MEE10:96

Direction of arrival estimation – A two microphones approach

Carlos Fernández Scola
María Dolores Bolaños Ortega

Master Thesis

This thesis is presented as part of Degree of Master of Science in Electrical Engineering

Blekinge Institute of Technology

September 2010

Blekinge Institute of Technology
School of Engineering
Department of Signal Processing
Supervisor: Dr. Nedelko Grbic
Examiner: Dr.Nedelko Grbic

Blekinge Tekniska Högskola
SE–371 79 Karlskrona
Tel.vx 0455-38 50 00
Fax 0455-38 50 57

Abstract

This thesis presents a solution to the problem of sound source localization. The target lies in getting the direction of a source by capturing its sound field with two omni-directional microphones. The solution allows the source to be either located at a fixed location or it can be in motion within a specified area. The sought direction is represented by the angle existing between a reference line and the line where the speaker stands.

Considering that the sound signal reaches each microphone at different time instants, corresponding to propagation in different paths, it can be assumed that the captured signals in the two microphones have a time difference of arrival (TDOA). The solution proposed in this thesis first estimates this time difference, and then by using trigonometry, the desired angle is calculated.

Acknowledgments

We would like to thank Dr. Nedelko Grbic for all the help provided throughout this thesis. We believe that his guidelines have been vital and without them, this work could not have been achieved properly.

We would also like to thank our families and friends for all the support they have given us during the last years.

Contents Page

List of Tables Page

List of Figures Page

1 Introduction

Sound location is the science of determining the direction and distance of a source with the only help of its sounds [1]. Getting these two parameters allow an accurate localization of a speaker which is crucial for a certain number of applications. Nevertheless, obtaining the distance is not always necessary since most of these applications only need the direction to be efficient. This enables to design less-complex systems, without giving up a good performance. For example on videoconference, the system does not need to calculate the distance in which the source is emitting sounds. With the knowledge of the direction exclusively the camera can focus the speakers [2]. Other applications that require only the calculation of the direction are the audio surveillance systems. This kind of systems, used for intrusion detection [3] or gunfire location [4], determines the direction for locating the source, but ignores the distance.

This project focuses on a simple way to determine the direction of a source with the help of two microphones. In this thesis one of these methods is presented and tested in order to check its reliability. The system designed to process the signals was programmed with the computing tool MatLab.

This report is the summary of all the work that has been done to accomplish the system. After a short explanation of the background works, the whole system is explained. In section 3, the physical issue that motivated this thesis is presented as well as a solution based on geometry. Then the conversion from analog to digital (with all the practical restrictions it implies) is explained in section 4. Section 5 shows in detail the electrical system designed to solve the physical problem. After that, all the simulations that were carried out are presented (Section 6) and analyzed (Section 7). Finally the conclusions (Section 8) and future work (Section 9) are exposed.

2 Background

The human being has the capability of locating sounds. Actually the system formed by the ears and the brain can by itself, detect a signal, process it and determinate where a sound comes from. Thank to the shape of the ears and to the delay caused by the sound propagation, the brain can locate the source within a certain range of failure [5]. From XIX century until nowadays, men had intended to build devices with this human feature. In order to imitate the human ears different microphone-array systems have been implemented.

However, sound location can be found in the nature. Several animals use sound location to replace direct eye vision. For instance animals like bats or dolphins perform a technique called echolocation. By emitting sound and processing the echoes caused by the reflections, these animals can perceive their environment [6]-[7]. It was proved that human beings can also develop this capability and that it can be applied to blind people [8].

As many other inventions all along the history, one of the first purposes of sound location was for military applications. A method called *sound ranging* was developed. It consists on obtaining the coordinates of a hostile by the sound of its gun firing [9]. This way, even without direct vision, the enemy could be detected. This technique started to be used in World War I and has been used during the whole XX century. When new technologies like gun detective radars came out, sound ranging stopped to be useful. Even so, some armies still use it in their operations [10].

Besides the military application, many other studies about sound location have been carried out. Some of them focus on noise or echo cancellation [11]-[13]. These can be very important, depending if the system is designed for open or closed environments. In other works, the aim is to minimize the error [14] and thus make algorithms more efficient.

But in most of them, new methods or applications are proposed. The differences between them are commonly the number of microphones used [15]-[19] or the way the signals are processed after the sound is captured [20]-[23]. Despite of that, most applications of sound location systems are installed into robots designed to have human behavior [24]-[26].

Furthermore it is noteworthy that some radio localization systems have been implemented. In these the principle is the same as in sound location but the signals are radio waves instead of acoustic waves [27].

3 Physical preliminaries

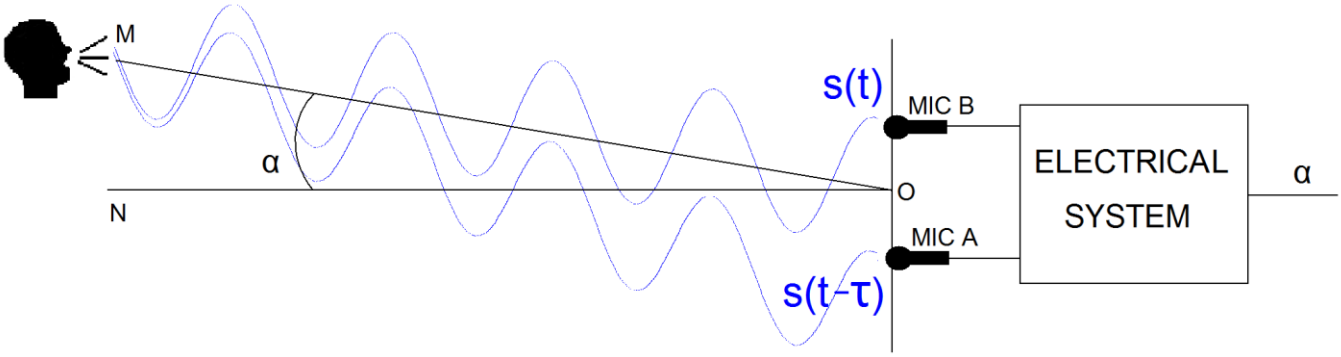


Figure 1: Description of the physical setup.

3.1. General approach

Figure 1 illustrates the entire system, including the physical property of time difference of arrival at two microphones receiving a sound wave propagating from a human speaker. The output of the electrical system is to estimate the real physical angle, α, as accurately as possible.

Considering a source (M) located in front of two microphones, the target is to determine the direction of arrival of its sounds. It is needed to fix the origin from which the measurements will be performed. The microphones are placed in a fixed position and separated by a certain distance. Then the origin was set in the middle of the microphones. Considering the orthogonal line to the microphone axis at the origin (ON), the angle α is defined by the separation angle between this line and line (OM). From now on, the term "direction" refers to the angle α where the speaker is located.

Observing the example shown in Figure 1, the speaker stands closer to MIC B than MIC A. Thus, the sound traveling through the air from the speaker to the microphones reaches first MIC B and then MIC A. The time elapsed between these two moments is denoted τ.

The sound signal can be represented as an analog signal s(t). Theoretically, the signals captured by both microphones would be equal in amplitude and would only be delayed a time

τ. Hence, considering s(t) the signal captured by MIC B, it can be affirmed that the signal captured by MIC A would be s(t- τ).

Figure 2 shows the different positions of the speaker which can be handled by this system. These positions can vary from -90° to +90°.

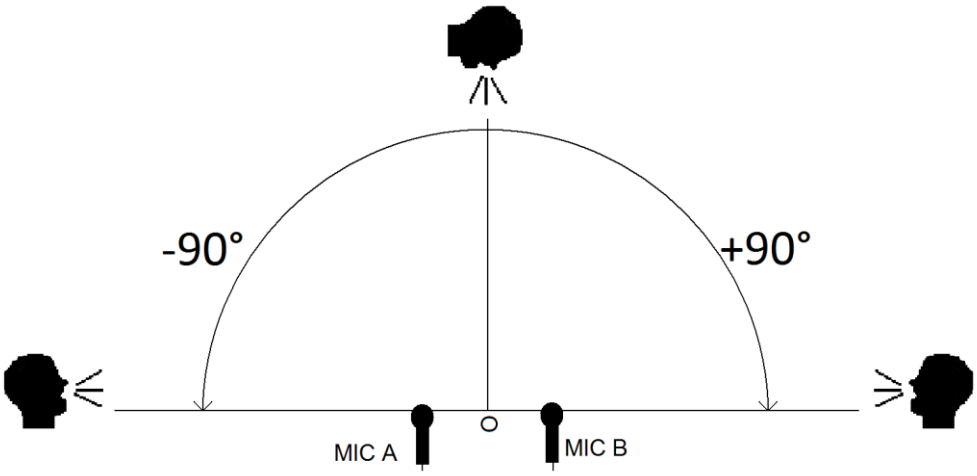


Figure 2: Range of directions in the front semicircle.

Once both signals are captured, they are processed to estimate this time delay. Then, with the help of trigonometric calculus described below, the angle α is returned.

3.2. Trigonometric solution

Once the delay τ between the two signals is obtained, the angle can be found with the help of trigonometric calculations. Considering a point M with coordinates x and y, which represent the position of the source. This two coordinates are assumed variable and unknown. Let's also consider two points, A and B with respective coordinates (x_A, y_A) and (x_B, y_B) corresponding to the positions of the microphones. The distance between them is fixed to d cm. The point of origin (origo) is defined as the middle point between A and B.

The target is to get the angle α which will give the direction of speaker location. A signal coming from the speaker reaches the point B at time t. In that moment, another point of the same wavefront is in the direction between M and A. This point is B' and as it belongs to the

wavefront, the distances BM and B'M are equal. Hence AB' is the distance traveled by the signal during the delay τ. The following figure illustrates the physical setup.

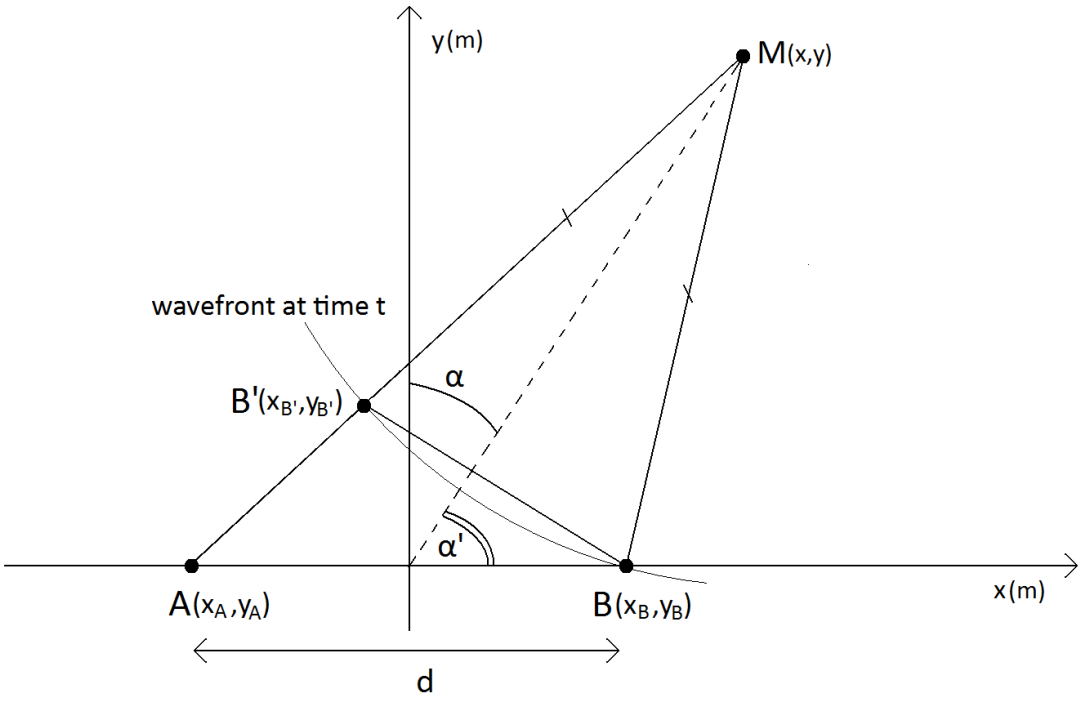


Figure 3: Diagram showing a possible situation of microphones and source.

Considering the suppositions exposed above, the following equations can be derived:

$$AB' = AM - B'M \tag{3.1}$$

Since

$$B'M = BM \tag{3.2}$$

The equation (3.1) becomes

$$AB' = AM - BM \tag{3.3}$$

with

$$AM = \sqrt{(x_A - x)^2 + (y_A - y)^2}$$
$$BM = \sqrt{(x_B - x)^2 + (y_B - y)^2} \quad (3.4)$$

In order to remove the square roots, the equation (3.3) is squared

$$AB'^2 = (AM - BM)^2 = AM^2 + BM^2 - 2 \cdot AM \cdot BM \quad (3.5)$$

Since the two microphones have fixed positions the following statements applies:

$$x_A = -x_B$$
$$y_A = y_B = 0 \quad (3.6)$$

This simplifies the equation (3.5) and after several calculations and term reordering, leads to

$$y = \pm\sqrt{\frac{AB'^2}{4} - x_B^2 + x^2\left(\frac{4 \cdot x_B^2}{AB'^2} - 1\right)} \quad (3.7)$$

In this expression the only variables are y and x. The value x_B is always constant since it represents the position of the microphones, which can be seen as reference points. Moreover even if the direction can vary, the length of AB' remains unchanged. So the equation represents all the possible positions of M, given a certain delay. Considering that the signal travels at the speed of sound c, the distance AB' is:

$$AB' = \tau \cdot c \quad (3.8)$$

To exemplify the previous formula, let's consider a distance d equals to 10 cm and a delay τ equals to 188.2 μs. This makes AB' be 6.4 cm.

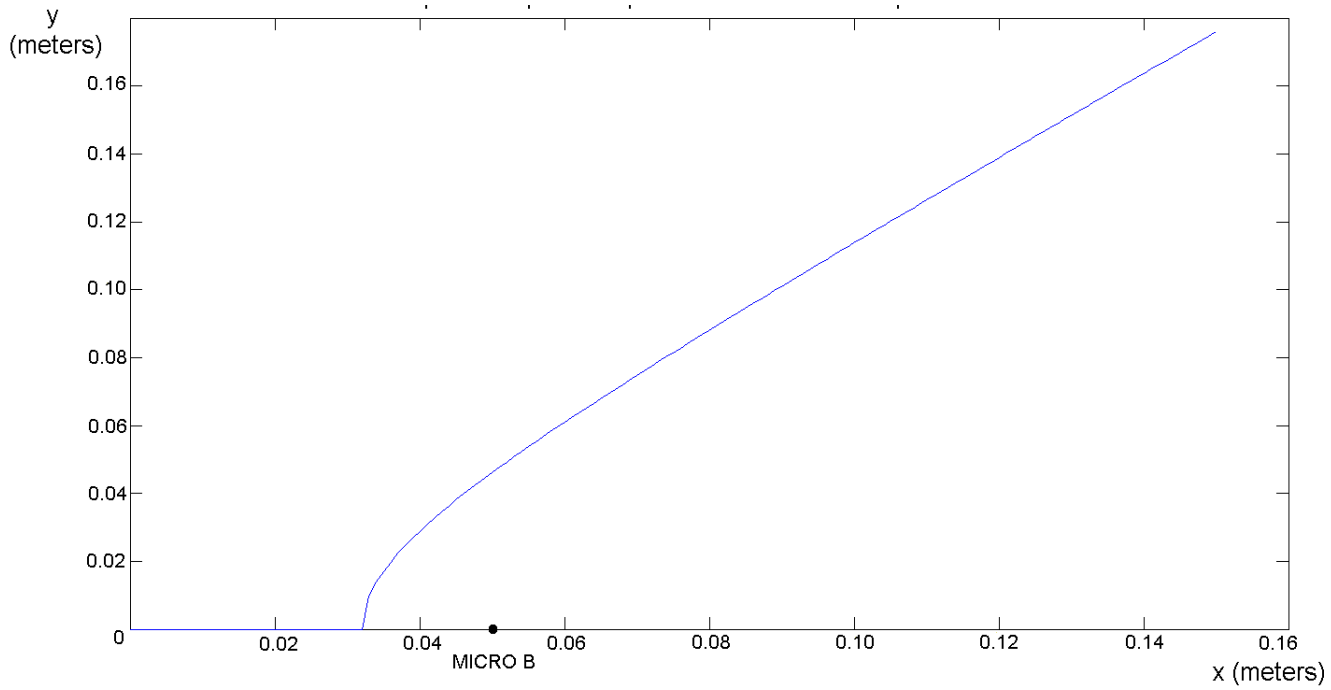


Figure 4: Speaker's possible positions for t = 188.2 µs.

For a certain delay, the function is not defined for all the values of x. Actually, since

$$AB' \leq 2 \cdot x_B \tag{3.9}$$

Then

$$\frac{AB'^2}{4} - x_B^2 \leq 0 \tag{3.10}$$

And so

$$x \geq \sqrt{-\frac{AB'^2(AB'^2 - 4x_B^2)}{4(4x_B^2 - AB'^2)}} \tag{3.11}$$

In the example x must be larger than 3.2 cm. The function has a hyperbolic evolution between this value and 5 cm and then it becomes linear. Only taking the linear part, first the slope must be obtained and then its arctangent.

$$\alpha' = \tan^{-1}\left(\frac{dy(x)}{dx}\right) \tag{3.12}$$

From this we may get the angle α', which is the one formed by the x-axis and the line y(x) (Figure 3). Since α is the angle formed by the y-axis and y(x), we do the following:

If $\alpha' \geq 0$ then

$$\alpha = 90 - \alpha' \tag{3.13}$$

And if $\alpha' < 0$ then

$$\alpha = -90 - \alpha' \tag{3.14}$$

Figures 5 and 6 show graphically how α' and α are obtained. It may seem confused to see negative values for the time delay τ. When this occurs, it means that the signal arrived first to microphone A than to microphone B (the speaker stands in the left part of the semicircle).

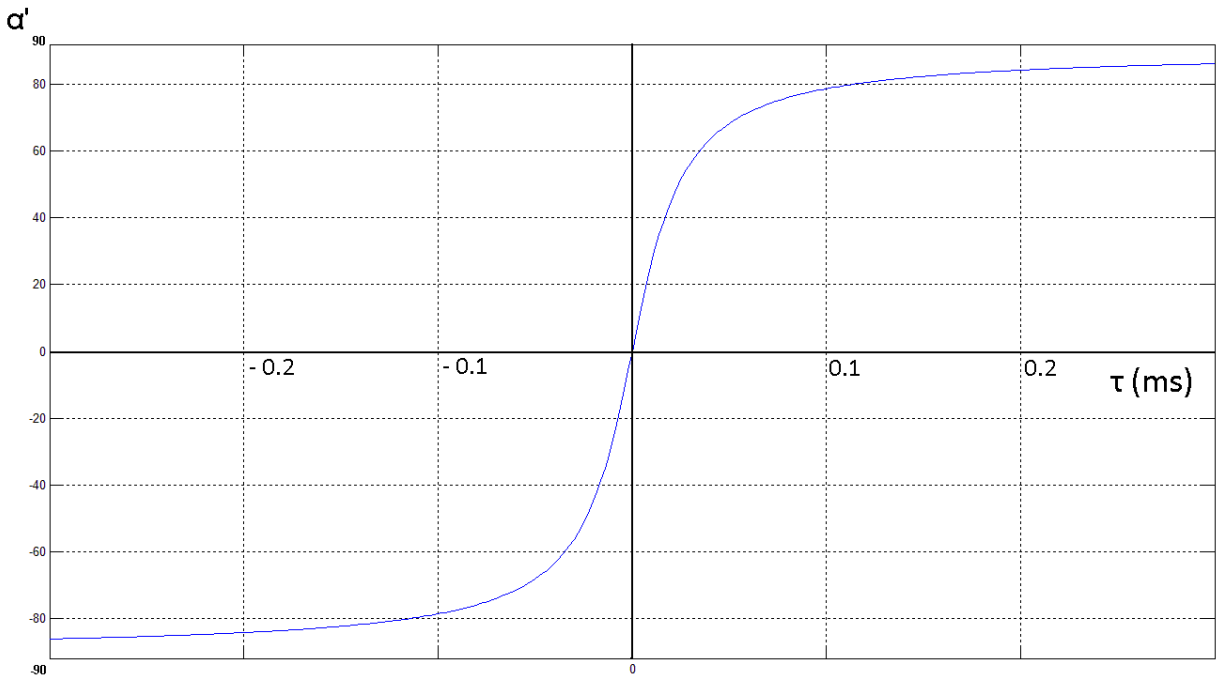


Figure 5: Obtaining of α'.

