k, i+1]\boldsymbol{\varepsilon}_f[k, i+1] \tag{20}$$

$$\xi_f^d[k,i+1]=\sum_{i=0}^{k}\lambda^{k-1}\boldsymbol{\varepsilon}_f^2[l,i+1] \tag{21}$$

$$\xi_f^d[k,i+1]=\sum_{l=0}^{k}\lambda^{k-1}\Big\{x[l]-\mathbf{x}^T[l-1,i+1]\mathbf{w}_f[k,i+1]\Big\}^2 \tag{22}$$

The optimal solution for the coefficient vector is:

$$\mathbf{w}_f[k,i+1]=\mathbf{R}_{Df}^{-1}[k-1,i+1]\mathbf{p}_{Df}[k,i+1] \tag{23}$$

Where $\mathbf{R}_{Df}[k-1,i+1]$ is equal to deterministic correlation matrix $\mathbf{R}_D[k-1]$ of order ***i* + *1*** y $\mathbf{p}_{Df}[k,i+1]$ is the deterministic vector of cross-correlation between $x[l]$ and $\mathbf{x}[l-1,i+1]$.

The minimum value of $\xi_f^d[k]$ is given by:

$$\xi_{f\,min}^d[k,i+1]=\sigma_f^2[k]-\mathbf{w}_f^T[k,i+1]\mathbf{p}_{Df}[k,i+1] \tag{24}$$

Combining Eqs. 23 and 24 you obtain:

$$\mathbf{R}_D[k,i+2]\begin{bmatrix}1\\-\mathbf{w}_f[k,i+1]\end{bmatrix}=\begin{bmatrix}\xi_{f\,min}^d[k,i+1]\\0\end{bmatrix} \tag{25}$$

Where $\mathbf{R}_D[k,i+2]$ is equal to $\mathbf{R}_D[k]$ of dimension i + 2. The previous equation refers to the deterministic order correlation matrix i + 2.

The purpose of the return prediction is to generate an estimate of a past sample of a given input sequence, using the current available sequence information

$$y_b[k,i+1]=\mathbf{w}_b^T[k,i+1]\mathbf{x}[k,i+1] \tag{26}$$

Where $y_b[k,i+1]$ is return prediction output signal, and

$$\mathbf{w}_b^T[k,i+1]=\{w_{b0}[k]w_{b1}[k]\ldots w_{bi}[k]\}^T \tag{27}$$

It is the return FIR prediction coefficient vector. The return instant prediction error is given by:

$$\varepsilon_b[k,i+1]=x[k-i-1]-\mathbf{w}_b^T[k,i+1]\mathbf{x}[k,i+1] \tag{28}$$

The weighted backward error vector is represented by:

$$\boldsymbol{\varepsilon}_b[k,i+1]=\hat{\mathbf{x}}[k]-\mathbf{X}^T[k-1,i+1]\mathbf{w}_b[k,i+1] \tag{29}$$

Where:

$$\hat{\mathbf{x}}[k-i-1]=\Big\{x[k-i-1]\lambda^{1/2}x[k-i-2]\ldots\lambda^{[k-i-1]/2}x[0]0\ldots0\Big\}^T \tag{30}$$

$$\boldsymbol{\varepsilon}_b[k,i+1]=\Big\{\varepsilon_b[k,i+1]\lambda^{1/2}\varepsilon_b[k-1,i+1]\ldots\lambda^{k/2}\varepsilon_b[0,i+1]\Big\}^T \tag{31}$$

$$\mathbf{x}[k, i+1] = \begin{bmatrix} x[k] & \lambda^{1/2}x[k-1] & \ldots & \lambda^{(k-1/2}x[1] & \lambda^{(k)/2}x[0] \\ x[k-1] & \lambda^{1/2}x[k-2] & \ldots & -\lambda^{[k-2]/2}x[0] & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x[k-i] & \lambda^{1/2}[k-i-1] & \cdots & 0 & 0 \end{bmatrix} \tag{32}$$

