Federal University of Pernambuco

Center for Technology and Geosciences

Department of Electronics and Systems

Bachelor of Science in Electronics Engineering

iOwlT: Sound Geolocalization System

Matheus Sobreira Farias

Davi Carvalho Moreno de Almeida

Senior Thesis

Recife

March, 2021

Federal University of Pernambuco
Center for Technology and Geosciences
Department of Electronics and Systems

Matheus Sobreira Farias
Davi Carvalho Moreno de Almeida

iOwlT: Sound Geolocalization System

A thesis presented to the Faculty of the Department of Electronics and Systems in partial fulfillment of the requirements for the Bachelor of Science in Electronics Engineering degree at Federal University of Pernambuco

Advisor: *Prof. Dr. Daniel de Filgueiras Gomes*
Co-Advisor: *Prof. Dr. Edna Natividade da Silva Barros*

Recife
March, 2021

Para Nega, Boboi e Tete, minhas três mães.

– Matheus

Para Ana e Jonas.

– Davi

Acknowledgements

To my family and my partner Clara, they always support and motivate me. For all love and dedication to provide me the best education I could ever have, my sincerely thanks.

I also would like to thank my advisors, Prof. Daniel de F. Gomes and Prof. Edna Barros, for all the support throughout the development of this amazing project. They showed me that it is possible to create relevant projects with the knowledge I got from college. For all their time spent in and out office hours, and even traveling with me to watch my presentations.

I can not forget my other advisor, Prof. Sergio Rezende, such an inspiring scientist that I admire since high school. It was my pleasure to work in Prof. Rezende's lab with Daniel Souto Maior during my time at college. There I could learn how is it to be part of a team that works carefully to expand the boundaries of science. Thanks for this outstanding opportunity.

To all my colleagues from the Department of Electronics and Systems for the daily support in this journey. My special thanks to Arthur Pimentel, my partner in most of the disciplines in the course. I am grateful to all we faced and overcame along the way together. To Davi Moreno, the one I share some of the most important moments that culminated in this work. I really appreciate your focus and determination. To Gabriel Firmo for the relevant contributions to this work.

I also thank all involved in the projects I was engaged, especially from Diferencial (Ricardo and Vinicius) and iTraffic (Luiz and Levi). The receptiveness from the Department of Mathematics (Ricardo, Felipe, and Thiago), and also from the Department of Physics (Sergio, Flavio, Leonardo, and Daniel). To finish, my gratitude for all members of GMI to the daily motivation.

I am lucky to have all these people supporting me, I would be no one without them. Thank you!

– Matheus

I would like to thank my mother Ana Paula and my father Jonas, for educating me and helping me to face this journey that was the graduation, supporting in good and bad moments. Without them I would not have succeeded.

To my teachers Heleno and Fabiana, who at school were the first to make me interested in science.

To my uncles Severino and Fábio, who were fundamental in my decision to choose this course.

To my colleague Matheus Henrique, who, in addition to being a great friend, was my partner in various disciplines during graduation.

To my colleague Matheus Farias, who through his friendship and many sleepless nights has brought us great achievements, together with my colleague Gabriel Firmo and my professors and advisors Prof. Daniel de F. Gomes and Prof. Edna Barros, fundamental in this work and in my last years of graduation.

To my many colleagues and professors who have contributed to this journey, which was not easy, but with them it was possible to face all obstacles knowing that I was not alone.

Finally, I would like to thank my sister Ana Clara and my brother André, as well as my grandparents Agnelo, Melina, Severino and Berenice, and also all my other family members who helped me along this path.

– Davi

A genius without love and dedication is worse than a student without talent.

—FELIPE COIMBRA

Part of the journey is the end.