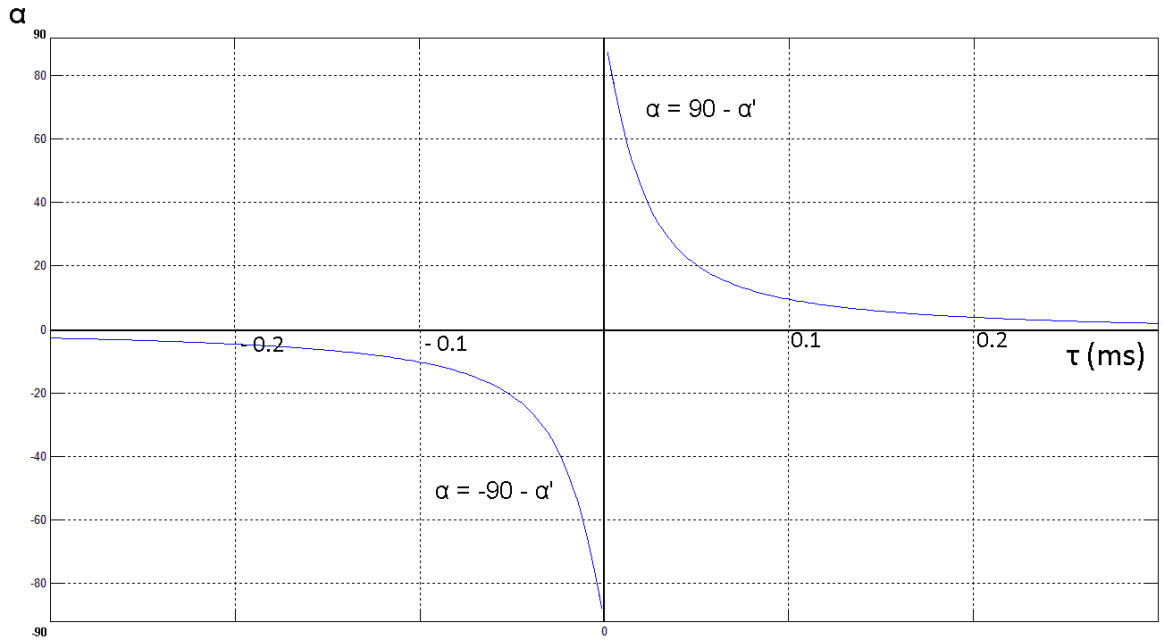


Figure 6: Obtaining of α.

Hence with this process it is possible to obtain the direction of arrival. As mentioned previously, this process only focuses on the linear part of the hyperbola. Thus the points belonging to the nonlinearity (Figure 4) must not be taken in account. Figure 4 shows that these points stand in an area with radius 5cm around the origin. So the system works suitably for speakers standing further than five centimeters from the microphones.

Once the physical problem has been described and a geometric solution has been proposed, the next step is introducing the microphones, which are in charge of transforming the audio signal into a digital signal.

3.3. Microphones

A microphone is an electro acoustic transducer receptor which translates acoustic signals into electrical signals. It is the basic element of sound recording. In a microphone two different parts can be considered: the Mechanical Acoustic Transducer (MAT) and the Electric Mechanical Transducer (EMT).

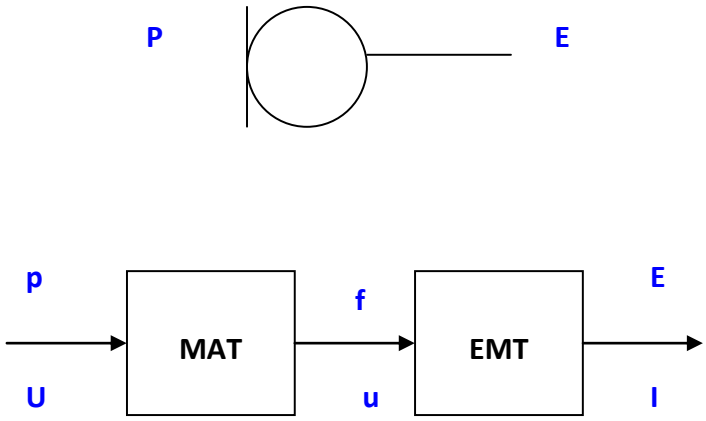


Figure 7: Acoustic-mechanical and mechanical-electrical transduction.

Where

P: pressure	f: force	E: voltage
U: volume of flow	u: diaphragm speed	I: current

The MAT turns pressure variations into vibrations of a mobile element, usually called diaphragm. The way in which this diaphragm faces the environment determines the frequency response.

The EMT converts these vibrations into voltage and electrical current. This is the engine of the microphone. Its operation is associated with a physical law which is related to mechanical and electrical variables.

3.3.1. Measurement scenarios

To determine the characteristics of a transducer, the acoustic settings in which the microphone is facing the pressure wave must be taken into account. The following measurement scenarios were studied [28].

- **Free field behaviour**

  Here the microphone is situated in an ideal anechoic chamber which has conditions of free field since it has no echoes or reverberation. This means that a point source, situated in some place of the chamber, must fulfil the condition of decreasing its sound level 6dB each time the measurement distance doubles. The changes produced by the microphone are not considered.

The chamber used in this project is not anechoic. In spite of having some furniture and low noise, the echoes and reflections they can produce are not significant enough to alter the free-field assumptions. So the behaviour of the microphones selected is approximately as free in field.

- **Near field behaviour**

  The microphone is situated near the source, not further than 50 cm. In this case the wavelength has the same order of magnitude as the size of the source. In the ideal situation the measurements are done exciting the microphones with an artificial mouth. The goal of these measurements is to increase the spherical character of the acoustic field, since some microphones present different response when the spherical divergence increases.

- **Far field behaviour**

  The microphone is located far from the source (further than 50 cm) at a distance from which its size does not affect the results of the measurements.

The scenario measurement used in this project corresponds to the far field since the distance used is around 1 m and the size of the source becomes negligibly. It is noteworthy that the assumed source is always a human mouth.

3.3.2. Characteristics

To be able to select a microphone in order to use it in a particular situation, it is necessary to know the main characteristics of each type.

- **Sensitivity**

  This is the ratio between acoustic pressure in the input of the microphone and the voltage provided by the electrical terminals in open circuit. The sensitivity is measured in the microphone axis and under the free field conditions previously mentionned. If the microphone is too sensitive, the received sound needs to be attenuated before being processed.

- **Distortion**

  This appears when the wave form in the output of the transducer is deformed compared to the wave form in the input. This situation can be produced by external or internal reasons. The most common one is saturation, which is produced when the amplitude of the input wave form is too high forcing the microphone to decrease it in the output.

- **Frequency response**

  This is the variation in sensitivity as a function of the frequency. In vocal applications the measures achieved to obtain the frequency response are carried out in free field environment. In near field, the frequency response shows the ability of some microphones to reinforce the low frequencies (proximity effect). The directivity of the microphone affects on the shape of the frequency response. In most of the cases the goal is to get a flat frequency response.

- **Directivity**

This characteristic is the faculty in which the microphone delivers in the output a different voltage according to the angle of incidence of the input wave. The three most common microphone patterns are the following:

- Omnidirectional

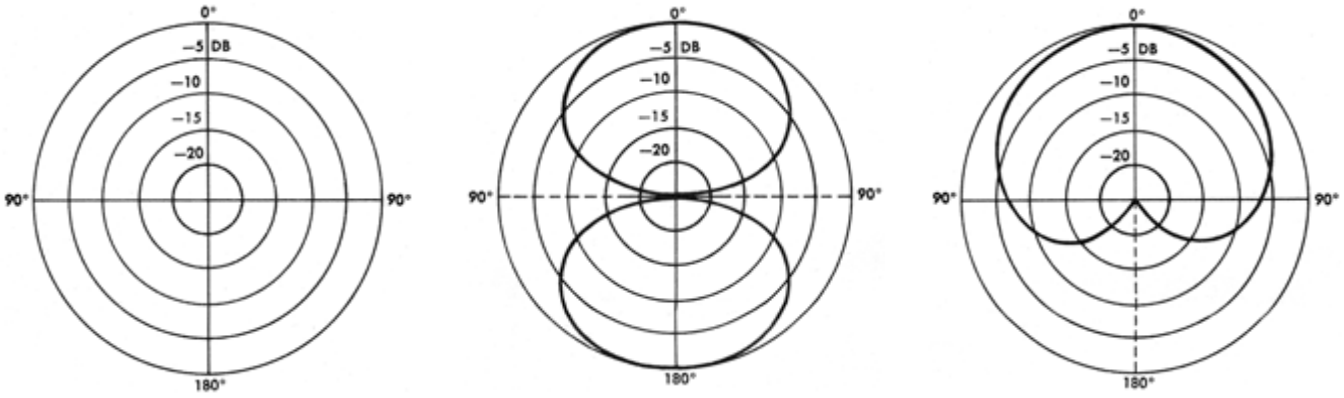
  The microphone delivers the same electrical output independently of the angle of incidence in the input. Generally it has a flat frequency response.

- Bidirectional

  The microphone captures the sound which comes from the front and rear but does not respond to sound from the sides. The frequency response is as flat as an omnidirectional pattern.

- Cardioid

  These are unidirectional microphones which polar diagram has heart shape. Their sensitivity is higher for sounds coming from the front than from the rear; in that situation the sound is reduced. The frequency response is flatter in mid frequencies. For low frequencies this response is more disperse and for high frequencies the microphone becomes more directional.



(a) (b) (c)

Figure8: Microphone patterns: a) Onmidirectional, b) Bi-directional, c) Cardioid.

After studying all these microphones the pattern selected is the omnidirectional one. With this model the direction of the speaker does not matter since it captures the same level of signal in all directions (considering the same frequency). So the microphone can be oriented in any direction.

3.3.3. Microphones used

This thesis deals with sound localization using two microphones, which is the minimum number of sensors needed to extract a time delay estimation and consequently calculate the direction of arrival. The chosen microphones are the AKG C417 Lavalier. It is one of the smallest Lavalier microphones available today. Its broadband and flat audio reproduction in an omnidirectional form (Figure 9) is ideal for several types of broadcast and theatrical applications. These microphones have a reinforcement of 5-6 dB around 8 kHz which aim is to compensate the signal loss due to the increase of the mouth's directivity in that frequency area [29].

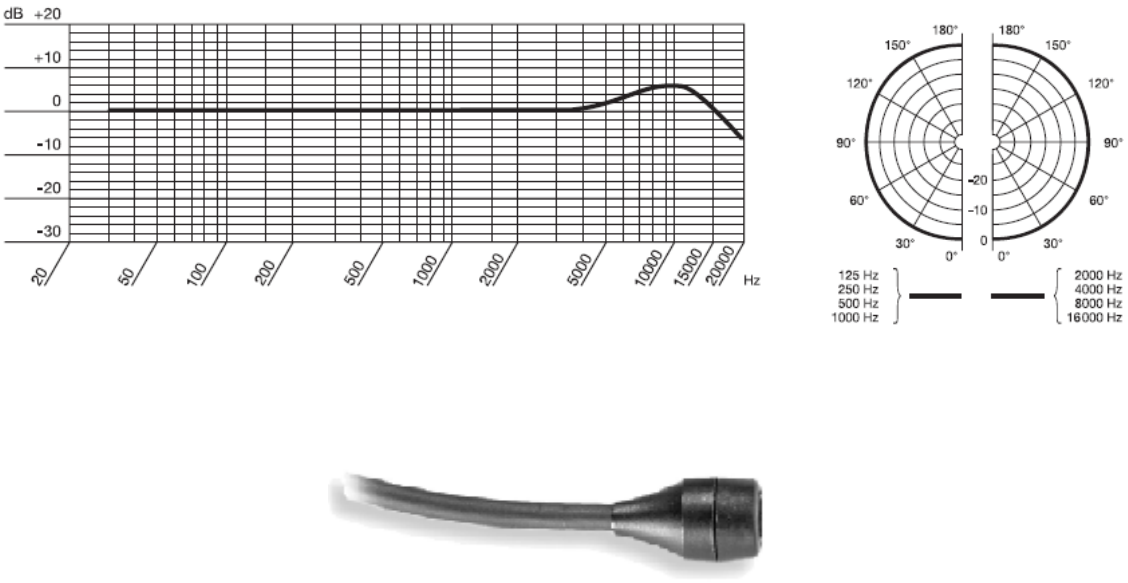


Figure 9: On-axis Frequency response (measured at 1 meter) and polar response [30].

4. Analog to Digital Conversion (ADC)

In the real world the most of signals sensed and processed by humans are analog and they must to be changed into numeric sequence with finite precision for being digitally processed. This change is made by Analog to Digital Converters (ADC). Therefore, a typical ADC has an analog input and digital output, which can be serial or parallel.

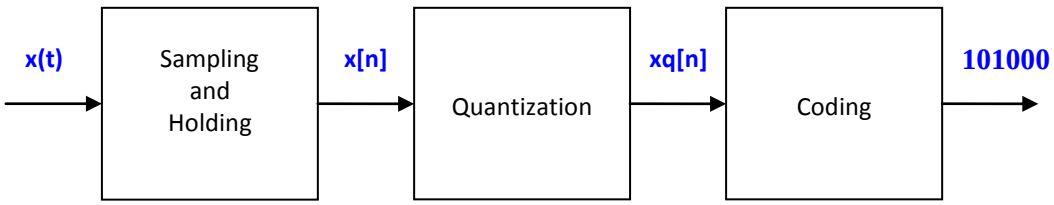


Figure 10: ADC structure.

Mainly this process is made in three steps:

1. Sampling: It consists of taking tension or voltage samples in different points of the signal x(t) to obtain x[n]. In addition to the sampling mechanism, another system known as Holding is commonly used. This way it is possible to hold the analog value steady for a while, while the following system performs other operations.

2. Quantization: The voltage level of each sample is measured. The possible values for these levels are infinite. The quantization system turns these levels into values coming from a finite range. The result is x_q [n].

3. Coding: Here the quantized values are converted into binary stream using codes previously established.

The frequency used during the sampling is known as sample rate or sampling frequency. For audio recording, the greater the number of samples is, the better audio quality and fidelity are obtained. The frequencies most used in digital audio are 24 kHz, 30 kHz, 44.1 kHz (CD quality) and 48 kHz.

The explanation below focuses only on the sampling process. During this process some phenomenons can appear. Two of them, which imply important restrictions, will be presented in sections 4.2 and 4.3.

4.1. Sampling

The sampling process converts signals from continuous-time domain to discrete-time domain. The main parameter is the sampling frequency fs. With reference to the physical problem described in section 3.1., the voice was expressed as an analog signal s(t). The time difference between the signals received by the two microphones was denoted τ. Looking at Figure 3, and considering a movement at speed of sound, the delay τ would represent the time elapsed to cross the distance B'A. Thus, denoting $s_1(t)$ the signal captured by MIC B and $s_2(t)$ the signal captured by MIC A, the following relations can be expressed.

$$s_1(t) = s_2(t - \tau) \tag{4.1}$$

$$\tau = \frac{B'A}{c} \tag{4.2}$$

Once the signals have been sampled by the microphones, the time is measured in samples. In this case, $s_1(t)$ and $s_2(t)$ become respectively $s_1[n]$ and $s_2[n]$. The variable t is changed into the variable n by the following relation to the sampling frequency fs.

$$n = t \cdot fs \tag{4.3}$$

The same way, the delay in time τ is turned into a certain number of samples N. Although discrete domain only accepts integer numbers, the delay N can also be fractional.

$$N = \tau \cdot fs \tag{4.4}$$

The discrete-time relation between the signals is given by

$$s_2[n] = s_1[n - N] \tag{4.5}$$

Figure 11 represents the same setup as in Figure 1 but including the sampled signals.

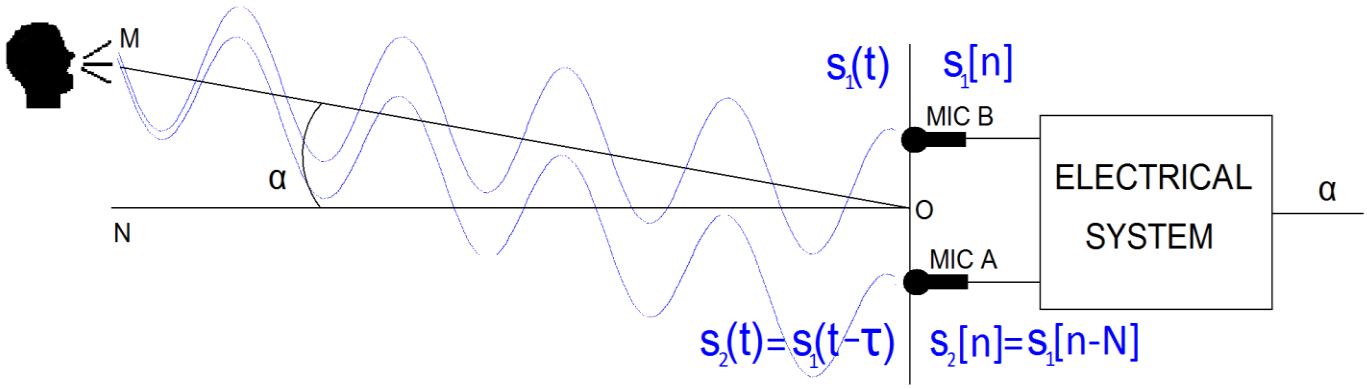


Figure 11: Representation after sampling.

Based on the equation (4.5), it derives that

$$s_2[n] = s_1[n] * \delta[n - N] \tag{4.6}$$

According to this equation, it can be affirmed that the propagation through the air which transforms $s_1[n]$ in $s_2[n]$ is equivalent to filter the signal $s_1[n]$ with a Dirac Delta centered in the position N. The importance of this fact will be explained in a further section.

Even if the value of N depends on τ and fs, which are supposed to be always positive, N can be negative. Actually its sign depends on the position of the source. It was decided that when the source stands closer to MIC B, the delay is positive and vice versa when it stand closer to MIC A. Hence, if the speaker stands in the left half of the front semicircle the relation between the signals is

$$s_2[n] = s_1[n + |N|] \tag{4.7}$$

In order to highlight the fact that N is negative, the operator minus has been turned into plus and N has been represented in absolute value. Otherwise equation (4.7) do not defer from equation (4.5). Figure 12 shows three different situations: the first with a speaker standing in the direction -90°, another standing in 0° and the last one in +90°. The delay in the extreme positions is called N_{max} .

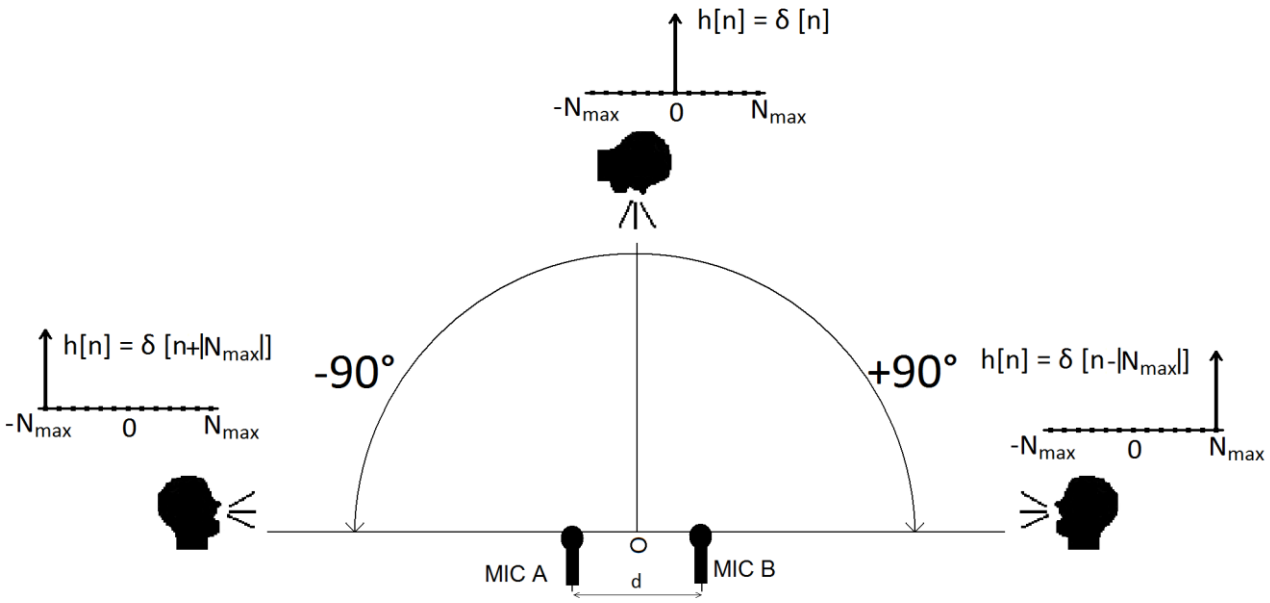


Figure 12: Three scenarios considered: the extreme positions and the middle one.