The error vector $\boldsymbol{\varepsilon}_b[k, i+1]$ can be defined as:

$$\boldsymbol{\varepsilon}_b[k, i+1] = \mathbf{X}^T[k, i+2]\left\{ \begin{matrix} -\mathbf{w}_b\lfloor k, i+1 \rfloor\} \\ 1 \end{matrix} \right\} \tag{33}$$

The objective function to be minimized in the return prediction problem is shown as:

$$\xi_b^d[k, i+1] = \sum_{l=0}^{k} \lambda^{k-1}\left\{x[l-i-1] - \mathbf{x}^T[l, i+1]\mathbf{w}_b[k, i+1]\right\}^2 \tag{34}$$

The optimal solution for the coefficient vector is:

$$\mathbf{w}_b[k, i+1] = \mathbf{R}_{Df}^{-1}[k, i+1]\mathbf{p}_{Df}[k, i+1] \tag{35}$$

Where $\mathbf{R}_{Db}[k, i+1]$ is equal to deterministic correlation matrix $\mathbf{R}_D[k]$ of order $i+1$ and $\mathbf{p}_{Db}[k, i+1]$ is the deterministic vector of cross-correlation between $x[l-i-1]$ and $\mathbf{x}[l, i+1]$.

The minimum value for $\xi_b^d[k]$ is given by:

$$\xi_{bmin}^d[k, i+1] = \sigma_b^2[k] - \mathbf{w}_b^T[k, i+1]\mathbf{p}_{Db}[k, i+1] \tag{36}$$

Combining Eqs. 35 y 36, you get:

$$\mathbf{R}_D[k, i+2]\begin{bmatrix} -\mathbf{w}_b\lfloor k, i+1 \rfloor \\ 1 \end{bmatrix} = \begin{bmatrix} o \\ \xi_{bmin}^d[k, i+1] \end{bmatrix} \tag{37}$$

Where $\mathbf{R}_D[k, i+2]$ is equal to $\mathbf{R}_D[k]$ of dimensions $i+2$. The above equation refers to the deterministic order correlation matrix $i+1$.

4 Algorithm Implementation

A microphone serves as an acoustic sensor to record audio signals and monitor sounds levels. MEMS ADMP504 microphones were used for this project. The ADMP504 consists of a MEMS microphone element, an impedance converter and an output amplifier. The sensitivity specification makes it an excellent choice for both near field and far field applications. The ADMP504 has very high signal/noise ratio and extended broadband frequency response, resulting in a natural sound with ba high intelligibility (Figs. 7 and 8).

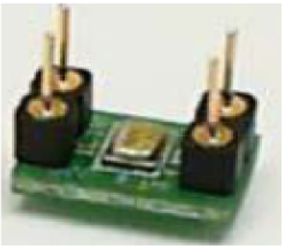

Fig. 7. Microphone MEMS ADMP504

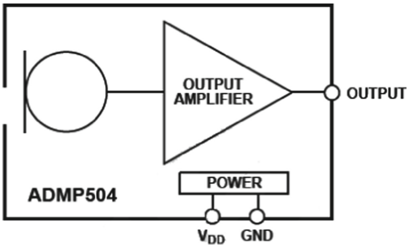

Fig. 8. Functional microphone MEMS ADMP504 block diagram

4.1 Data Acquisition

The response of the microphones used in this project was evaluated. The array of sensors must have an isotropic behavior.

For the analysis we performed a ULA arrangement that complies with the Nyquist theorem, where the sampling frequency is greater than or equal to twice the maximum component of signal frequency and thus avoid aliasing (overlap) frequently. The overlap occurs when the separation between the microphones is not correct, causing the microphones to be unable to differentiate the incoming signals from different directions.

Based on the Nyquist criterion, 4 microphones were used equidistant between them by 0.1 m. in front of a source that casts AWGN signals that by its mathematical characteristics its spectrum covers all frequencies spaced to 1 m. relative to the microphone array and 90 to a microphone to which we will nominate M4, as shown in Fig. 9.