—TONY STARK (Avengers: Endgame)

Abstract

Locating targets play an important role in nature. Some species may have specific anatomic characteristics that allow enhancing accuracy when finding their prey. In particular, barn owls have an asymmetric ears disposition that made them outstanding night hunters. On the other hand, this capacity is also relevant to engineering applications: radars, Global Positioning System (GPS) and sonars are great examples. When it comes to Sound Source Localization (SSL) systems, the Time Difference of Arrival (TDOA) estimation between sensors is a very common approach. This method is very dependant on three factors: the geometry of the sensors' disposition, the synchronization between sensors and the sampling rate of the received signals. iOwlT is an intelligent cyber-physical SSL system inspired by owls and focused on locating impulsive sounds, as gunshots. In this work, the system was physically built and trained to recognize and locate some kinds of impulsive sounds. Also, it presents a comprehensive simulation study on optimizing SSL hardware parameters with Particle Swarm Optimization (PSO) algorithm. The system's sound classification, based on neural networks, scored 91.38% accuracy, and the localization algorithm performed 97.21% and 88.32% on determining impulsive sounds direction and position, respectively. The optimization method presented potential directions on building better SSL systems, performing up to 33.0% better than similar problem approaches.

Keywords: cyber-physical systems, sound source localization, particle swarm optimization, biologically-inspired systems, sensor array.

Resumo

A localização de alvos possui um importante papel na natureza. Algumas espécies apresentam características anatômicas específicas as quais garantem uma maior precisão na procura de suas presas. Em particular, as corujas-das-torres têm uma disposição assimétrica de suas orelhas que as tornam excelentes caçadoras noturnas. Concomitantemente, essa capacidade é também relevante em aplicações na engenharia: radares, GPS e sonares são bons exemplos. No tocante a sistemas de localização de fontes sonoras (SSL), métodos de localização por estimação da diferença de tempo de chegada entre os sensores (TDOA) são comumente utilizados. Esses métodos são bastante dependentes de três fatores: a geometria de disposição dos sensores, o número total de sensores e a taxa de amostragem dos sinais recebidos. iOwlT é um sistema ciberfísico inteligente de localização de fontes sonoras inspirado em corujas e focado em localizar sons impulsivos, como tiros de armas de fogo. Neste trabalho, o sistema foi fisicamente desenvolvido e treinado para reconhecer e localizar sons impulsivos. Além disso, apresenta um estudo simulado sobre a otimização de parâmetros de hardware em sistemas SSL com o uso do algoritmo Particle Swarm Optimization (PSO). A classificação do som feita pelo sistema, baseada em redes neurais, obteve 91.38% de acerto, e o algoritmo de localização possuiu uma performance de 97.21% e 88.32% na determinação da direção e da posição de sons impulsivos, respectivamente. O procedimento de otimização apresentou potenciais direções em construções de melhores sistemas SSL, com uma melhoria de até 33.0% em relação a metodologias similares.

Palavras-chave: sistemas ciberfísicos, localização de fontes sonoras, particle swarm optimization, sistemas biologicamente inspirados, matriz de sensores.

Contents

List of Figures

List of Tables

List of Acronyms

CHAPTER 1

Introduction

Hoot, hoot!

—A BARN OWL

1.1 Contextualization and Motivation

Owls are animals that have nocturnal habits. To hunt their prey during the night, they can not trust only in their vision capacity. At night the sight is naturally more overshadowed by the absence of light, especially in forests (their natural habitat). To guarantee their everyday food, owls have to use other benefits of evolution to improve the accuracy of the target location.

Experiments conducted by neurobiologists [1] proved that barn owls – a particular species of owl – are able to locate prey, even being immersed in a totally dark room. They are capable of doing that just by hearing the sound emitted by their prey. In fact, the great evolutionary advantage present in this species is the considerable asymmetric disposition of their ears: the left ear is positioned some centimeters below the right one. Due to this relevant difference, these owls receive the sound information phase-shifted, being possible to accurately determine the target location.

An important fact is that barn owls are not born [2] with the localization technique already well developed. It is necessary some time for the young owl to adapt himself to his own physical characteristics such as skull diameter, height difference between the ears, etc. These attributes can vary significantly even in the same species. Moreover, these owls have channels of rigid feathers on the side of their heads that allow them to regulate the passage of sound. Thus, these animals have a very efficient adaptive control, since their localization method maintains accurate even under different environmental conditions or physical differences inherent to the species.