Considering that the distance between microphones is d, the maximum number of samples that can exist between the two signals N_{max} can be expressed as follows.

$$N_{max} = \frac{d}{c} * fs \tag{4.8}$$

4.2. Temporal Aliasing

The Nyquist sampling Theorem says that a band limited signal can be reconstructed from a number of equally spaced samples if the sampling frequency is equal or higher than twice the maximum signal frequency.

$$f_s = 2 \cdot f_{max} \tag{4.9}$$

The highest frequency that can be represented by a discrete signal with this f_s is the Nyquist frequency. This frequency is half the sampling frequency, which is $fs/2$.

If this sampling theorem is not fulfilled the phenomenon called temporal aliasing occurs. When this happens, the consequences are mainly changes in the shape of spectrum and loss of information.

The human voice range includes frequencies from 300 Hz to 4000Hz [31]. Since the maximum frequency is 4 kHz, the minimum sampling frequency is 8 kHz. However, the precision and accuracy increase when the maximum number of samples N_{max} is high (Equation (4.8)). So a higher sampling frequency was desired. Finally it was decided to use the CD quality sampling frequency, which is

$$\boldsymbol{f_s = 44.1\ kHz} \tag{4.10}$$

4.3. Spatial Aliasing

To explain the phenomenon called spatial aliasing, it is useful to consider the following figure:

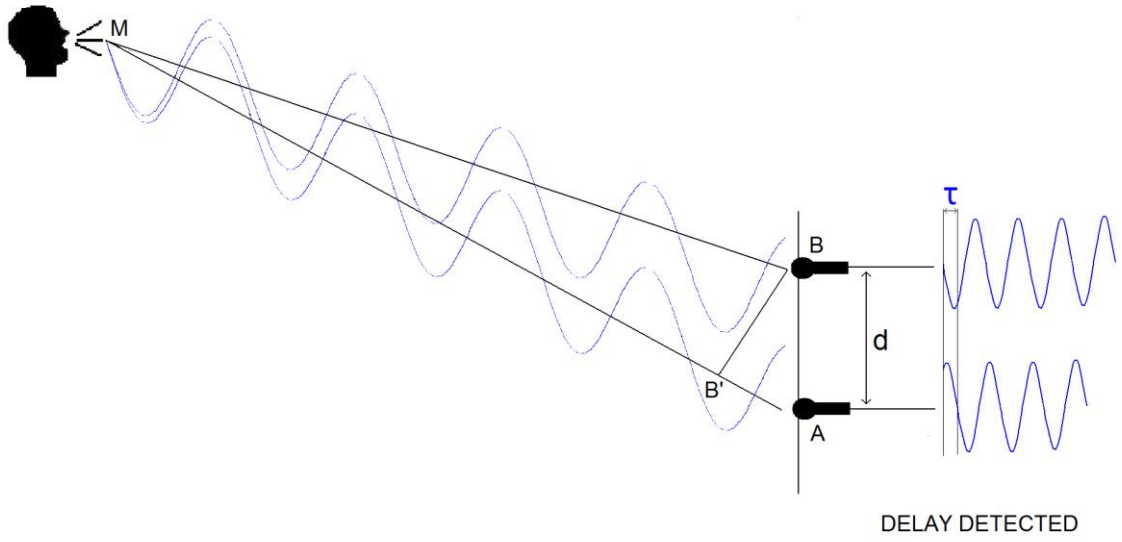


Figure 13: Normal situation where a delay between signals is detected.

In this situation the sound arrives simultaneously to B and B'. As explained in section 3, the time elapsed until the sound reaches A, is named τ. The situation is the one described in section 4.1. This parameter is measured in time, or in distance comparing it with the wave length λ. In Figure 13 this delay is less than λ/2.

The distance B'A depends on the distance between microphones, d. If this distance increases up to d', the distance B'A increases too. Looking at equation (4.2) it is clear that the delay has the same behaviour.

The problems appear when the microphones are too separated. In this case, the delay can increase until a higher value than λ/2. If that occurs, two situations may arise: the delay detected is false or the delay is not detected at all. Figure 14 and Figure 15 summarize both situations.

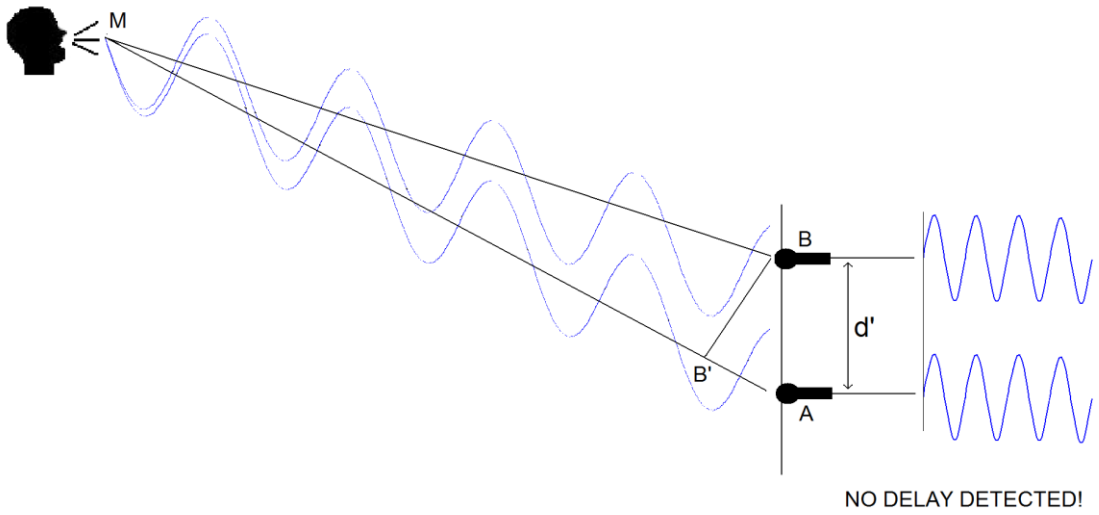


Figure 14: Scenario with a delay equal to λ.

The delay is the time it takes for the signal to traverse B'A. In this case, B'A is equal to a whole wavelength, so the signals captured by both microphones are equal and no delay is detected. This is obviously untrue, since the delay exists. In the other situation, the delay follows the equation (4.11). This can happen for a distance between microphones equal to d''.

$$\lambda/2 < B'A < \lambda \qquad (4.11)$$

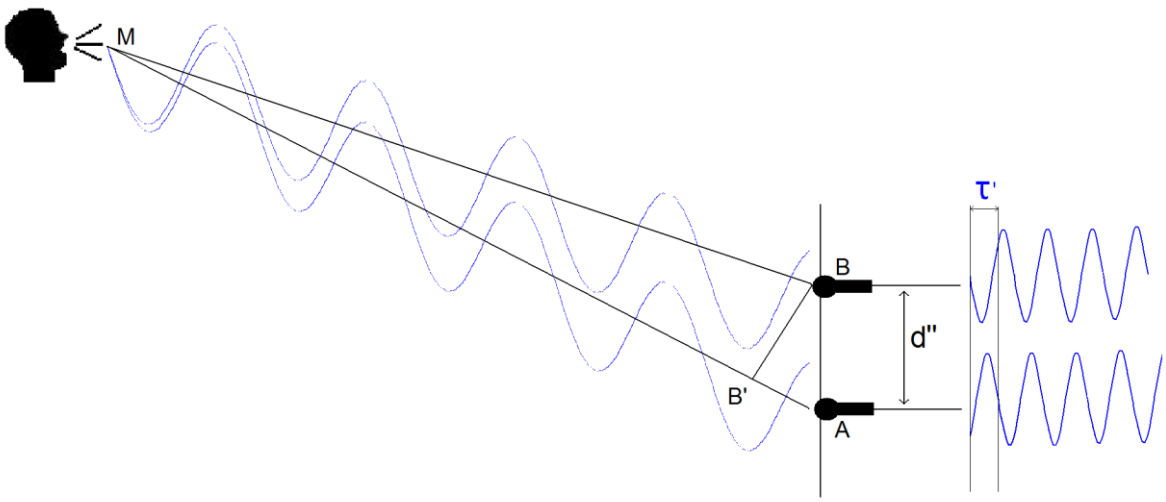


Figure 15: Scenario with a delay such that λ/2 < B'A < λ.

Taking in account the signals captured by the microphones, it can be proved that the delay detected won't be the real delay.

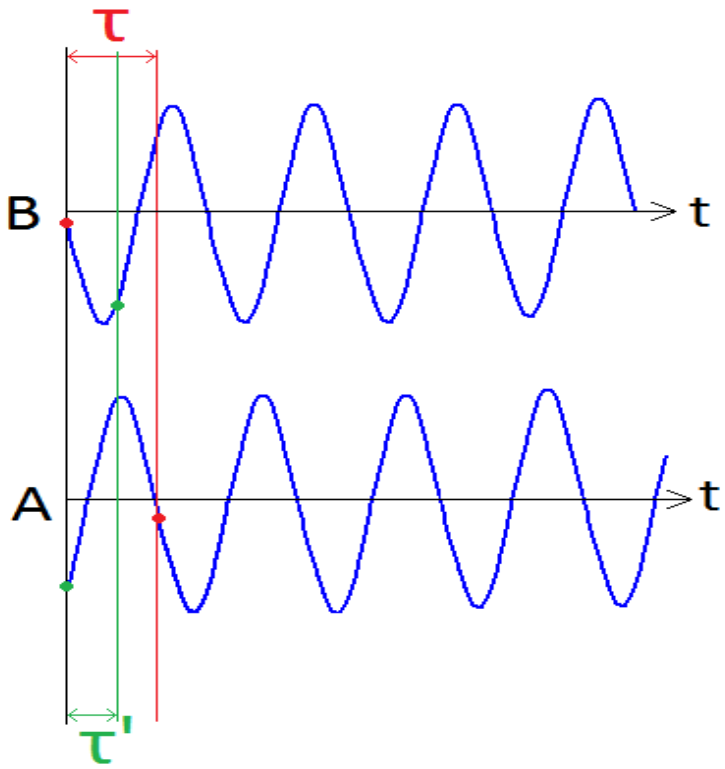


Figure 16: Two different delays can be detected τ and τ'.

According to Figure 15, the real delay τ is the one coloured in red on Figure 16. The speaker is closer to MIC B so the signal reaches MIC B before MIC A. However if these signals are inserted into a system which aim is to obtain the delay, the system would return the green coloured delay, τ'. Actually τ' is the only delay smaller than λ/2, thus it will be mistakenly identified as the existing delay. So when two signals like those presented on Figure 16 are captured, it would seem that the speaker is closer to MIC A than to MIC B. Hence the delay obtained is false. So the condition that must be fulfilled in order to avoid spatial aliasing is (4.12).

$$B'A \leq \frac{\lambda}{2} \tag{4.12}$$

Using the extreme case B'A = λ/2, is possible to observe (Figure 17) the different zones with and without aliasing.

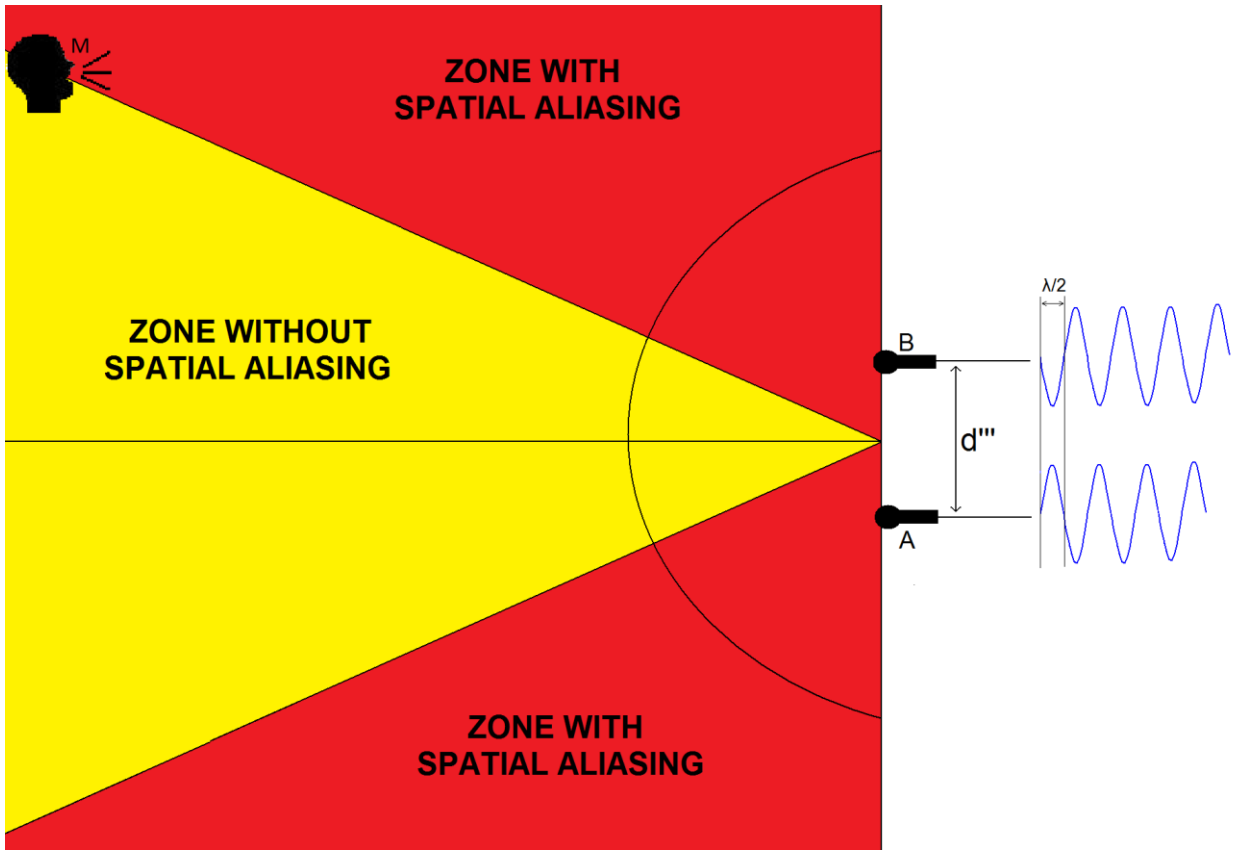


Figure 17: Zones with and without spatial aliasing for a certain distance between microphones.

But the aim of this project is to get a range within 90˚ and -90˚ so the main situation to take in account is when the speaker is aligned with the microphones (Figure 18). In this case the points B and B' are coincident and the distance B'A corresponds with the distance between microphones, d'''. Hence, to avoid the spatial aliasing in this range the distance between microphones must to be:

$$d \leq \frac{\lambda}{2} \tag{4.13}$$

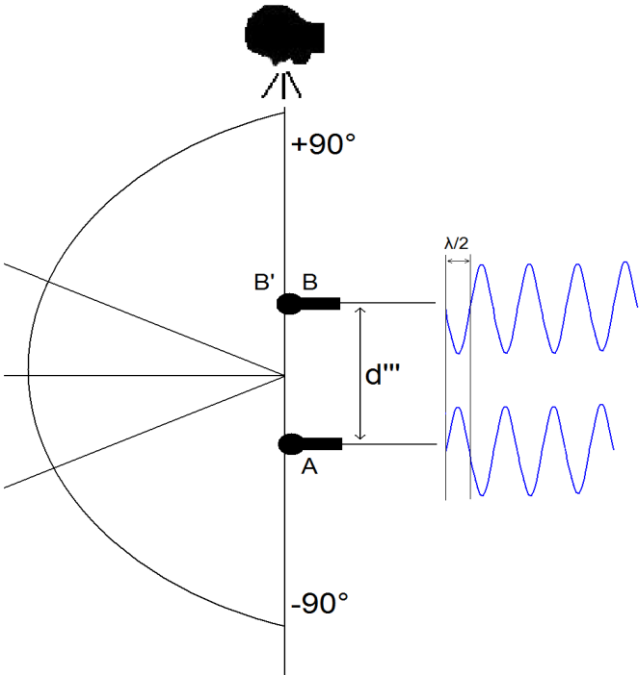


Figure 18: Desired scenario with no spatial aliasing in the range from -90˚ to +90˚.

To obtain the maximum distance between microphones, it is necessary to find the minimum value for λ. For that, the human voice range frequency must be taken in account. This range is comprised within 300 Hz and 4 kHz [24].

$$d \leq \frac{\lambda_{min}}{2} \quad ; \quad \lambda_{min} = \frac{c}{f_{max}} \quad ; \quad d \leq \frac{c}{2 \cdot f_{max}} \tag{4.14}$$

Considering $f_{max} = 4000$ Hz and $c = 343\ m/s$:

$$d \leq \frac{343}{2 \cdot 4000} = 4.3\ cm \tag{4.15}$$

But from a practical point of view this value causes difficulties. Actually the accuracy when placing the microphones was not assured: the smaller is the distance d and the higher would be the impact of a possible error of placement. Furthermore a higher value of d, leads to a higher number of delay samples N_{max} (4.8) which leads to a higher precision. For this reason and after several tests the distance selected was **d=10 cm.**

Since the chosen distance is higher than the maximum distance (4.3 cm), it seems that there will be spatial aliasing in a portion of the semicircle. Nevertheless the condition (4.13) can be fulfilled for a higher value of λ. The maximum frequency enabling a semicircle with no spatial aliasing is

$$f_{max} = \frac{c}{2 \cdot d} = \frac{343}{2 \cdot 0.1} = 1715\ Hz \tag{4.16}$$

Hence, even if the maximum voice frequency is 4 kHz, the captured signals would only be studied for frequencies lower than 1715Hz. Figure 19 shows the range of angles without spatial aliasing according to the distance between microphones and varying the maximum frequency.

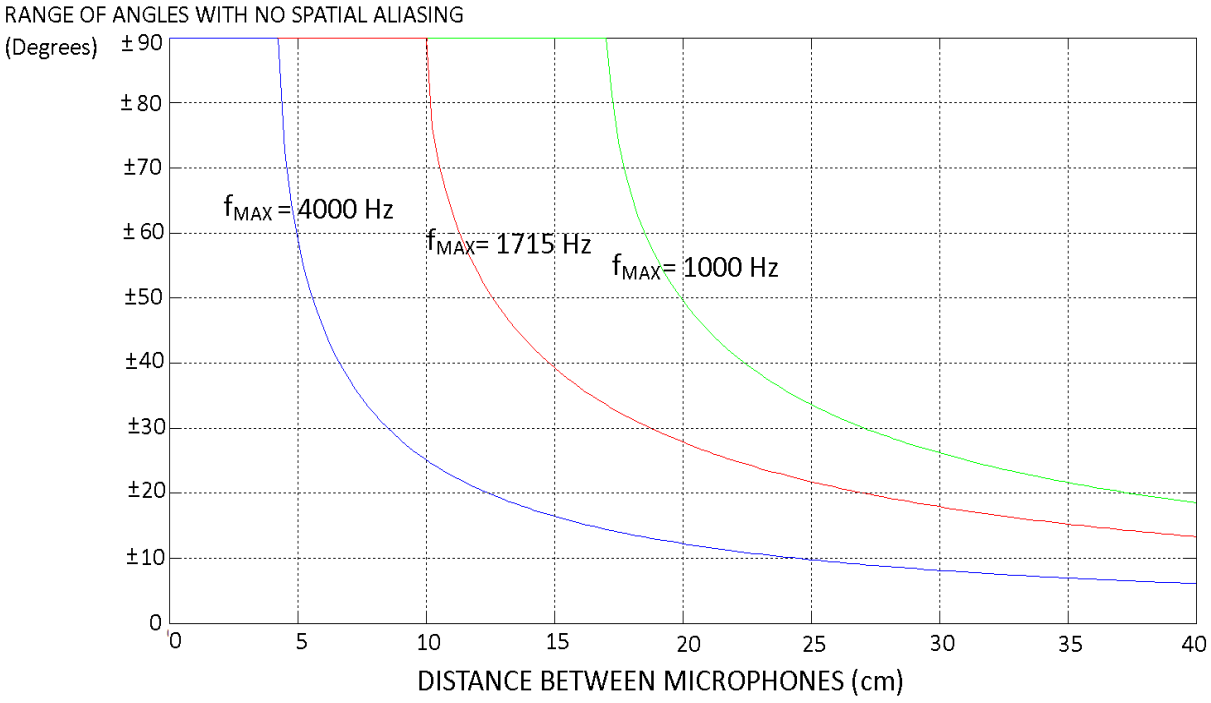


Figure 19: Range of directions without spatial aliasing according to the distance between microphones.