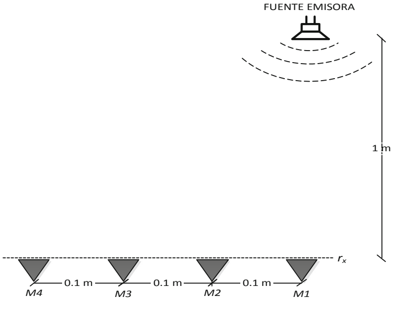

Fig. 9. Microphone ULA arrangement

4.2 NI myRIO 1900 Card Technical Specifications

It has aCortex™-A9 dual core processor of real-time performance and customized I/O, taking advantage of the default FPGA configuration, which they can customize according to the projects, by means of their components internal data, access to transparent software and resource library. Programable with Labview, C or adaptable.

4.3 Construction of the MEMS Microphone Interface Circuit

The amplifier circuit for microphones is constructed as seen in Fig. 10, where the microphone output signal will be connected directly to the analog MXP inputs of the NI myRIO card.

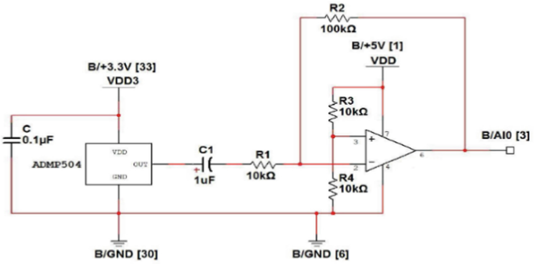

Fig. 10. MEMS microphone circuit with Analog Input (AI)

4.4 Interface with NI Labview

Data acquisition will be done from the FPGA card. A FPGA.vi file is generated where analog inputs are created for the 4 microphones that are connected (Fig. 11).

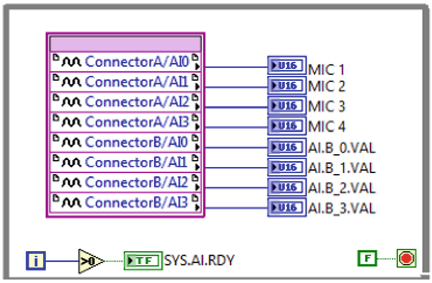

Fig. 11. Analogous entries for data acquisition in the FPGA

The data acquisition must be executed at 8000 Hz, which is no more than the voice encoding process, using the Timed Loop icon that executes each loop iteration in the period specified in this case.

Data acquired at a constant rate (125 µs) will be displayed using the Waveform Chart icon, which is a numerical indicator that maintains a data history.

$$\text{history length} = 8000\,\text{samples/s} \times 40\text{ s} = 320,000\ \text{samples} \tag{38}$$

4.5 Frequency Microphone Response Interface

Frequency microphone responses are presented in Fig. 12, where the behavior of microphones is seen to be different. In order to compensate for the differences between the microphones used in their manufacture, the system identification structure is applied and the M1 is used as a reference.

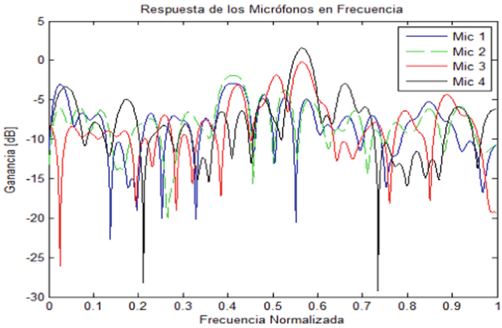

Fig. 12. Frequency microphone response for a ULA array

Because the signals were not acquired in an anechoic cabin (a room designed to absorb all reflections produced by acoustic or electromagnetic waves on any of the surfaces forming it (floor, ceiling and side walls), in turn, the camera is isolated from the outside of any external noise source or sound influence) is executed the RLS algorithm to obtain the same reference signal that is acquired by taking the white noise signal as a signal desired and the input signal of the M1 microphone, this acquired signal is the reference signal for M2, M3 and M4 in the system identification structure.