Translating the situation to an engineering view, the idea is to determine the localization of a sound source given an array of sensors receiving their signals. There are several mathematical methods to solve this so-called localization problem. Some of them are: steered beamforming [3], high-resolution spectral estimation [4, 5], and TDOA estimation [6]. The last is the most similar to what barn owls do.

The TDOA estimation procedure consists of measuring the time difference of arrival between every two sensors, commonly obtained through cross-correlation techniques [7, 8]. With these times, one can build a system of non-linear hyperbolic equations in which the solution is the target location. Considering the cyber-physical system to have exactly four sensors (the minimum necessary to solve the system in a three-dimensional space), there are some approaches to solve the system. For instance, it can be used iterative algorithms based on least-squares [9, 10, 11], non-iterative as those based on Maximum Likelihood Estimation (MLE) [12], stochastic approaches with PSO algorithm [13], and via neural networks [14]. The optimization of the hardware parameters, that admits an arbitrary number of sensors, discussed in Section 5.2.2, used a combination of MLE and Hyperbolic Least Squares (HLS), known as Maximum Likelihood Estimation-Hyperbolic Least Squares (MLE-HLS) [15]. However, in the iOwlT system, we took the advantage of a strategy to solve the hyperbolic system turning it into a linear system, using more than four sensors.

To build a cyber-physical SSL system based on TDOA estimation, there are three important factors to take into account: the geometry of sensor array [16], the sampling rate of received signals [15], and its synchronization. The first is important because depending on the array's symmetry, it may cause less accuracy in determining the localization in some directions due to the loss of degrees of freedom. There are analytical studies discussing good configurations under far-field conditions [17, 18] and also stochastic approaches, optimizing the geometry given an environment [19]. Plato solids, pyramidal, spherical and planar configurations are considered traditional geometries and can be found in many other works [20, 21, 22, 23, 24, 25]. The other factors are important in the accurate determination of the TDOA and also in the good discrete representation of the signals.

In the 21st century, the world, and particularly, Brazil, faces an alarming security problem

related to firearm deaths. According to a survey from the Brazilian Forum of Public Security, among 57,956 homicides in 2018, up to 71.1% were by firearms [26]. Also, the Institute for Health Metrics and Evaluation did comprehensive research from 1990 to 2016 concluding that Brazil is ranked first as the country with most firearm deaths (43,200 in 2016) [27]. This way, SSL systems could be beneficial to improve this situation if trained to recognize gunshots effectively.

Impulsive sound recognition plays an important role in security systems: car crashes and gunshots are good examples of typical impulsive sounds that would be pertinent to detect. After this step, one can think of locating this sound source or even directing a camera to record the event in real-time. The hardest part of this process is to classify which impulsive sound the system is listening. Some common approaches establish an intensity threshold and windowing time to serve as input for a neural network [28]. Also, there are some feature extraction techniques to pre-process the sound data as MFCC before sending it to the neural network.

This work presents a cyber-physical implementation of a SSL system inspired by barn owls to locate some desired impulsive sounds, as gunshots. We have trained the classification part with support from the Brazilian Special Ops (BOPE). The acquisition, synchronization, and communication part of the system was designed with Field Programmable Gate Array (FPGA). Initially, iOwlT was designed to participate in the InnovateFPGA 2019 [29], a world FPGA design contest held in Tianjin, China. There, we earned three international awards, 1) 2nd place out of 40 teams in the Regional Finals (considering North, Central, and South America), 2) 2nd place out of 270 teams in the Grand Finals (over the world), 3) Community Award, elected by the community as the best project (considering North, Central, and South America). The classification algorithm scored up to 91.38% accuracy, and the localization algorithm performed up to 97.21% and 88.32% on determining impulsive sounds direction and position, respectively. After the competition, we continued working on the paper "Optimization of Hardware Parameters on a Real-Time Sound Localization System", proposing a comprehensive study about important parameters and an optimization strategy to find novel configurations. Using PSO algorithm, this method returned parameters that scored up to 33.0% better than similar approaches.