Now that the physical issue described in Figure 1 has been explained, as well as the microphones and the sampling process, the time has come to detail the electrical system. Its target is to obtain the direction of the source α.

5. Electrical system

First of all, an overview of the whole system will be presented. Then a detailed explanation of every step will be given in order to understand it better. Figure 20 summarizes the process.

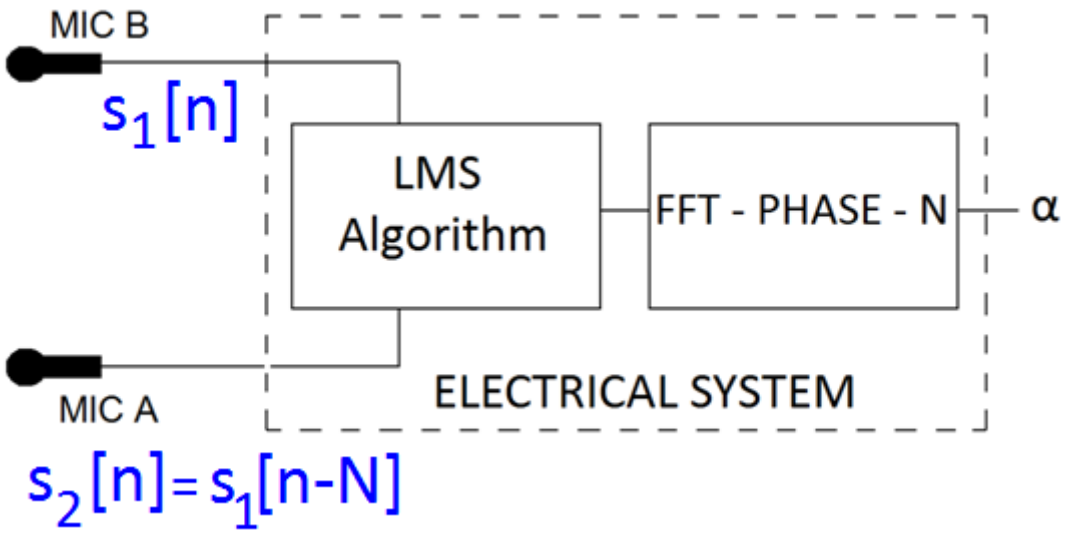


Figure 20: Diagram of the electrical system.

As explained in the previous sections, the main target is to obtain the direction where a speaker is emitting sounds. To achieve that, the system shown in Figure 20 was designed in MatLab code. In section 4.1 it was mentioned that two delayed signals were captured by the microphones. MIC B records the signal $s_1[n]$ and MIC A records $s_2[n]$. According to equation (4.5) the two signals are delayed a number of samples N. Getting this delay is crucial in order to obtain α. Actually, as explained in section 3.2, with the samples delay N the value of B'A can be obtained and after several trigonometric calculations, the direction α can be returned.

To obtain α, the signals are first processed by an algorithm (Least Mean Square) which will provide N. This chapter will be structured in two sections: in section 5.1, the Least Mean Square algorithm will be explained and then in section 5.2 the next steps leading to obtain the direction.

5.1. Least-Mean Square Algorithm (LMS)

5.1.1. General approach

The Least-Mean-Square algorithm (LMS) was invented by Stanford's professor Bernard Widrow and PHD student Ted Hoff In 1959 [32]. It is used in adaptive filters to calculate the filter coefficients which allow getting the minimum expected value of the error signal. This error signal is defined as the difference between the desired signal and the output signal. LMS belongs to the family of stochastic gradient algorithms, i.e. the filter is adapted based on the error in the present moment only. It does not require correlation function calculation nor does it require matrix inversions, so it is relatively simple.

Consider two signals x[n] and d[n], and then consider the filter h[n] such that:

$$d[n] = x[n] * w[n] \tag{5.1}$$

where * represents convolution operator. Applying LMS algorithm to x[n] and d[n] will theoretically give w[n] as an output. As shown in the picture below, LMS algorithm has two inputs, x[n] and d[n], and three outputs, y[n], e[n] and w[n].

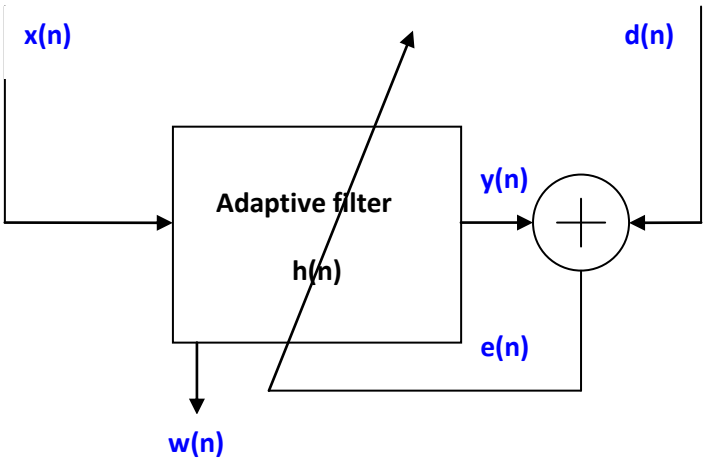


Figure 21: LMS algorithm diagram.

The algorithm has two main parameters, M and μ. M is the order of the filter and μ, called the step-size, controls the convergence of the algorithm. The coefficients of the filter h[n] are called weights and will be represented as a vector w[n]. LMS consists of two main processes, the filtering and the adaptive process.

- Filtering process:

First w[n] must be initiate with an arbitrary value w[0]. At each instant k, the input x[n] we are working with is:

$$x[n] = [x[k], x[k-1], x[k-2], \dots, x[k-M+1]] \quad (5.2)$$

which are the last M samples of the signal x[n]. Then y[n] is obtained by filtering x[n] with h[n]:

$$y[n] = w[n]^H * x[n] \quad (5.3)$$

where $w[n]^H$ is the conjugate transpose of w[n]. After that, the estimated error function e[n] is calculated as:

$$e[n] = d[n] - y[n] \quad (5.4)$$

- Adaptive process

The new weights are then calculated, based on the previous w[n] and the error function e[n]:

$$w[n+1] = w[n] + \mu \cdot x[n] * e^T[n] \quad (5.5)$$

The successive corrections of the weight vector eventually leads to the minimum value of the mean square error.

As shown in the equation, the parameter μ is of crucial importance. It controls the convergence of the algorithm, which is essential in a real-time system. Considering R[n] the correlation matrix of x[n] which is obtained as:

$$R[n] = x[n] * x[n]^H \tag{5.6}$$

and let λ_{max} be the largest eigenvalue of the matrix R. It is proved [24] that the LMS algorithm is seen to converge and stay stable if the step-size satisfies the following condition:

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{5.7}$$

5.1.2. Application to the system

In section 4.1 it was explained that the delay existing between the two captured signals $s_1[n]$ and $s_2[n]$ can be expressed on discrete domain as a value N. Besides, according to the expression (4.6) the function allowing the transformation from $s_1[n]$ to $s_2[n]$ can be represented as a Dirac Delta centered in N. Taking a look at the LMS algorithm, and comparing equations (4.6) and (5.1), it is easy to presume that if

$$\begin{aligned} x[n] &= s_2[n] \\ d[n] &= s_1[n] \end{aligned} \tag{5.8}$$

then

$$h[n] = \delta[n - N] \tag{5.9}$$

With this information, the process which is explained in section 5.2 will help to obtain the value of N and thus the direction α. Hence the LMS algorithm plays a major role in the whole process. For that reason, it is important that the design is performed with the highest possible precision. A right choice of the parameters, M and μ, is essential to obtain a system with good performance.

5.1.3. Choice of the parameters

The first parameter is the step-size of the algorithm, μ and it must fulfill the condition shown in (5.7). It is also proved [24] that an optimum value for the step-size would be:

$$\mu = \frac{1}{\lambda_{max} + \lambda_{min}} \tag{5.10}$$

λ_{max} and λ_{min} are the highest and lowest eigenvalues of the correlation matrix R[n] respectively and this matrix depends obviously on the signal x[n]. Since x[n] changes on every loop iteration, a new matrix R[n] and new values for λ_{max} and λ_{min} should be calculated.

Since this take a high amount of time it was decided to choose a constant value for μ. Several tests were held for over one hundred signals calculating values for λ that returns acceptable results. Then an average of all these values was obtained, resulting that a good constant value could be

$$\boldsymbol{\mu = 0.0117.} \tag{5.11}$$

The second parameter, M, is the order of the filter and it is basically the length (measured in number of samples) that h[n] must have. It depends on other two important constant values: the distance between microphones *d* and the sampling frequency *fs*. As previously mentioned, the LMS' main output should be an N-delayed Dirac delta. Each one of these possible delays represents a direction were the speaker can be. So the minimum length of the filter must correspond to the maximum number of samples delay N_{max}. With the data calculated in sections 4.2 and 4.3, and with the formula (4.8), N_{max} can be calculated.

$$N_{max} = \frac{d}{c} \cdot fs = \frac{0.1}{343} \cdot 44100 = 12{,}97 \approx 13 \tag{5.12}$$

So the maximum number of samples is 13, since in discrete domain the delays must be integers. Hence the function h[n] must include samples from N=-13 to N=13 which means at least 27 samples. Unfortunately, working with real signals makes results be not as ideal as expected. As shown in the Simulations chapter, the functions obtained are not pure Dirac deltas, but sinc-shaped functions. That means a big part of the information lies before – N_{max}

and after N_{max}. So security margins were added in both sides of the filter to assure that this information is not lost. After several tests it was decided that the filter should have a length of

M=50 samples. (5.13)

So far, the parameters were proper from the LMS algorithm. The next one, on the contrary, is due to the use of MatLab in the process. Theoretically, the filter h[n] could randomly be like the one in Figure 22. In that example N is positive, which means the voice gets to MIC B before than to MIC A (so the speaker is closer to that microphone). According to LMS diagram on Figure 21 the situation is

$$d[n] = x[n - N] \quad (5.14)$$

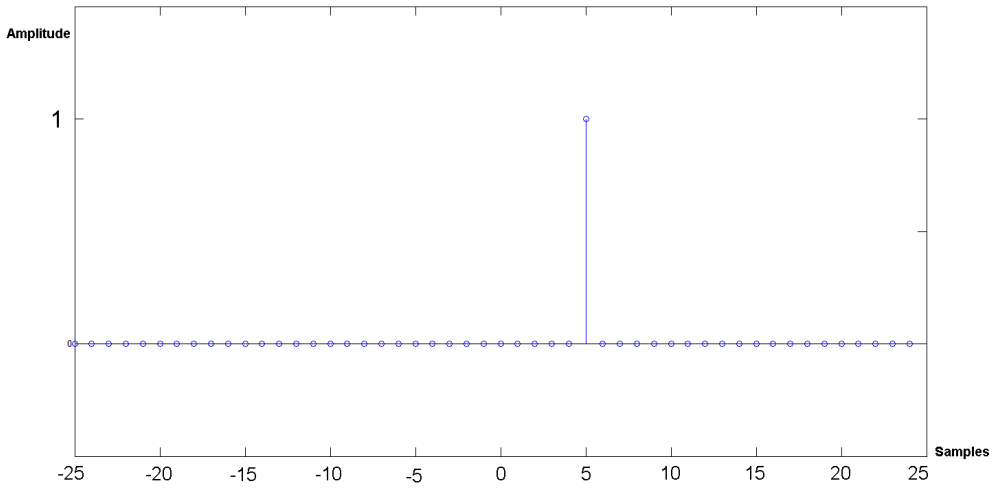


Figure 22: Theoretical filter h[n] for N=5.

But it can also happen that the speaker is closer to microphone A. As explained in section 4.1, N would be negative, which apparently is not a problem. The operator minus from the previous expression (5.14) turns into plus and the situation is

$$d[n] = x[n + |N|]$$

Or (5.15)

$$x[n] = d[n - |N|]$$

Nevertheless if N is negative, the code should be able to index negative values in an array, which is in fact a problem. Actually MatLab cannot index negative integers (i.e. h[-5]). The solution for that is to delay the signal d[n] in a certain number of samples before running the LMS algorithm. This way the system always acts like if the speaker's voice reached MIC B before MIC A and the situation is always the one described by (5.14). This extra delay, called DESP, is the third parameter of the system. Figure 23 sums the situation.

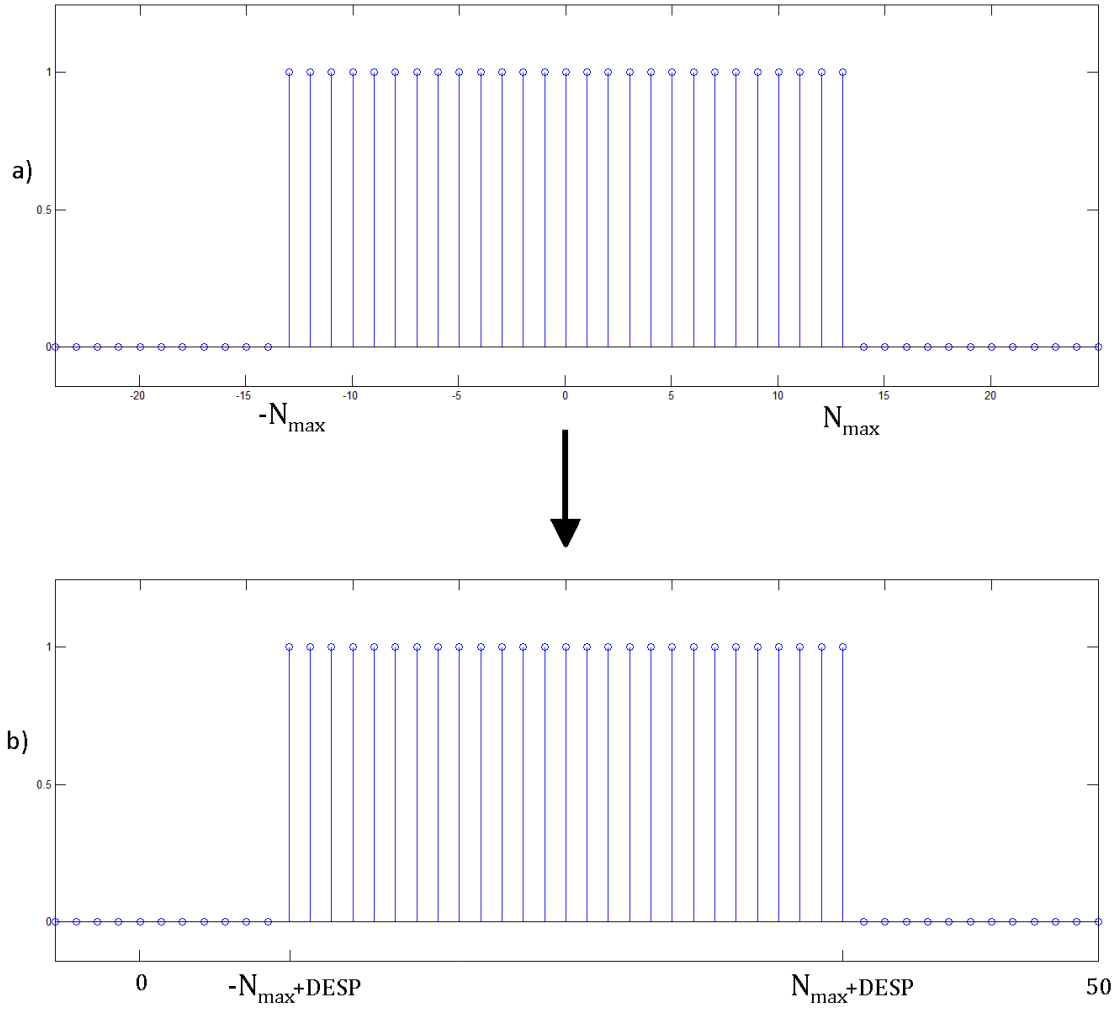


Figure 23: a) All possible integer delays from -13 to +13; b) Same situation after adding DESP.

This number depends mainly on the maximum number of samples in the negative part of the function. N_{max} was already calculated in (5.12) and it is equal to 13. So that is the minimum value for DESP. However, as explained previously, part of the information remains out of the borders, so it is necessary to increase the number. After running several tests, it was decided that the vale should be 20. All these values stay related one another, and always depend on the sampling frequency and the distance between microphones. A change in fs or d would automatically modify the parameters, so it will not affect the behavior of the system.

Once the parameters are calculated, the algorithm should work as desired. Thus at the end of the LMS algorithm block, it is necessary to process the filter function h[n] in order to obtain the delay N.

5.2. Delay calculation

The delay calculation block shown in Figure 22 is composed of three steps: First obtaining the Fast Fourier Transform, then its phase and finally the delay N. The choice of this three step method was due to its simplicity and good performance.

The FFT is an efficient algorithm used to calculate the Discrete Fourier Transform (DFT), so it will help to switch from time domain to frequency domain. Furthermore, it is a tool that MatLab can run very efficiently. Consider a discrete function $x[n] = \{x_0, x_1, \dots, x_{n-1}\}$, where x_k are complex numbers. Its DFT is defined as

$$f_j = \sum_{k=0}^{n-1} x_k \; e^{\frac{-2\pi i}{n} jk} \quad j = 0, \dots, n-1 \tag{5.16}$$

A direct evaluation of this formula requires O (n^2) arithmetic operations, whereas the FFT leads to the same results with only O (n log n) operations [33]. Considering an ideal situation, the filter would be a pure Dirac delta delayed N' samples.

$$h[n] = \delta[n - N'] \tag{5.17}$$

This means

$$h[n] = \{0, 0, \dots, 1, \dots, 0\} \tag{5.18}$$

So its n-point DFT would be

$$H_j = 1 \cdot e^{\frac{-2\pi i}{n} jN'} \tag{5.19}$$

This transform has two main components: modulus and phase. Since the desired information is the position of the delta (N'), only the phase will be useful (the modulus only has information about amplitude). So the next step consists on calculating the phase, which is really simple with MatLab commands. Calling Ω the phase

$$\Omega = \frac{-2\pi}{n} jN'. \tag{5.20}$$

The only variable is j, which is the index of the FFT. So the phase is linear and depends directly on the delay N'. With the derivate, the variable j disappears and the slope S is obtained.

$$S = \frac{-2\pi}{n} N' \tag{5.21}$$

So

$$N' = \frac{-S \cdot n}{2\pi} \tag{5.22}$$

At this point, the parameter DESP must be subtracted in order to get the right delay.

$$N = N' - DESP \tag{5.23}$$

Figure 24 shows graphically all the process incurred by two ideal delayed functions.