By offsetting the delay of the signals, which occur due to the hardware, processing and propagation of the signal, it is seen in Fig. 13 that the response of the microphones in frequency has changed and the behavior between them is similar. The identification structure of the system generates the coefficients with which a transfer function is estimated to perform the equalization of the channel and thus compensate for the linear distortion caused by the channel.

NI myRIO is an embedded hardware specifically designed to develop advanced engineering systems more quickly. NI myRIO has a fully programmable dual-core ARM Cortex-A9 processor that runs a real-time OS, as well as a FPGA. The system identification structure generates the coefficients with which a transfer function will be estimated to perform the equalization of the channel and thus compensate for the linear distortion caused by the channel.

The MSE is further illustrated in Fig. 14 where the two algorithms are represented with all interactions, it can be observed how the algorithms converge reaching the HNCLMS algorithm an average MSE of −28.8352 dB, while the HCCG algorithm reaches an MSE lower than −33.8471 dB. Figure 14 shows the beampattern of the HNCLMS and HCCG algorithms, according to this figure the angle obtained matches the input signal of 90°. The two algorithms have a high resolution.

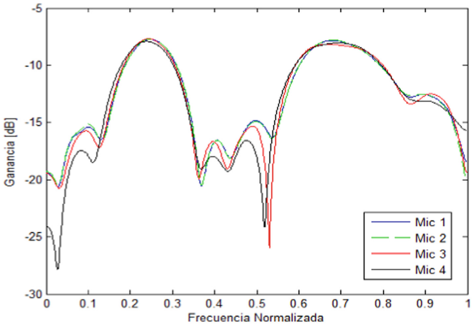

Fig. 13. Frequency microphone compensated response for a ULA arrangement

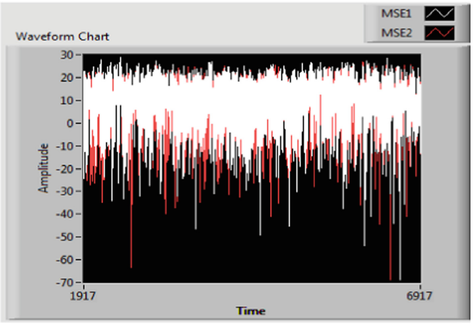

Fig. 14. Comparison of Householder algorithms to MSE

The HCCG algorithm has a high performance and this can be corroborated when comparing the beampattern, as can be seen in Fig. 15, in which the gain of the HCCG algorithm is acceptable.

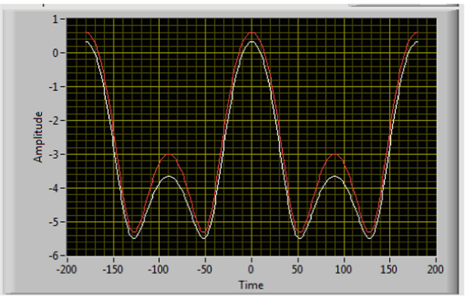

Fig. 15. Comparison of Householder algorithms to the beampattern for a semi-spherical array

5 Conclusions and Recommendations

The theory of sensors arrangements shows that these sensors must have an isotropic behavior, when performing the analysis it was found that in practical cases does not comply; this is typical of their manufacture, attributed to the physical-chemical properties of the materials used, therefore their behavior was approximated by the equalization of the same, taking as reference Mic 1.

The execution time of the HCCG algorithm is 8 ms and the HCNLMS algorithm is 6 ms; due to the acquisition of data, processing of signals, algorithms executed in the Matlab Math Scrip.

The NLMS and CG algorithms present a similar value in their MSE and a good execution when performing graph analysis. However, it should be considered that the HCNLMS algorithm has a shorter run time because its algorithm has a smaller number of operations.