1.2 Objectives

This section presents a list of objectives to be accomplished. First the general objective and then the specific ones.

1.2.1 General Objective

- Implementation of an intelligent cyber-physical sound source localization system inspired by owls and focused on locating impulsive sounds, as gunshots.

1.2.2 Specific Objectives

- Design an acquisition module for receiving the sound signal properly;
- Guarantee confiability in the synchronization between each sensor, avoiding inaccurate measurements;
- Develop a neural network classification module to discriminate the desired sound;
- Implement an efficient low-complexity localization algorithm;
- Elaborate a Bluetooth communication channel between the system and a mobile phone to show the target location;
- Discuss a novel method for evaluating configurations and optimize hardware parameters.

1.3 Organization

This work is divided into 6 chapters. In the end, references are provided.

Chapter 2 – Sound Localization. It contains the theoretical formulation of a SSL. Both the two localization algorithms that were used in this work as well as a comprehensive discussion about the relevant parameters for the problem are presented.

Chapter 3 – Sound Classification. It describes the definition of impulsive sounds and methods for their recognition and classification. From pre-processing to discrimination.

Chapter 4 – Methodology and Development. It presents the whole methodology used for the system's development. From planning to execution, since coding to physical aspects.

Chapter 5 – Results and Discussion. It shows the results the system obtained, from classification to localization, also promotes a comprehensive analysis that motivated "Optimization of Hardware Parameters on a Real-Time Sound Localization System".

Chapter 6 – Conclusion. It exposes final considerations related to the work, and also future potential directions to building novel SSL systems.

Bibliography. The bibliographic references used during the work.

Appendix A – Original Work. The original work, submitted to InnovateFPGA 2019 Design Contest. Authored by Davi Almeida, Gabriel Firmo, and Matheus Farias.

1.4 Author Contributions

These were the author contributions in the original iOwlT: Sound Geolocalization System work submitted to InnovateFPGA 2019 Design Contest:

Matheus Farias. Developed the acquisition module, contributed to the software code (more related to the neural network part), wrote the final submission, presented, and made the live demonstration.

Davi Moreno. Developed the acquisition module, contributed to the FPGA code (more related to the circular buffer, Advanced RISC Machine (ARM)/FPGA communication and Bluetooth communication part), and contributed to the software code (more related to the neural network part).

Gabriel Firmo. Developed the acquisition module, contributed to the FPGA code (more related to the ARM/FPGA communication part), and contributed to the software code (more related to the localization algorithm part).

This senior thesis is a summary written by Matheus and Davi of what was developed in iOwlT: Sound Geolocalization System before and after the competition.

CHAPTER 2

Sound Localization

There is nothing more deceptive than an obvious fact.

— ARTHUR CONAN DOYLE (The Boscombe Valley Mystery)

The problem of SSL is one of the main ones to be faced in this work. In addition to having applications in engineering, it is also a problem found in the animal kingdom. For instance, owls use the asymmetry of their ears, Figure 2.1, to locate and capture prey only through hearing, allowing them to hunt in dark environments or with a lot of snow.