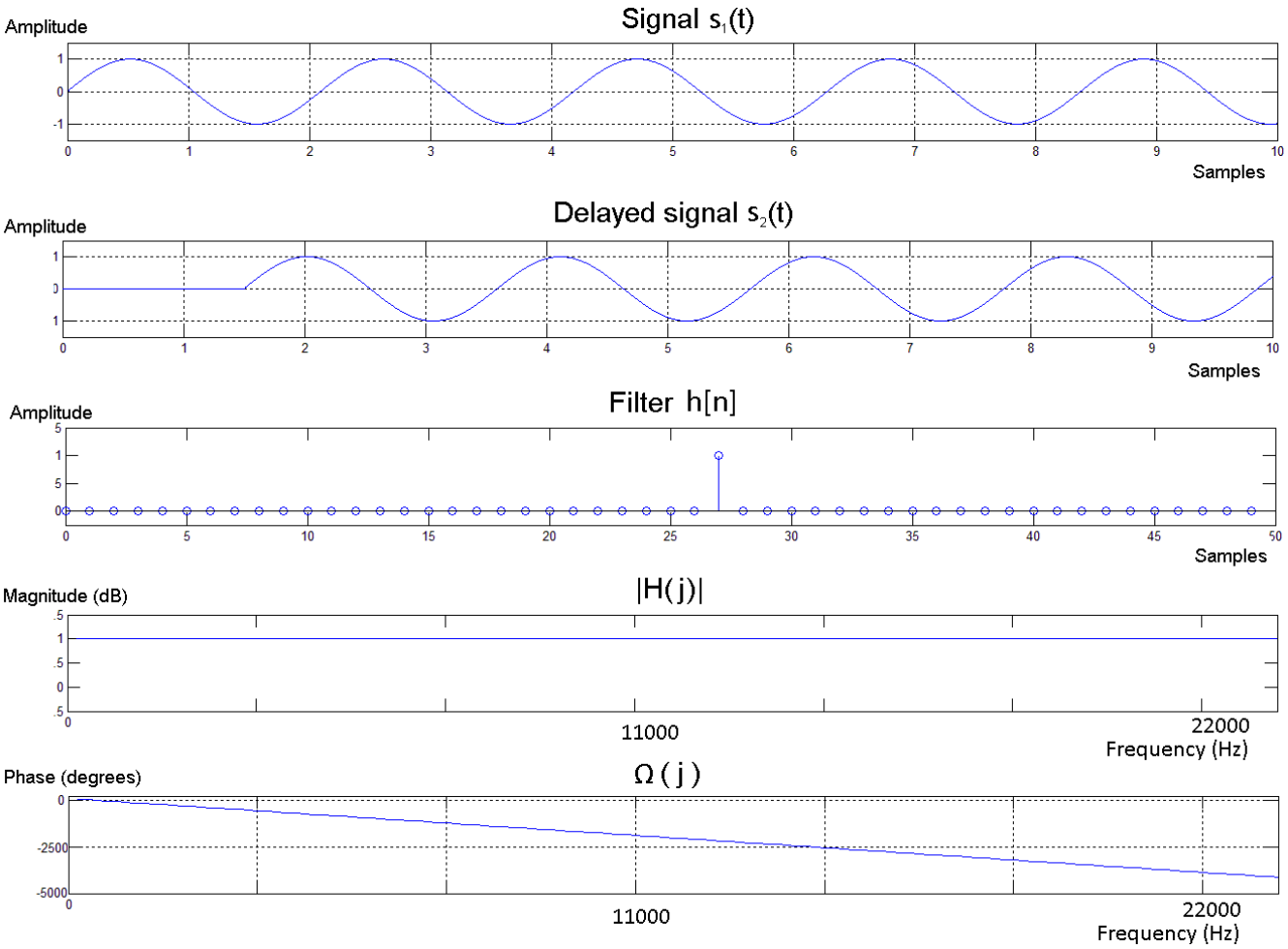


Figure 24: Process to obtain the phase with integer delay.

After calculating the value of N, expression (4.4) will help to get the delay in time τ

$$\tau = \frac{N}{fs} \qquad (5.24)$$

Then with (3.8) the distance B'A can be obtained and thus all trigonometric calculations presented in Section 3.2, which lead to the direction α, can be applied.

6. Simulations

The following simulations were carried out in order to test the system. Each group of tests was made to check that a specific part of the program or, in the case of the last tests, the whole program worked correctly. The results are shown and explained in the order they were completed.

The system was built in a two step methods than can be called partial real-time. Two groups of signals were tested: real and non-real signals. The first ones are stereo recorded signals, coming from human speakers in different positions. The others are firstly, White Gaussian Noise and then, mono recorded signals.

The first results presented are the ones corresponding to the White Gaussian Noise (WGN) generated by MatLab. Those signals were highly useful to check that the LMS-algorithm worked as expected. As explained previously, the algorithm needs two inputs, each one coming from one of the microphones. A WGN signal corresponds to the signal captured by one of the microphones $s_1[n]$. The signal captured by the other microphone $s_2[n]$, is generated by delaying N samples the signal $s_1[n]$. The simplest way to obtain this delay is to perform the following convolution

$$s_2[n] = s_1[n-N] = s_1[n] * \delta[n-N] \tag{6.1}$$

This way, two inputs $s_1[n]$ and $s_2[n]$ are generated. When these signals are introduced to the LMS, the output should be

$$h[n] = \delta[n-N] \tag{6.2}$$

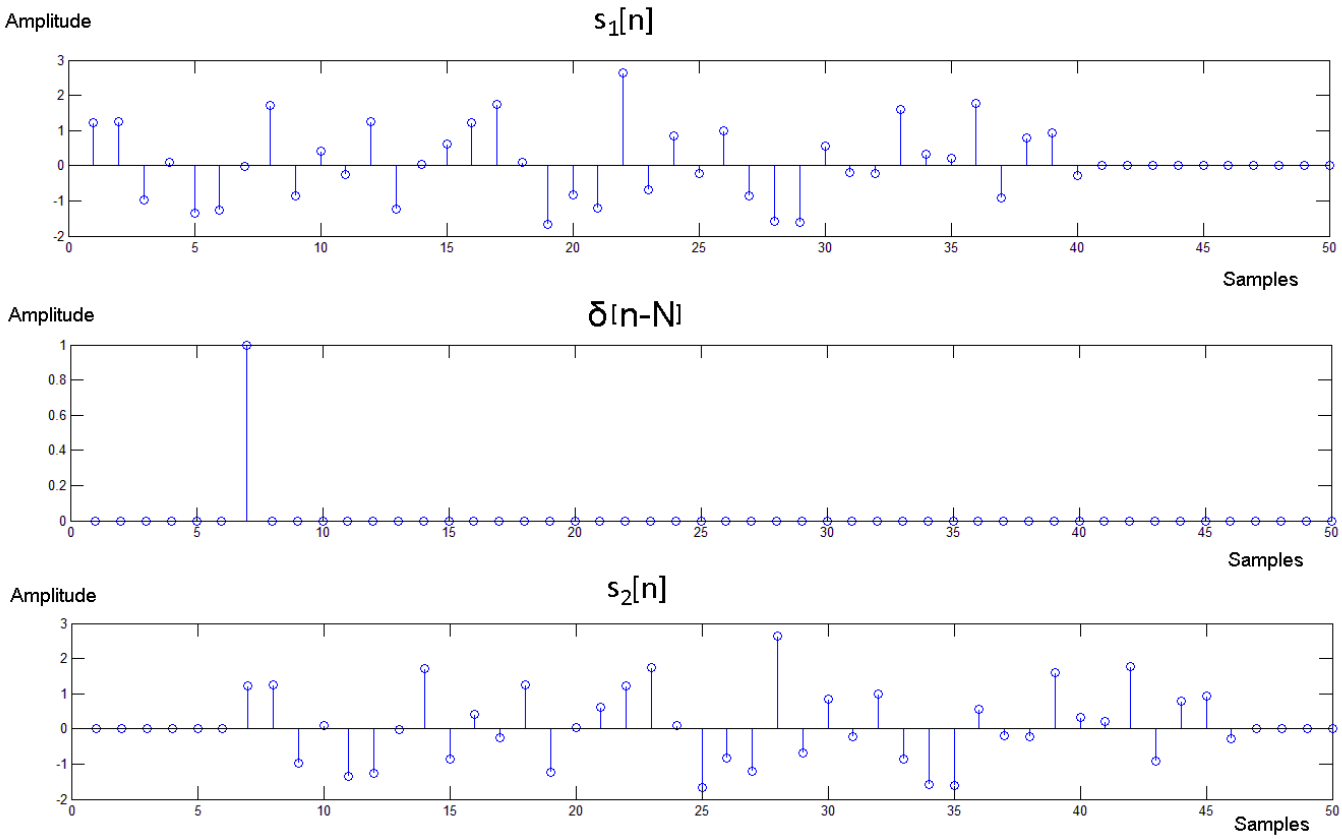


Figure 25: Input $s_2[n]$ generated by filtering $s_1[n]$ with function $h[n] = \delta[n - N]$.

6.1. Non-real signals

6.1.1. White Gaussian Noise

The maximum delay calculated in (5.12) is 13, so the signals were delayed from -13 to +13 samples. As explained before, MatLab cannot index negative values so the value DESP must be added. This makes the delay values oscillate from +7 to +33. On Table 1 the results of the simulations with White Gaussian Noise are shown. For each one of the tests, a noise signal with a length of 1000 samples was generated and then delayed the corresponding number of samples. The expected result for each one of the signal would be Delay + DESP, which means Delay+20.

Delay / δ	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
White Gaussian Noise	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Delay / δ	1	2	3	4	5	6	7	8	9	10	11	12	13
White Gaussian Noise	21	22	23	24	25	26	27	28	29	30	31	32	33

Table 1: Position of the Delta obtained inserting White Gaussian Noise in the LMS algorithm.

6.1.2. Recorded signals (MONO)

Once the LMS has been tested for random noise signals, the time had come to prove the system with human voices. Two possibilities were considered. The first one consisted on recording the signals in stereo with the two microphones. This way, each microphone would capture a different signal and a delay would exist between them (Figure 26). The other possibility was to record the signals in mono. The microphones would record only one signal, and, as in WGN, a delay should be applied (Figure 27). Although the discrete domain only accepts integer number, the delay can be fractional (4.2). Building a Dirac delta like in (6.1) is easy for an integer value of N. However, if N is fractional the design of the filter is more complicated. APENDIX A explains in detail a method to build Fractional Delay filters. All Fractional Delay tests were carried out using this method.

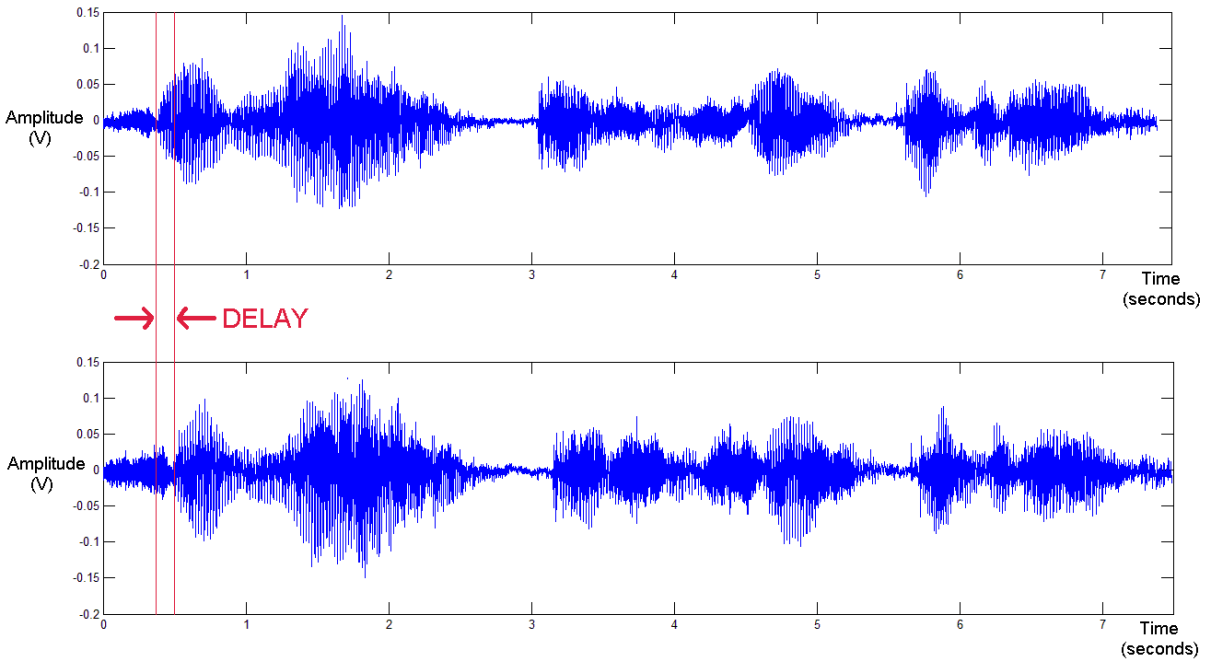


Figure 26: Stereo recorded signal emphasizing the delay.

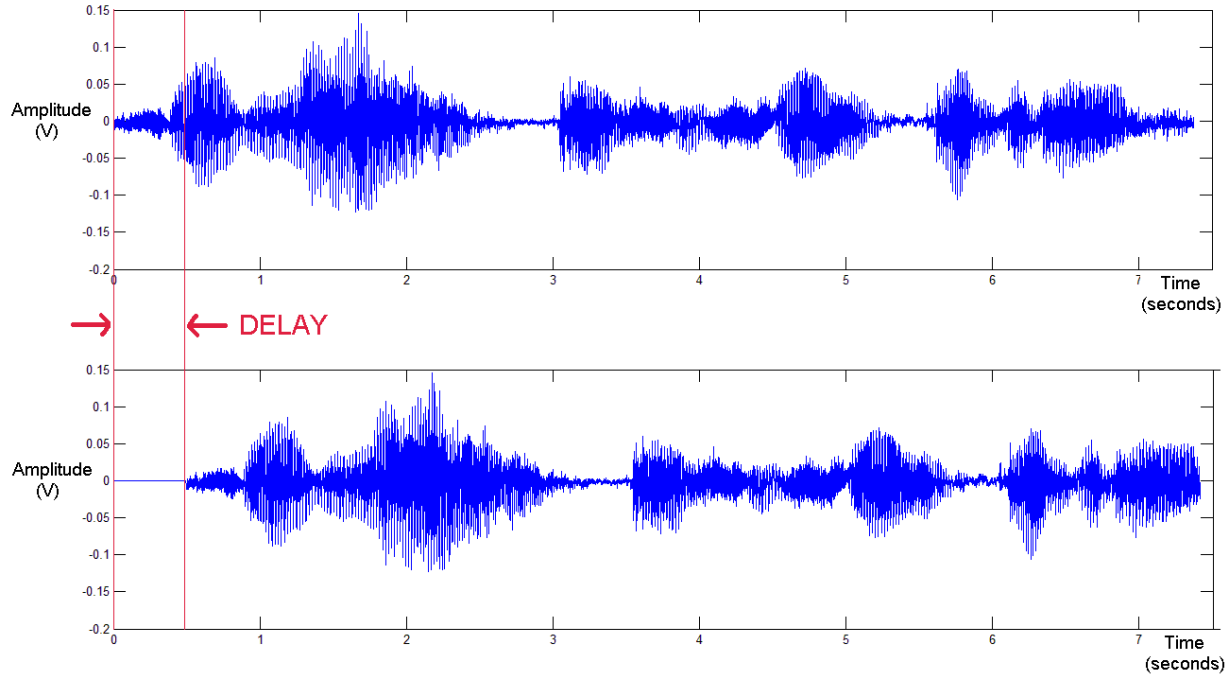


Figure 27: Mono recorded signal and manually delayed signal.

The first option simulates a real situation which makes it more accurate whereas the second is utopist (in real systems, there is no frontwave). However, the second option allows more flexibility in simulations, since with only one signal many tests can be performed just by

varying the delay. With the first option, one recording is necessary to test each one of the delays. So it was decided to choose the second option to check the LMS algorithm and the first option for further tests of the whole system.

The signals recorded came from a vast range: women, men, musical signals, etc. Since the system only takes care of the delay, the origin of the signal should not matter.

Delay / δ	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
Expected result	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Female 1	6.98	7.99	9.01	10.01	11.00	11.99	12.99	13.99	14.99	16.00	17.00	18.00	18.99	19.99
Female music 1	6.98	7.97	8.96	9.96	10.99	12.02	13.04	14.05	15.02	16.00	16.99	17.99	18.99	19.99
Male 1	6.99	7.99	9.01	10.00	10.98	11.99	13.01	14.00	14.99	15.99	17.00	18.00	19.00	20.00
Male 2	6.99	7.99	8.99	9.99	10.99	11.99	13.00	14.01	15.00	16.00	17.00	17.99	18.99	19.99
Male music 1	7.00	8.01	8.99	9.98	10.99	12.01	13.00	13.99	14.99	16.00	17.00	18.00	18.99	19.99
Female 2	6.99	8.01	9.00	9.99	10.99	12.00	13.00	13.99	14.99	16.00	17.00	17.99	18.99	19.99
Female music 2	6.99	8.00	8.99	10.00	10.00	12.00	12.99	14.00	15.00	15.99	17.00	18.00	18.99	20.00
Female 3	7.11	7.86	8.64	9.48	10.39	11.37	12.39	13.43	14.45	15.45	16.46	17.51	18.60	19.73
Male 3	7.01	7.99	8.99	9.99	11.00	12.00	13.00	13.99	14.99	16.00	17.00	17.99	18.99	20.00
Female music 3	6.99	7.99	9.01	10.01	10.99	11.98	12.98	13.99	15.01	16.001	17.00	18.00	18.99	19.99
Male 4	7.57	8.55	9.55	10.61	11.73	12.87	13.97	14.98	15.89	16.75	17.57	18.40	19.28	20.22
Average	**7.05**	**8.03**	**9.01**	**10.00**	**10.92**	**12.02**	**13.03**	**14.04**	**15.03**	**16.02**	**17.00**	**17.99**	**18.98**	**19.99**
Error	**0.05**	**0.03**	**0.01**	**0**	**0.08**	**0.02**	**0.03**	**0.04**	**0.03**	**0.02**	**0**	**0.01**	**0.02**	**0.01**

Table 2: Results for different recorded signals with integer delays (in samples).

6.1.3. Fractional Delay

As before, the expected value for each column is Delay+20. The first signal of the Table below is a pure Dirac Delta, used to show the best approximation to the expected value. The fractional values were taken in steps of 0.1 between 2 and 3.

Delay / δ	2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3
Dirac Delta	22	22.135	22.274	22.421	22.576	22.384	22.485	22.601	22.723	22.850	23
Female 1	22.00	22.132	22.277	22.426	22.572	22.381	22.482	22.590	22.712	22.856	22.99
Female music 1	21.97	22.133	22.275	22.427	22.571	22.382	22.481	22.600	22.722	22.857	22.98
Male 1	21.99	22.109	22.246	22.399	22.554	22.368	22.467	22.582	22.707	22.838	23.00
Male 2	21.99	22.131	22.271	22.422	22.579	22.385	22.482	22.602	22.728	22.852	22.99
Male music 1	22.00	22.132	22.278	22.424	22.586	22.389	22.491	22.601	22.725	22.850	22.99
Female 2	22.00	22.136	22.277	22.424	22.578	22.385	22.488	22.600	22.721	22.853	22.99
Female music 2	21.99	22.134	22.275	22.423	22.577	22.385	22.489	22.602	22.724	22.857	23.00
Female 3	21.95	22.096	22.244	22.398	22.557	22.361	22.469	22.587	22.713	22.849	22.99
Male 3	21.99	22.131	22.272	22.419	22.573	22.381	22.485	22.597	22.719	22.852	22.99
Female music 3	21.99	22.132	22.273	22.420	22.574	22.382	22.485	22.598	22.720	22.852	22.99
Male 4	22.17	22.112	22.254	22.401	22.556	22.364	22.468	22.581	22.705	22.838	23.12
Average	**22**	**22,126**	**22.268**	**22.417**	**22.571**	**22.379**	**22.481**	**22.595**	**22.718**	**22.850**	**23**
Error	**0**	**0,026**	**0,068**	**0,117**	**0,171**	**0,121**	**0,119**	**0,105**	**0,082**	**0,05**	**0**

Table 3: Results of applying Fractional delay to the previous signals (in samples).

6.2. Real system

After testing the system for non real signals, the next level needed to be reached. By using real stereo signals, more significant conclusions could be drawn. The results presented below will clear up whether the system is useful or not.

All the tests were carried out in the same environment to make invariant as many factors as possible. The system is supposed to work correctly in applications like videoconferencing or robots, which implies random environmental conditions. For that reason, the room chosen to make the records was a regular room (not anechoic). The echo produced by the reflections in the walls was also intended to be weak, so it did not interfere in the measurements. Different kinds of signals were used (women and men, high-pitched and deep, still and in movement, etc) in order to test a higher number of situations and make the system more realistic.

In this kind of experiments, it is very important to install correctly the different components. First, the distance between microphones must stay constant and, in our case, equal to 10 cm. For that the microphones were attached to a ruler, with the previous distance of separation between them. Then, the ruler was fixed to a structure around 1,7m high. The aim of this was to prevent the possible reflections on the floor which would bring undesired signals. Since the speaker's voice could come from different angles, it was necessary to know the exact angle before recording; otherwise it would be impossible to know if the results were right or wrong. To achieve this task, a board like the one shown below was built (Figure 28). This way, the angle was written down before recording and then compared with the system's result. The point of origin in this diagram had to coincide with the middle point of the microphones.