It is recommended to extend the study of the signals acquired in a suitable environment, since in the present research AWGN signals are used but these signals were not acquired in an anechoic cabin, in this way the final result will be improved.

References

1. Agilent EEsof EDA Software: Herramienta Advanced Design System. Recuperado el 7 de Agosto de 2013 (2000). http://www.home.agilent.com/en/pc-1297113/advanced-design-system
2. Apolinário, J., de Campos, M., Bernal, C.: The constrained conjugate gradient algorithm. IEEE Signal Process. Lett. **7**(12) (2000)
3. Papoulis, A., Pillai, U.: Probability, Random Variables and Stochastic Processes. International Edition. McGraw-Hill, New York (2002)
4. Apolinário, J.: Processamento Digital de Sinais. Bookman Editora, São Paulo (2003)
5. Apolinário, J.A.: QDR-RLS Adaptive Filtering. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-09734-3
6. Apolinario, J., de Campos, M.: Instituto Militar de Engenharia. Recuperado el 21 de 12 de 2014 (2011). http://aquarius.ime.eb.br/~apolin
7. Benesty, J., Chen, J., Huang, Y.: Microphone Array Signal Processing. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78612-2
8. Bernal Oñate, C.: Principios y Aplicaciones de CDMA "Code Division multiple access" con implementación de algoritmos para la detección de multiusuarios. Tesis, Quito (2000)
9. Bernal, P., Sáenz, F., Romero, C.: Análisis de Señales Acústicas. Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE (2014)
10. Caisapanta, A.: Optimización de las señales acústicas en un arreglo semiesférico de micrófonos utilizando la metodología de beamforming de banda ancha. ESPE, Quito (2015)
11. Chandran, S.: Adaptive Antenna Arrays Trends and Applications. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-662-05592-2
12. de Campos, M., Werner, S., Apolinário, J.: Constrained adaptation algorithms employing householder transformation. IEEE Trans. Signal Process. **50**(9), 9 (2002)
13. Diniz, P.: Adaptive Filtering: Algorithms and Practical Implementation, 4th edn. Springer, Rio de Janeiro (2013). https://doi.org/10.1007/978-1-4614-4106-9
14. Golub, G., Van Loan, C.: Matrix Computations, 4th edn. The Johns Hopkins University Press, Baltimore (2013)

15. Gundersen, K., Hakon Husoy, J.: Preconditioner structures for the CLMS adaptive filtering algorithm. In: IEEE (2006)
16. Huang, Z., Balanis, C.: Adaptive beamforming using spherical array. In: IEEE (2005)
17. Liu, W., Weiss, S.: Wideband Beamforming: Concepts and Techniques. Wiley, Chichester (2010)
18. Madisetti, V.: The Digital Signal Processing Handbook, 2nd edn. Taylor and Francis Group, LLC, Boca Raton (2010)
19. Medina, C.A., Rodríguez, C.V., Apolinário, J.A., León, R.D.: Implementación de un arreglo superdirectivo de micrófonos con múltiples líneas de retardo. Quito (2000)
20. Monzingo, R., Haupt, R., Miller, T.: Introduction to Adaptive Arrays, 2nd edn. SciTech Publishing Inc, Raleigh (2011)
21. Rabiner, L., Schafer, R.W.: Theory and Application of Digital Speech Processing. Prentice Hall, Englewood Cliffs (2009)
22. Sansaloni, T., Valls, J.: Simulador de Sistemas Digitales de Preisión Finita (2000)
23. Valle, S. d.: Manual Práctico de Acústica, Terceira edn. Música & Tecnología, Rio de Janeiro (2009)
24. Van Veen, B., Buckley, K.: Beamforming Techniques for Spatial Filtering. CRCnetBASE (2000)
25. Werner, S., Apolinário, J., de Campos, M.L.R.: On the Equivalence of the Constrained RLS and the GSC-RLS Beamformers. Helsinki University of Technology, Instituto Militar de Engenharia, and Universidade Federal do Rio de Janeiro (s.f.)