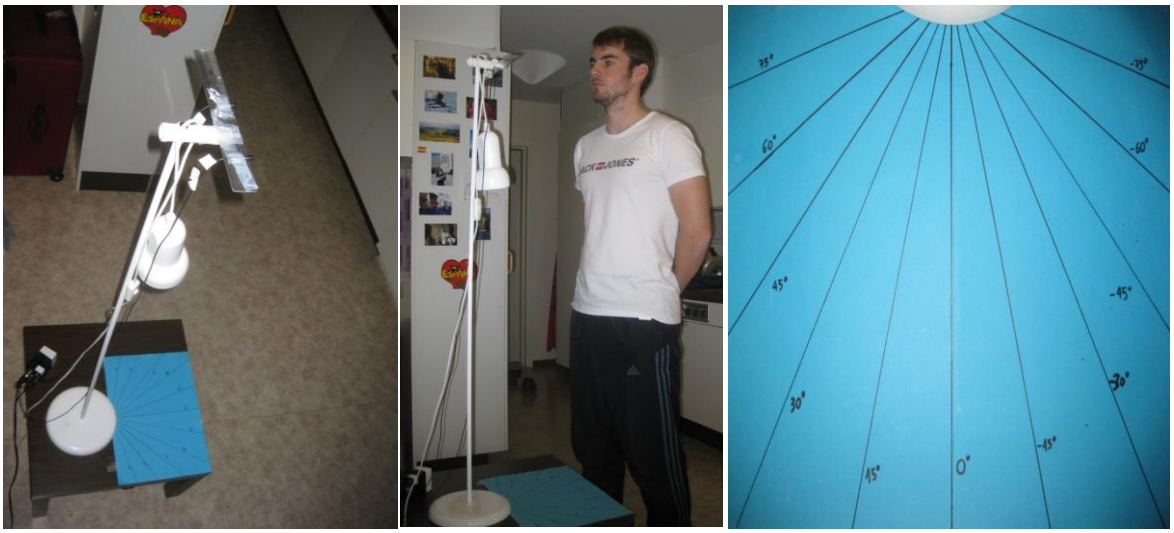


Figure 28: Pictures showing the system position, its height and the board used to know the angles.

Any mistake committed in any of the previous steps would cause the malfunction of the system. For instance, let's consider a small mistake of 0.5cm in the distance between microphones and let's consider a speaker standing in the direction -55º. The distance d is equal to 9.5cm, so the signals arrive to the microphones with a relative delay of 10 samples. But, since the system is designed for d=10cm, when it detects a 10 samples delay, it return a value of α around 50º. So a short disarrangement in the microphones provokes a big error.

As mentioned before, two kinds of scenarios were considered. First, the speakers were recorded from a fixed direction. For each one of these recordings, the system was supposed to return a unique value. In the second scenario the speakers were recorded in movement between two positions. In those situations, the following problem was detected;

During the movement, the speaker walks through many different directions and the system is supposed to return all the angles corresponding to these directions. However since a unique recording is done, a unique signal is introduced in the system, thus a unique angle α is returned. But this value is meaningless since the expected result was a group of angles describing the path followed by the speaker. A unique value of α cannot describe a whole trajectory. The following picture illustrates the problem.

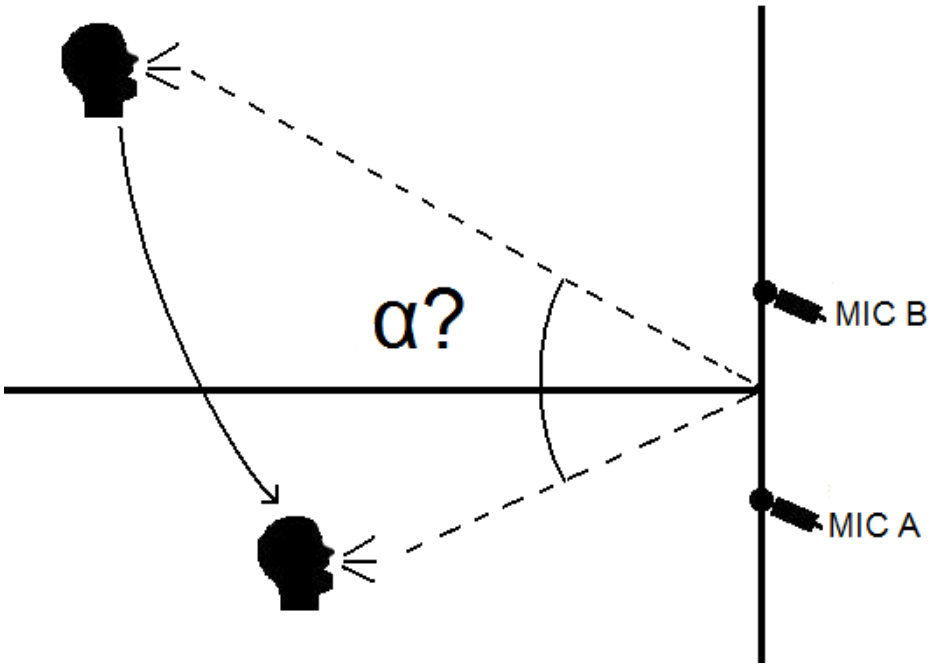


Figure 29: Speaker in movement: only one angle returned.

To avoid this problem it was decided to divide the recorded signal into a certain number of sub-signals and then introduce them in the system. This process generates one angle for each one of the sub-signals and so indicates all the directions covered by the speaker (Figure 30).

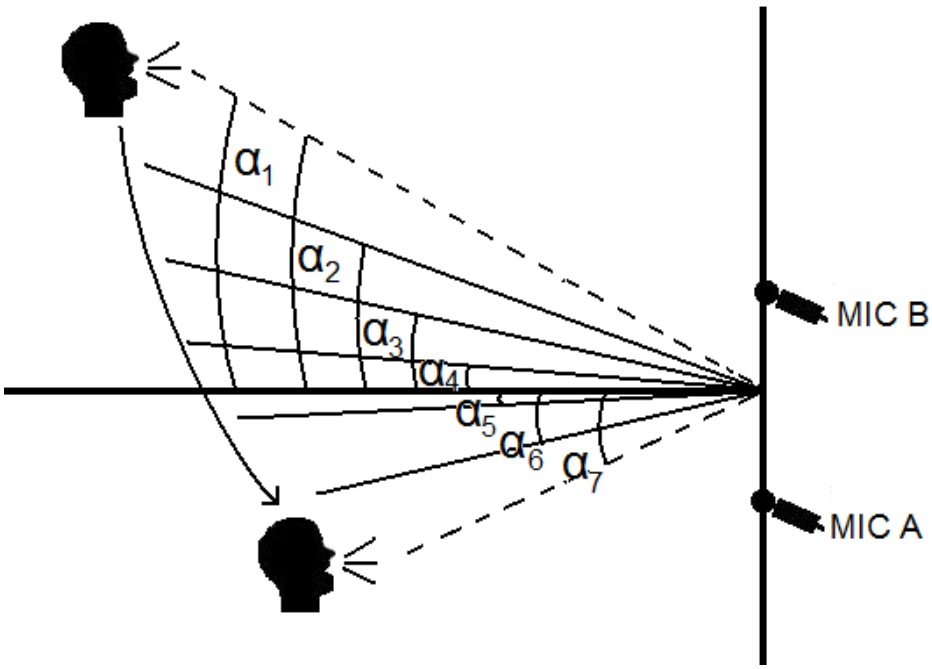


Figure 30: Speaker in movement: several angles returned.

In this example, seven angles are returned. This allows a better knowledge of the speaker's movement. The following picture (Figure 31) shows a real signal, captured from a speaker in movement, and generated with MatLab. Each signal corresponds to the capture made by each microphone. Three specific positions are remarked in red due to its relevance on this explanation: In the step number 1, the lower signal is delayed in comparison to the upper, in step number 2 the delay between them is zero and in step number 3 the situation is inversed and the upper signal is delayed in comparison to the lower.

Considering the theoretical explanations given in previous sections, the movement of the speaker can be approximately derived: In step number 1, the speaker stands closer to MIC B (positive delay), in step 2 he is equidistant to both microphones (no delay)and then in step 3 he is closer to MIC A (negative delay). Hence a movement from right to left can be predicted by a quick analysis of the recording. But, if these two whole signals were introduced in the system, the result would be a unique α which would not describe the real movement of the speaker.

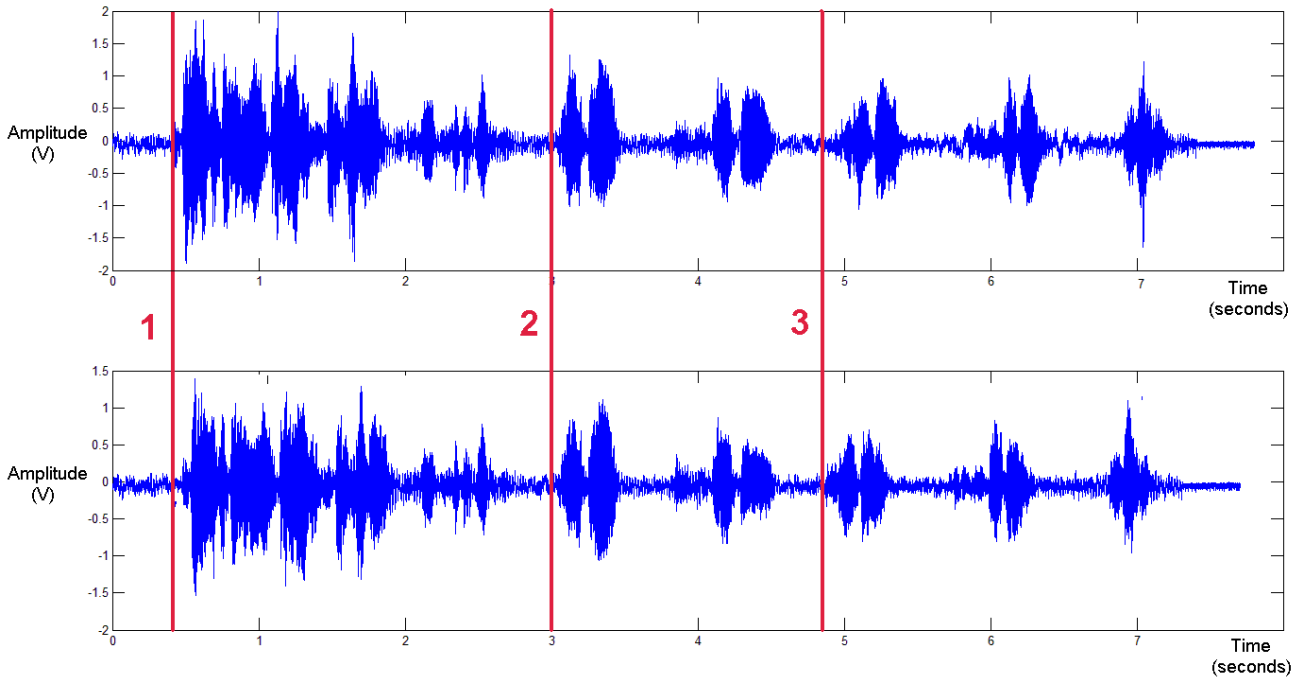


Figure 31: Stereo signal of speaker in movement.

On the contrary if the signals are divided and processed separately, the results would ideally be right. Continuing with the example from above the division would be the one shown in the next figure. There are seven results that would clarify the exact path of the speaker.

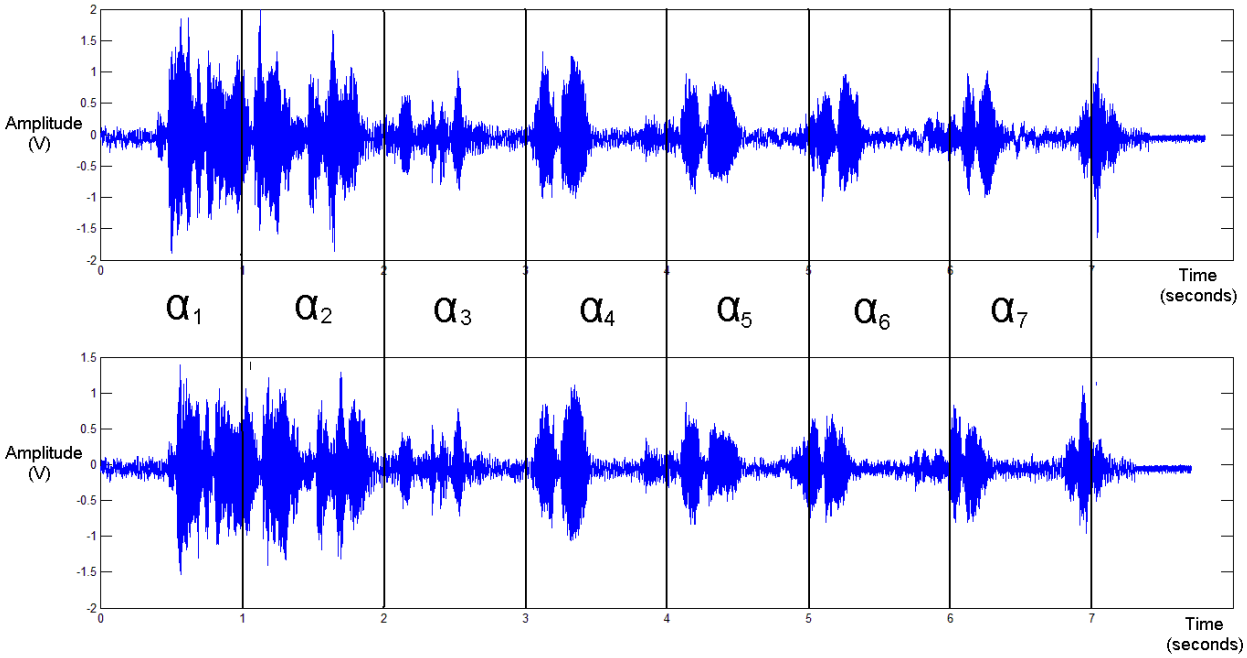
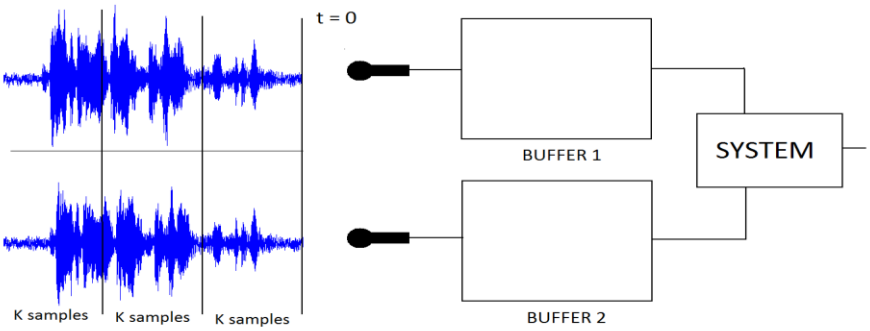
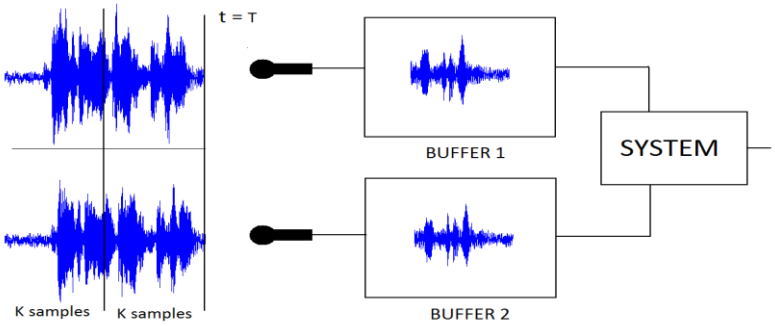


Figure 32: Signal divided uniformly.

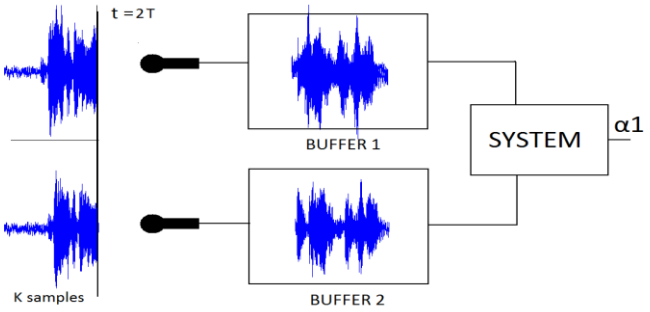
Once the signal is recorded, it is divided and processed in a loop. Figure 33 illustrates how a real time system would work. First the signals would automatically be divided in pieces of k samples and then processed. The mechanism would look like a pipeline system. The signal would go through the microphones and would reach a buffer in portions of k samples (a). Once the first k samples would get into this buffer (b), they would be sent to the system and a result α_1 would be produced. At the same moment (c), since the buffer is empty, the next k samples would enter the buffer and the process would be repeated (d).



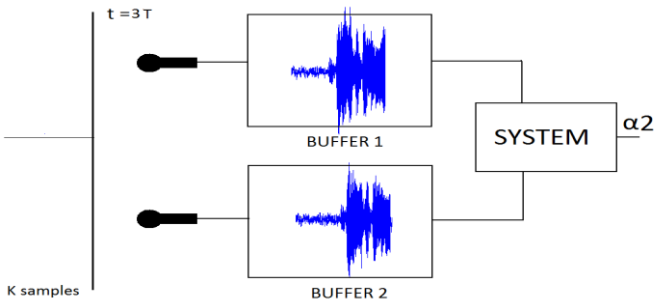
(a)



(b)



(c)



(d)

Figure 33: Ideal pipeline process to get the angles.

The choice of k is delicate, since it may be harmful for the efficiency of the whole system. It is necessary that the time spent in processing the signals in the system is lower than the time spent by the k samples to get into the buffer. Thus for a high value of k the system risks to be discontinuous. In the other hand, it must not be too small because the LMS needs a certain number of samples to work correctly. Taking all this in account, it was decided to choose k=22050 which represents 0.5 seconds. Actually with this length the system is able to return acceptable results approximately in 0.2 seconds, so, the processing time does not interfere in the possibility of building a real-time system. Furthermore, considering a speaker moving at a normal speed, the angle traveled in half of a second is not too big. For instance, a subject moving at 3 km/h (which correspond to 0.83 m/s) walks 41 cm in 0.5 seconds. Considering that the subject stands at 2meters in front of the microphones, he travels 11˚ in this period of time, which is reasonable. After following all these constraints, the results obtained are the ones shown below.

Signal / α	Male1	Male 2	Male 3	Female 1	Female2	Male 4	Female 3	Male 5	Average angle	Error commited
-90˚	-82.34˚	-83.45˚	-90.00˚	-90.00˚	-84.64˚	-84.46˚	-89.77˚	-87.92˚	**-86,57˚**	**2,08˚**
-75˚	-67.67˚	-74.63˚	-82.03˚	-69.83˚	-76.57˚	-77.68˚	-71.43˚	-72.88˚	**-74,09˚**	**2,12˚**
-60˚	-58.91˚	-65.97˚	-58.09˚	-55.71˚	-61.71˚	-61.37˚	-56.91˚	-56.81˚	**-59,43˚**	**3,19˚**
-45˚	-46.28˚	-38.28˚	-48.04˚	-46.32˚	-45.38˚	-39.78˚	-46.74˚	-45.54˚	**-44,54˚**	**0,54˚**
-30˚	-33.27˚	-30.93˚	-37.03˚	-30.54˚	-30.09˚	-30.81˚	-32.73˚	-34.88˚	**-32,53˚**	**4,88˚**
-15˚	-17.00˚	-16.38˚	-20.65˚	-14.88˚	-20.62˚	-15.22˚	-16.13˚	-18.98˚	**-17,48˚**	**3,98˚**
0˚	2.80˚	-2.23˚	0.42˚	1.99˚	3.40˚	1.20˚	1.64˚	2.03˚	**1,40˚**	**2,03˚**
15˚	13.96˚	21.64˚	13.35˚	16.76˚	14.16˚	15.63˚	13.21˚	14.57˚	**15,41˚**	**0,43˚**
30˚	31.94˚	37.42˚	31.80˚	31.75˚	34.95˚	38.23˚	31.27˚	30.10˚	**33,43˚**	**0,1˚**
45˚	43.98˚	53.04˚	48.56˚	43.01˚	47.91˚	43.44˚	47.16˚	46.20˚	**46,66˚**	**1,2˚**
60˚	63.19˚	63.31˚	58.82˚	58.02˚	66.43˚	60.35˚	56.65˚	57.23˚	**60,50˚**	**2,77˚**
75˚	76.34˚	82.23˚	76.90˚	74.97˚	79.04˚	77.27˚	80.03˚	78.79˚	**78,19˚**	**3,79˚**
90˚	90.00˚	90.00˚	90.00˚	79.93˚	90.00˚	88.10˚	90.00˚	86.07˚	**88,01˚**	**3,93˚**

Table 4: Results of the angle for fixed positions from -90˚ to +90˚.

As explained in the beginning, another target of the project was to make it able to follow a speaker in movement. In the following graphics, the results of the experiments in motion are presented. Each line corresponds to the path followed by a speaker. Two scenarios were considered: in the first one the speakers were asked to move from 0˚ to +90˚and in the second one from 0 to -90˚, walking uniformly and at constant speed.

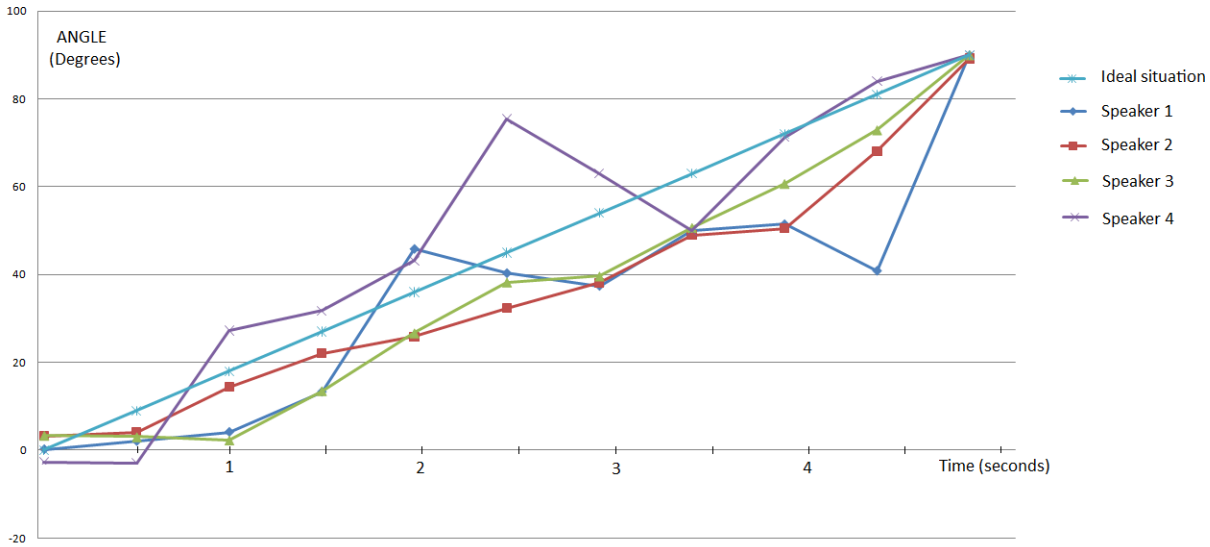


Figure 34: Graphic for a speaker moving from 0 to +90˚.

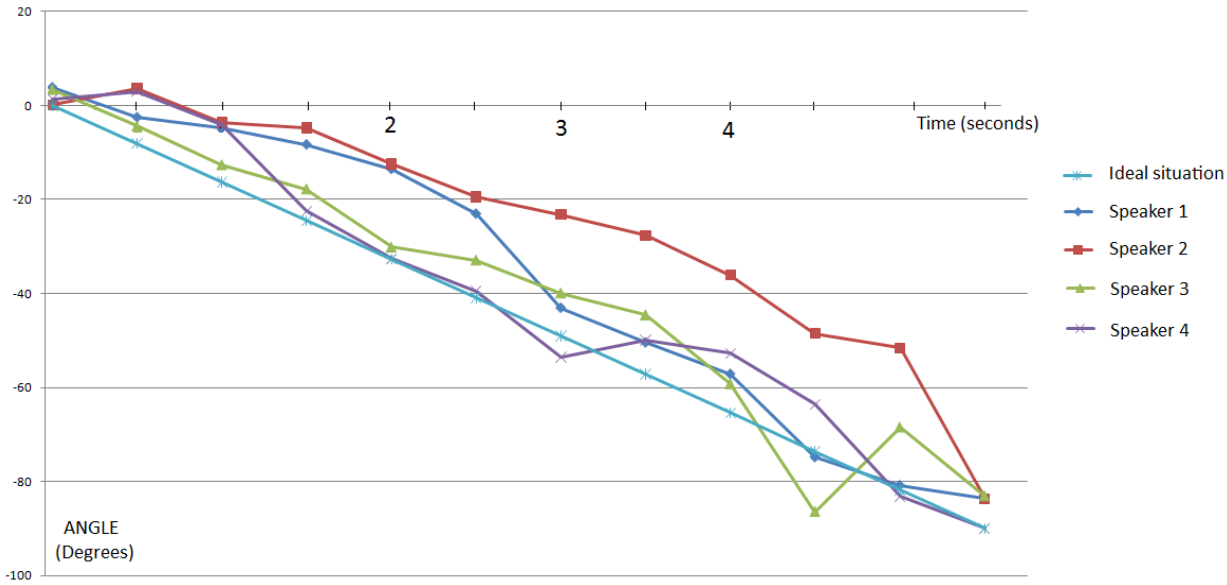


Figure 35: Graphic for a speaker moving from 0 to -90˚.

7 Result analysis

According to the previous data, an analysis of the results would be performed. In order to make it simpler for the reader, the explanations would follow the same order as the results: first non-real signals and then the real ones.

7.1. Non-real signals

7.1.1. White Gaussian noise

The tests carried out with white Gaussian noise signal aimed to prove that LMS algorithm worked correctly. Since the delays were introduced manually before running the algorithm, both signals would have a defined frontwave and same amplitudes, so the determining of the delay was assumed to be highly precise.

As mentioned previously, when a delay was introduced, the expected filter was a Dirac delta centered in Delay + DESP. Since DESP was equal to 20 and since the delay range varied from -13 to +13, the results were supposed to be included in the range +7 to +33.

Table 1 shows that the expected results were reached. This proves that the designed LMS algorithm operates with total precision for random noise signals. This conclusion may not seem to be very convincing, considering the fact that the system must work with voice signals. Nevertheless by proving that the algorithm works, the first milestone was reached, which allowed pointing at the next target: recorded voice signals.

7.1.2. Recorded signals (MONO)

As mentioned in a previous chapter, mono recorded signals were used to test the LMS algorithm for human voice signals. A unique signal permits several tests by varying the delay. On the contrary of the White Gaussian noise (where only LMS was run), here the second block of Figure 20 was also tested. This needed to be done because the filters obtained from the LMS were not pure Dirac deltas but sinc functions. For these tests many signals from different sources were chosen. To cover a bigger range, men and women from different distances were

recorded. Furthermore, a few musical signals were also used. The aim of this was to check if the source of the signal had any influence in the behavior of the algorithm.

To draw any conclusion regarding this, Table 2 must be analyzed. Comparing the results of the tests with the expected results (on the first line of each table), it is easy to observe that the highest error committed is 0.08 samples. However this error happens in a few cases. The tables show that the most common deviation is around 0.02 samples.

These results are convenient enough to affirm that the algorithm has a correct behavior for voice signals. Besides, the operation is correct regardless of the origin of the signals. However, good results were expected in these tests too since the delay is artificial. In fact, the presence of frontwave makes easier the determining of the delay. Nevertheless the conclusions drawn with these tests do not only concern the LMS algorithm. Actually it was proved that the second block of Figure 20 worked as expected.

7.1.3. Fractional delay

The aim of running fractional delay tests was to determine the accuracy of the system. The delays were introduced manually before running the LMS algorithm. As in the previous ones, a unique signal allowed many tests. In order to compare suitably, the same signals used for integer delay were used here. Besides, a pure Dirac delta was tested too. The aim of this was to check how close from the real value it gets. Actually the results obtained with a pure Dirac delta would always be the most precise ones.

The results are shown in table 3. On the first line of each table stand the results of the Dirac deltas. The most remarkable aspect of these results is that the more the delay stand close to the integer numbers (2 or 3), the more exact is the delay returned. For instance, when the signals are delayed 2.4 samples, the average result (which was supposed to be 22.4) is 25.571 whereas in the case of 2.1 samples, the result is 22.126. Thus the table shows that the highest error committed is 0.171 samples, which won't cause a significant variation in the final direction (0.0226° variation).

For this reason the conclusion that can be drawn is the same as in Integer delays. Hence the accuracy was proved for fractional delays, which shows that the system is strong enough to handle all the possible situations.

7.2. Real system

The results of the stereo recorded signals are the ones that will determine how well the system operates. Before drawing any conclusion, it is necessary to take some considerations. The obtained results will be compared with the desired ones. In these tests the results were not expected to be as exact as they were in the previous ones. After a thorough study, some reasons were determined as possible causes from these inaccuracies.

First of all, a bad design of the software was considered. Nevertheless, attending to the exactness shown for theoretical signals, this possibility was practically discarded. Another important issue to take care of was the reflections. As shown on the pictures above, there are no walls or flat surfaces close to the microphones that could disturb the recordings, so this possibility was also rejected.

Thus, the main cause for the mistakes can be a bad positioning of the speaker with respect to the microphones. That is the principal reason why the tests were implemented rigorously. However, even taking all the precautions if the speaker's mouth is not perfectly aligned with the direction where he is supposed to be, some errors can occur. For that reason, a margin of error of $\pm 4°$ was chosen. If the direction obtained is inside the margin, the test is considered a success. To choose this value it was taken in account the fact that for several applications the exactness of 100% is not always necessary. For example in videoconference, if the system commits a mistake of $\pm 4°$ the target is still pointed by the camera.

The first tests carried out were the one with still speakers. For each direction, a unique recording was made and then the signals were introduced to the system. Figure 36 shows the percentage of success of these tests. Three kind of results are identified: first the ones considered as a total success (belonging to $\pm 4°$), then the ones considered partial success ($\pm 8°$) and then those considered a fail ($> \pm 8°$). According to the diagram, fail only exists for directions $\pm 75°$ and $\pm 90°$. Even so, the percentage of fail in these cases is lower than 20%. In the other hand, in the range from -60° to $+60°$ there is no failure and the percentage of total success rises increasingly when the directions get close to 0°.

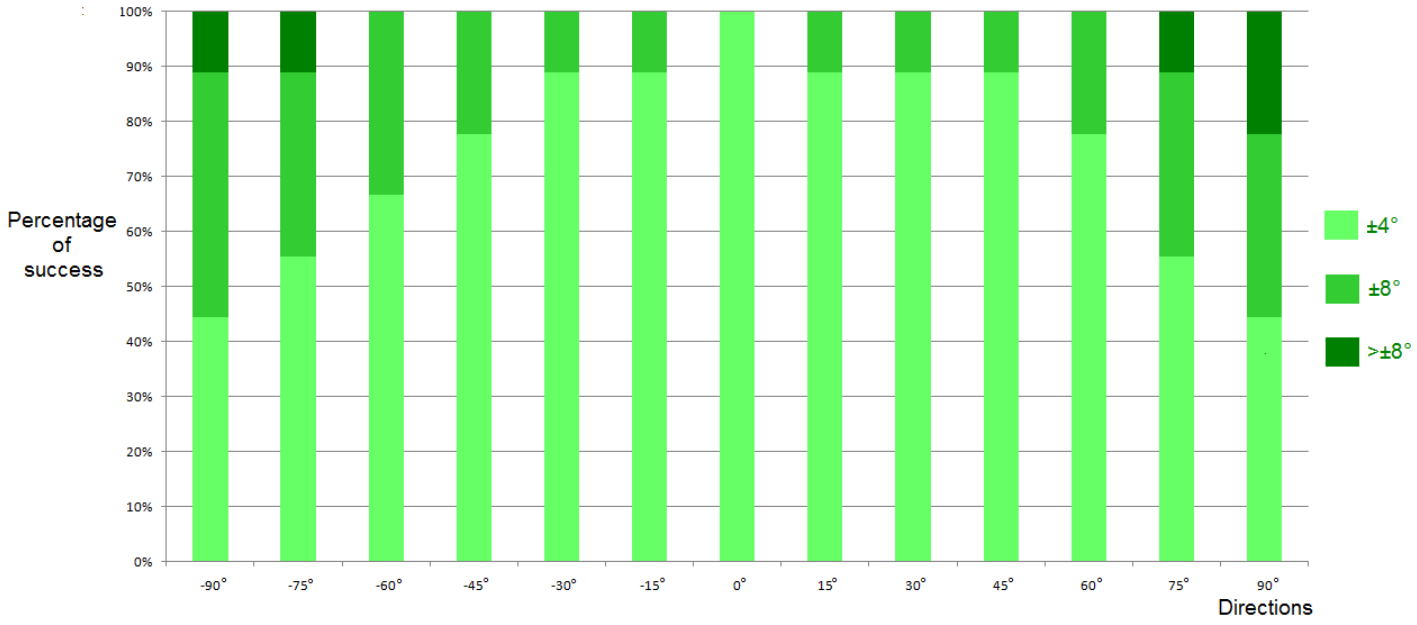


Figure 36: Diagram showing the percentage of success on the tests for still sources.

The explanation for these results lies on trigonometry. As referred in a previous chapter, first α' (3.12) is calculated and then α ((3.13) and (3.14)). Considering Figure 6, it is clear that the more the values get close to +90° or -90°, the more the function is nonlinear. Thus the closer it gets to the borders, the higher the precision should be. For instance, the impact of a mistake of 0.5 samples is much higher in this region than in the region close to 0°. Choosing the same range of colors as in the Figure 36, the problem is illustrated on Figure 37.

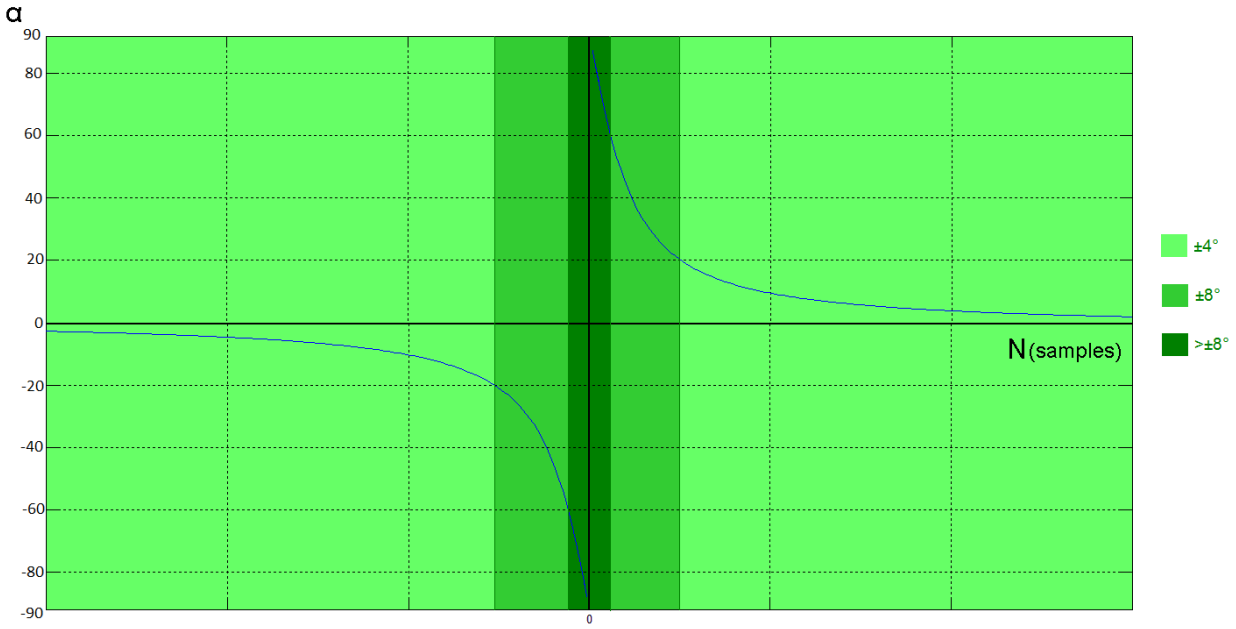


Figure 37: Graphic explanation of the possible errors committed.

As explained before, dark green represents the failure and the others the success. Thus, the system operates with 100% of effectiveness from -60° to +60° and with at least 80% of from ±60° to±90°. All this results are actually for still speakers. Now, the results of the tests in motion will be analyzed.

Two different scenarios were considered: speakers in movement from 0° to +90° and then from 0° to -90°. The aim was to determine if the system is able to follow correctly a moving source. As mentioned above, the speakers were asked to move at constant speed and not too fast, in order to have enough values to do a meaningful study. Looking at the data obtained for still sources, the results were expected to be adequate at least in the range −60° to +60°.

Figures 36 and 37 present the results graphically. The light blue line represents a theoretical uniform movement, so the closest the other lines stand from this one, the better the result is. To understand the differences existing between the obtained results and the ideal one, the human factor must be taken in account. Actually the speakers were not moving uniformly even if they tried. This is revealed on the graphic as peaks in both sides of the light blue line. Nevertheless, in both figures the results follow the same path as the speaker: they start at the same point, follow a similar route and end in the same final position. Thus it can be affirmed that the system is able to follow a moving source. Furthermore, since the time response (0.2 seconds) is lower than the capturing signal time (0.5 seconds) the system could easily work in real time.

These analyses prove that the system can operate suitably for all kind of signals and for the desired range of directions. Although it may be improved, the results are satisfying enough to ensure that it can work without any significant modification.

8 Conclusions

After all the simulations and the subsequent results, some conclusions can be drawn. First of all it was proved that the LMS algorithm worked suitably. This means that the parameters were chosen adequately to guarantee a good operation. Hence the design was strong enough to process signals regardless of their origin (noise, voices, etc.)

As mentioned previously, two kinds of signals were tested. Firstly the signals were recorded coming from fixed sources and then from sources in motion. In both cases the results showed a good performance of the system. Furthermore, with those results it was proved that the distance between source and microphones does not matter. Actually the only relevant aspect is the delay existing between the signals.

For fixed sources, the operating range extends from -90° to +90°, which covers the whole front semicircle. For this kind of signals the system has proved a high effectiveness. For sources in movement, the precision was not so wide. Nevertheless it remained 100% accurate in the range from -60° to +60° and 80% in the rest. In order to handle long signals in short time and with acceptable precision, the recorded signals were divided in blocks of 22050 samples (corresponding to half a second).

Finally two considerations must be taken in account when using this system. The first one is related to the reflections. The microphones must not be placed near flat surfaces or walls in order to avoid rebounds. The second one has to do with location in movement. If the speaker moves too fast the system would not be able to follow him and so bad results would be returned.

9 Future Work

After designing and analyzing the system it would be interesting to mention some aspects that can be improved or researched to increase the number of application in which the system could be applied.

As mentioned in a previous section, the system is operating in partial-real time since the process is performed in two steps (record and then process). A possible improvement would be designing it in full real time. This way, the recorded signals would automatically be transferred to the MatLab system, without the intervention of the user.

In the proposed system only the direction of the source is obtained but not the distance. A reasonable future work could be designing an improvement to obtain the distance, basing the design on the current system. For instance, adding a third microphone in the array and running two LMS algorithms in parallel could theoretically enable to get the distance. Thus it could be interesting to study the influence of this third microphone in the system.

In order to make the system more real it could be possible to increase the precision of the in-movement scenarios. An option could be analyzing and discussing the consequences of reducing the time intervals in which the sound signals are divided (in the current system it was half of a second).

It would be also useful to complement this system with other systems, such as people tracking. For instance, considering an in-movement camera following the speaker, the recognition patterns of the people tracking could at the same time recognize his. Thus the camera could relocate itself to make the face be on the centre of the screen.

Finally the system can be improved researching and designing specific filters to prevent noise and reflections. This way the system could be used in noisy environments and the microphones placed near the walls or flat surfaces.

APENDIX A: Building a Fractional Delay Filter

The manual delay made with MatLab by creating Deltas is very simple and useful to carry out tests of the system. Unfortunately it only allows delays of integer values. That means for example, a signal can be delayed three samples or four samples, but not three and a half. Thus the simulations are not as reliable as they could be. In order to obtain higher accuracy, a fractional delay all pass filter was implemented.

Among different types of fractional delay filters, the maximally flat one could satisfy the requirements. A discrete time all-pass filter has a transfer function as below

$$A(z) = \frac{z^{-N}D(z^{-1})}{D(z)} = \frac{a_N + a_{N-1}z^{-1} + \cdots + a_1 z^{-(N-1)} + z^{-N}}{1 + a_1 z^1 + \cdots + a_{N-1}z^{-(N-1)} + a_N z^{-N}} \quad \text{(A.1)}$$

Where a_k are the filter coefficients and N is the order of the filter. The coefficients a_k can be designed for having maximally flat group delay D with the following formula

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^{N} \frac{D - N + n}{D - N + k + n}, k = 0,1,2, \dots, N \quad \text{(A.2)}$$

Where

$$\binom{N}{k} = \frac{N!}{k!\,(N-k)!} \quad \text{(A.3)}$$

specifies the k-th binomial coefficient. The coefficient a_0 is always equals to 1, which makes normalization unnecessary. If D > N, the poles are inside the unit circle in the complex plane. In this case, the filter is stable. Since the nominator is a mirrored version of the denominator, the zeroes lie outside the circle. For the same reason, the radii of the poles and the zeroes are inverse of each others. That makes the amplitude response be flat.

$$\left|A\left(e^{-jw}\right)\right| = \left|\frac{e^{-jwN}D\left(e^{-jw}\right)}{D(e^{jw})}\right| = 1 \quad \text{(A.4)}$$

In this filter the phase is linearly dependent with D. To design the filter, the coefficients in (A.1) must be calculated with appropriate values for N and D. These parameters are chosen according to the desired delay. As explained before, the possible delays can vary from +7 to +33 samples, so the order N must be an integer remaining between these two values. D is a fractional number indicating the desired delay. Its value must follow the inequation

$$N - 0.5 < D < N + 0.5 \qquad \text{(A.5)}$$

So according to (A.5) D, N can be changed. For instance, for a delay D=21.6 samples, N needs to be 22, but for D=21.4, N must be 21. The more accurate the fractional delay filter is, the more number of angles can be obtained. Considering a delay between signals of 11.6 samples, the angle obtained should be $\alpha = -63.4º$. If the fractional delay is not used, the test only could be made for D=11 or D=12, which leads to angles of $\alpha = -58º$ and $\alpha = -67.7º$ respectively. In this case the difference is almost ten degrees, which is unacceptable.

In the contrary as the case with the integer delays, here the filter is not a Delta centered on the delay N. Actually, MatLab cannot index non-integer values, so instead of returning a Delta, the output is a sinc function. Figure 38 shows the process followed by a integer delay filter to obtain the delay in samples (FFT-Phase-N).

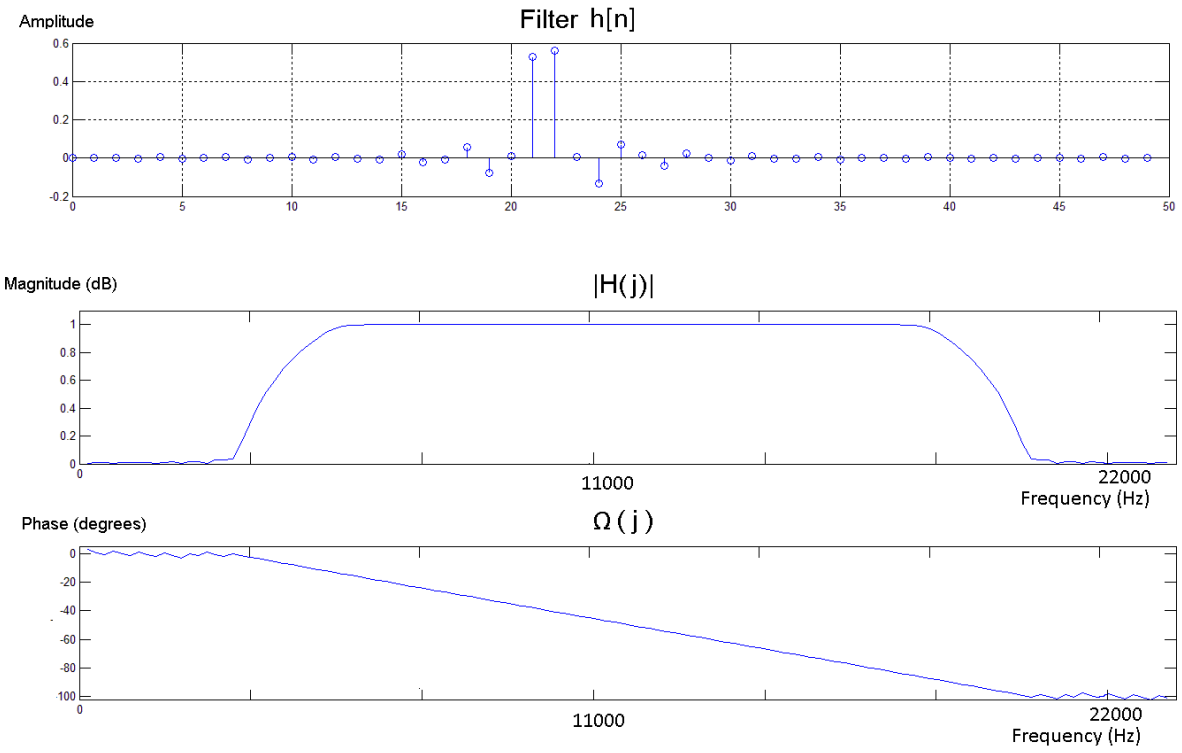


Figure 38: Steps to get the number of samples delayed N for a fractional delay filter.

APENDIX B: MatLab methods

System.m

```
% In this program, the signals are divided in pieces of lengData
samples
% and process separately. This program calls two functions:
% f_adap
% get_angle
% At the end it returns the direction in degrees.


% CONSTANT VALUES
fs=44100;                          % Sampling frequency (Hz)
d_micro=0.1;                       % Distance between microphones (m)
c=340;                             % Speed of sound (m/s)
muestrasMAX=ceil(d_micro*fs/c);    % Maximum number of samples Nmax

DESP=ceil(muestrasMAX*1.5);        % Delay we insert in the micro 2
                                   % We leave 50% of margin of error.

lengData=44100*0.5;                % Number of samples in which the
voice
                                   % signals are divided to be
processed.

                                   % All values depend on fs and d_micro
in
                                   % case it was necessary to change
them.

signal1=wavread('90.wav')';        % Importing the file to process.

% LOOP WHERE THE DIFFERENT PARTS OF THE IMPORTED FILE ARE PROCESSED
for k=lengData:lengData:length(signal1)
tic;                               % Measure of time tic;toc;

signal=signal1(1,k-(lengData-1):k);  % Signal in MIC B
d=signal1(2,k-(lengData-1):k);       % Signal in MIC A

% NORMALIZATION PROCESS
M1=max(abs(signal));                 % Maximum of channel 1
M2=max(abs(d));                      % Maximum of channel 2
M3=max(M1,M2);                       % Normalization value
signal=signal/M3*2;                  % Normalizing
d=d/M3*2;

% LMS ALGORITHM
hDESP=[zeros(1,DESP) 1];             % Filter to delay the signal DESP
samples.
d1=conv(hDESP,d);
P=50;                                % Parameters of the algorithm
mu=0.0117;

h0=zeros(1,P); h0(1)=0;              % Initialazing the adaptative
filter
```

```
[h y e]=f_adap1(signal,d1,h0,mu);     % Recursive function calculating
the
                                      % coefficients of the filter h(n)

% PROCESSING THE FILTER BEFORE THE FREQUENCY ANALYSIS.
h1=[zeros(1,DESP-muestrasMAX-3),h(DESP-muestrasMAX-2:length(h))];
h1(DESP+muestrasMAX+2:length(h1))=0;
h1(DESP+1)=h1(DESP+1)/2;
[B,I]=sort(h1,'descend');
H1=[zeros(1,I(1)-3),h(I(1)-2:I(1)+2),zeros(1,length(h)-(I(1)+2))];


% FREQUENCY ANALYSIS TO OBTAIN THE DELAY (IN SAMPLES)
% 1-FFT
lh=128;                               % Length of the FFT
H=fft(h1,lh);                        % FFT of the filter h(n)

% 2-ANGLE(+UNWRAP)
alpha=angle(fftshift(H));         % Obtaining the phase
q=unwrap(angle(fftshift(H)));

% 3-SLOPE
M=diff(q);                           % Obtaining the slope of the phase

% 4-SLOPE'S AVERAGE
lM=length(M)+2;                    % The slope M1 is not a unique value,
p1=floor(lM/2-4);                   % it's an array. So we calculate the
p2=ceil(lM/2+4);                    % average of the values, K.
K=mean(M(p1:p2));
Nprime=(-K*lh/(2*pi));               % Number of samples before
                                       % substracting DESP.

% 5-SAMPLES
if Nprime<0                           % Two possible cases: negative or
positive
    N=Nprime+lh;
    N=N-DESP;
else
    N=Nprime;
    N=N-DESP;
end

% CALLING THE FUNCTION WHICH RETURNS THE ANGLE
angleGRAD1=get_angle(N1,fs,d_micro);

if isreal(angleGRAD1)==1              % Security measures in case
    angleGRAD                         % the number is complex
else
    angleGRAD1=real(angleGRAD1)
end

timeElapsed=toc;                     % Time is kept in variable
timeElapsed
timeElapsed


end
```

Function *f_adap*

```
% PERFORMS THE CALCULATION OF THE COEFFICIENTS OF THE FILTER h(n).
% Inputs:  - x  = Signal in MIC A
%          - d  = Signal in MIC B
%          - h0 = Initial filter (equals to 0)
%          - mu = Step-size
% Outputs: - h  = Desired filter
%          - y  = Convolution between x and h
%          - e  = Error function
function [h,y,e] = f_adap(x,d,h0,mu)
% Implements the LMS algorithm.
%Inputs:   x(n) Original signal
%          d(n) Delayed signal
%          h0   Original filter
%          mu   Constant value
%Outputs:  h(n) Filter
%          y(n)= x(n)*h(n)
%          e(n) Error function (must be zero)

h=h0; P=length(h);
N=length(x);
y=zeros(1,N); e=y; % Reserve space for y[] y e[]
rP=0:-1:-P+1;
for k=P:N,
xx=x(k+rP);      % Last P inputs x[k], x[k-1], ... x[k-P]
y(k)=xx*h';      % Filter output: x*h Convolution
e(k)=d(k)-y(k); % Error
h=h+mu*e(k)*xx; % We update the filter coefficients.
end
end
```

Function *get_angle*

```
% OBTAINS THE ANGLE BY PERFORMING A CERTAIN NUMBER OF TRIGONOMETRIC
% CALCULATIONS. IT CALLS THE FUNCTION:
% hiper
% Inputs:  - N       = Number of samples
%          - fs      = Sampling Frequency
%          - d_micro = Distance between microphones
% Outputs: - angle   = Angle in degrees
function[angle]= get_angle(N,fs,d_micro)
if N~=0,
    j=0.1;                          % Steps
    x=-20:j:20;                     % x axis
    [y1]=hiper(N,x,-d_micro/2,fs);% Calling function hiper
    x1=round(length(x)/4);x2=round(length(x)/8);
    pendiente=(y1(x1)-y1(x2))/(j*(x1-x2));       % Slope
    if N>0
        angulorad=atan(pendiente);
        angulo1=angulorad*180/pi;
        angle=-90-angulo1;
    else
        angulorad=atan(-pendiente);
        angulo1=angulorad*180/pi;
        angle=90-angulo1;
    end

else
   angle=0;
end
end
```

Function *hiper*

```
% OBTAIN THE COORDINATES OF THE POSITIONS WHERE THE SPEAKER CAN BE.
% Inputs:  - muestras = Delay between signals in samples
%          - x        = Values of the x-axis
%          - xA       = x coordinate of MIC A
%          - fs       = Sampling frequency
% Outputs: - y1       = Values of the y coordinate of the speaker
function [y1]= hiper(muestras,x,xA,fs)

c=340;                                                  % speed of sound
(m/sec)
pot=2*ones(1,length(x));
dist=muestras*c/fs;                                     % distance to B
prime
y1=sqrt(dist^2/4-xA^2+(4*xA^2/dist^2-1)*x.^pot);        % formula with
following
                                                        % requisitions:
                                                        % xA=-xB; yA=yB=0

end
```

List of References

[1] Scott, James; Dragovic, Boris. “Audio location: Accurate Low-Cost Location Sensing”. Intel research Cambridge. Proceedings of the Third International Conference on Pervasive Computing. 2005.

[2] http://www.sonimalaga.com/pages/PCSA-CTG70.pdf

[3] Zieger, Christian. ; Brutti, Alessio. ; Svaizer, Piergiorgio. “Advanced Video and Signal Based Surveillance”.; CIT-IRST, Fondazione Bruno Kessler, Trento Italy; AVSS '09. Sixth IEEE International Conference; ppp 314-319; 2-4 Sept 2009.

[4]http://www.shotspotter.com/documentation/2010-02/ShotSpotter_Security_v6-3_4p_A4_2010-02-10_spa.pdf

[5] Parhizkari, Parvaneh. “Binaural Hearing-Human Ability of Sound Source Localization”. Master Thesis in Electrical Engineering. Blekinge Institute of Technology.Karlskrona, Sweden. December 2008.

[6] Yannis, Kopsinis; Elias, Aboutanios; Dean, A. Waters; Steve, McLaughlin. “Investigation of bat echolocation calls using high resolution spectrogram and instantaneous frequency based analysis”. Inst. for Digital Commun., Univ. of Edinburgh, Edinburgh, UK; Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop; ppp 557-560; Aug 31 2009 – Sept 3 2009.

[7] Houser, Dorian; Martin, Steve; Phillips, Mike; Bauer, Eric; Herrin, Tim; Moore, Patrick. “Signal Processing Applied to the Dolphin-Based Sonar System”. BIOMIMETICA, La Mesa, CA, USA. OCEANS 2003 Proceedings. Vol 1. ppp 297-303. 2003.

[8] Papadopoulos, Timos; Edwards, David S.; Rowan, Daniel; Allen, Robert. “Identification of auditory cues utilized in human echolocation - objective measurement results”. Inst. of Sound & Vibration Res. (ISVR), Southampton, UK; Information Technology and Applications in Biomedicine, 2009. ITAB 2009. 9th International Conference. ppp 1-4. 4-7 nov 2009.

[9] Ludeman, Lonnie. “Multisignal time difference estimator with application to the sound ranging problem”. New Mexico State University, Las Cruces, New Mexico; Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80. Vol 5. pp 800 – 803. Apr 1980.

[10] Mitchell, Alister J.; Communications for Artillery Location in the British Army 1914 -1970.

[11] Grbic, Nedelko; Dahl, Mattias; Claesson Ingvar. ”Acoustic Echo Cancelling and Noise Suppression with Microphone Arrays”. Department of Telecommunications and Signal Processing. University of Karlskrona/Ronneby. Research report 1999:5, ISSN 1103-1581. 1999.

[12] Ka Fai, Cedric Yiu; Grbic, Nedelko; Kok-Lay, Teo; Nordholm, Sven. “A New Design Method for Broadband Microphone Arrays for Speech Input in Automobiles”. Signal Processing Letters, IEEE; Vol 9. Nº 7. pp 222-224. ISSN 1070-9908. July 2002.

[13] Djendi, Mohamed; Gilloire André; Scarlat Pascal. “Noise Cancellation using Two Closely Spaced Microphones: Experimental Study witha Specific Model and Two Adaptive Algorithms”. University of Rennes. Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference. Vol 3. pp III-III. ISSN 1520-6149. 14-19 May 2006

[14] Maganti, Hari Krishna and Gatica Perez, Daniel. “Speaker Localization for Microphone ArrayBased ASR:The Effects of Accuracy on Overlapping Speech”. IDIAP Research Institute Martigny, Switzerland and University of Ulm, Ulm, Germany. 2006.

[15] McKinney, E. D.; DeBrunner, V. E. “A Two-microphone Adaptive Broadband Array for Hearing Aids. School of Electrical Engineering”, The University of Oklahoma, U.S. Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference. Vol 2. pp 933-936. 7-10 May1996.

[16] Genescà, M.; Romeu J.; Boone M.M . “Evaluación de un Sistema Matricial de Ocho Micrófonos para la Localización de Fuentes Sonoras”. Escuela Técnica Superior de Ingeniería Industrial de Terrassa U.P.C. Terrassa, España and Lab. of Acoustical Imaging and Sound Control,Delft University of Technology, Delft, The Nederlands 2003.

[17] Saxena, Ashutosh; Ng, Andrew Y. “Learning Sound Location from a Single Microphone. Computer Science Department”, Stanford University, Robotics and Automation, 2009. ICRA '09. IEEE International Conference. pp 1737-1742. ISSN 1050-4729. Kobe, Japan. May 12-172009

[18] Rubio, Juan E.; Ishizuka, Kentaro; Sawada, Hiroshi; Araki, Shoko; Nakatani, Tomohiro; Fujimoto, Masakiyo. “Two-Microphone Voice Activity Detection Based on the Homogeneity of the Direction of Arrival Estimates”. NTT Communication Science Laboratories,NTT Corporation. Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference. Vol 4. Pp IV-385 – IV-388. ISSN 1-4244-0727-3. 15-20 April 2007.

[19] Kurihara , Masaki; Ono, Nobutaka; Ando, Shigeru. “Theory and Experiment of Dual Sound Sources Localization with Five Proximate Microphones”. Graduate School of Information Science and Technology, University of Tokyo, Japan. SICE 2002. Proceedings of the 41st SICE Annual Conference. Vol 2. pp 1100 – 1101. Aug 5-7 2002.

[20] Swartling, Mikael; Grbic, Nedelko; Ingvar Claesson. “Direction of Arrival Estimation for Multiple Speaker using Time-Frequency Orthogonal Signal Separation”. Department of Signal Processing, School of Engineering, Blekinge Institute of Technology. Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference. Vol 4. pp IV-IV. ISSN 1520-6149. 14-19 May 2006.

[21] Zhang, Wenyi; Ra, Bhaskar D. “Two Microphone Based Direction of Arrival Estimation for Multiple Speech Sources using Spectral Properties of Speech”. Department of Electrical and Computer Engineering, University of California, San Diego. U.S. Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference. pp 2193 – 2196. ISSN 1520-6149. 19-24 April 2009.

[22] Pollefeys Marc ; Nister David. “Direct Computation of Sound Microphone Locations From Time-Difference of Arrival Data”. ETH Zürich and UNC-Chapel Hill, Department of Computer Science and Microsoft, Live Labs. Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference. pp 2445 – 2448. ISSN 1520 – 6149. March 31 -April 4 2008.

[23] Nagata, Yoshifumi; Fujioka, Toyota; Abe, Masato. “Two-Dimensional DOA Estimation of Sound Sources Based on Weighted Wiener Gain Exploiting Two-Directional Microphones”. Audio, Speech, and Language Processing, IEEE Transactions. Vol 2. Nº 2. pp 416 – 429. Feb 2007.

[24] Kwok ,N. M.; Buchholz, J.; Fang, G.; Gal. J. “Sound Source Localization: Microphone Array Design and Evolutionary Estimation”. MARCS Auditory Laboratory and School of Engineering and Industrial Design University of Western Sydney. Industrial Technology, 2005. ICIT 2005. IEEE International Conference. pp 14 – 17 Dec 2005.Taipei, Taiwan, 14-17 Dec. 2005.

[25] Usman, Muhammad; Keyrouz, Fakheredine and Diepold, Klaus. “Real Time Humanoid Sound Source Localization and Tracking in a Highly Reverberant Environment”. Munich University of Technology, Munich, Germany. Signal Processing, 2008. ICSP 2008. 9th International Conference. pp 2661 – 2664. 26-29 Oct. 2008.

[26] Valin, Jean-Marc; Michaud, François; Rouat, Jean; Létourneau, Dominic . “Robust Sound Source Localization Using a Microphone Array on a Mobile Robot”. LABORIUS - Research Laboratory on Mobile Robotics and Intelligent Systems, Department of Electrical Engineering and Computer Engineering. Université de SherbrookeQuébec, Canada. Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference. Vol 2. pp 1228 – 1233. 27-31 Oct. 2003.

[27] Sathyan, thuraiappah; Humphrey, David; Hedley, Mark; “WASP: A System and Algorithms for Accurate Radio Localization Using Low-Cost Hardware”. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions. Vol. PP. Issue 09. ppp 1-12. ISSN 1094-6977. 1 july 2010.

[28] Sánchez Bote, José Luis. “Micrófonos”, in: E.U.I.T of Telecommunications Publication Department, 1st edition, page number 5.

[29] http://media.musicalplanet.com/pdf/AKG007.PDF

[30] http://www.akg.com/mediendatenbank2/psfile/datei/76/C4174055c23615331.pdf

[31] de Boer, B. “The Evolution of Speech”, in: Brown, K (Ed.) Encyclopedia of Language and Linguistics 2nd edition, Elsevier.2006.

[32] http://etd.lib.fsu.edu/theses/available/etd-04092004-143712/unrestricted/Ch_6lms.pdf

[33] Proakis, John G.; Manolakis, Dimitris G.; “Digital Signal Processing. Principles, Algorithms and Applications”; Third Edition Prentice Hall; pp 448 – 450; Northeastern University / Boston College; 1996