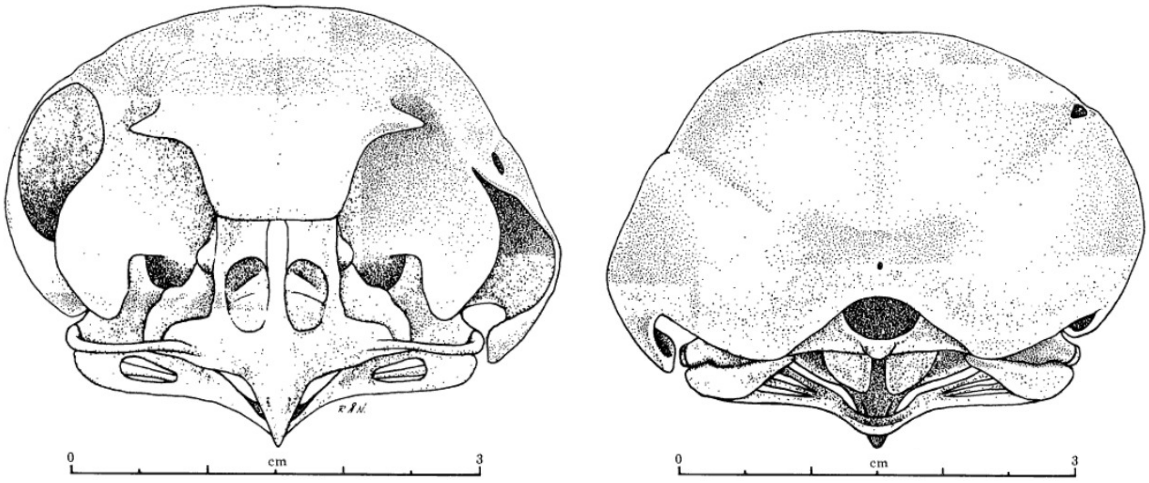
The SSL is a non-linear problem, sensitive to several factors, such as the geometry of the sensors, their synchronization, and the digitization of the sound signal due to the sampling rate. This chapter presents this problem, shows some possible ways to solve it, and introduces some sensitivity factors that affect or help the design of the system.

2.1 Sound Source Localization Problem

Given a geometry of sensors with known positions, it is desired to locate a sound source of unknown position, only with the signal received by each of these sensors, the position of these sensors, and the speed of sound in the environment. This is the SSL problem studied here.

A mathematical formulation of the problem can be obtained directly. Initially, suppose a geometry A of n sensors with positions $P_i = (x_i, y_i, z_i)$, $i = 1, \ldots, n$, where we want to estimate the position of the sound source given by $P_s = (x_s, y_s, z_s)$. The distance between the source and the i-th sensor is given by

$$D_{is} = \|P_i - P_s\|_2 = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2}. \tag{2.1}$$



(a) (b)

Figure 2.1 Visualization of the asymmetry of a barn owl's ears a) front and b) back.

Also, knowing that v is the speed of sound in the environment, we can write:

$$D_{is} = \|P_i - P_s\|_2 = t_{is} v, \tag{2.2}$$

where t_{is} is the Time of Arrival (TOA) of the sound emitted by the source on the i-th sensor. These TOAs cannot be calculated directly, as the instant at which the sound was emitted by the source is not known in advance. Thus, it is necessary to use the relative distances between the sensors, that is, the distance that the sound travels to the i-th sensor after it has already been received by the j-th sensor. This distance is called D_{ij}, and is given by

$$D_{ij} = D_{is} - D_{js} = \underbrace{(t_{is} - t_{js})}_{t_{ij}} v, \tag{2.3}$$

where t_{ij} is the TDOA of the sound between the sensors i and j. The TDOA for each pair of sensors can be estimated from the signals received by the sensors. For this, one can use, for example, cross-correlation techniques.

Considering $i = 1$ as the reference sensor, the mathematical description of the problem is obtained by the set of $n-1$ equations given by

$$D_{1j} = t_{1j} \cdot v, \quad j = 2, \ldots, n. \tag{2.4}$$

For these equations, only the values of P_s are unknown, as P_i and v are known and t_{ij} can be estimated. This problem is also called the hyperbolic positioning problem since it is defined by this set of hyperbolic equations.

As an example, suppose the two-dimensional case, in which there is a geometry composed of 3 sensors with positions given by

$$\begin{aligned} P_1 &= (0,0), \\ P_2 &= (1,0), \\ P_3 &= (0,1). \end{aligned} \tag{2.5}$$

For a source located at $P_s = (4,4)$, the hyperbolic system will be defined from (2.4) as

$$\begin{aligned} \sqrt{x_s^2 + y_s^2} - \sqrt{(x_2 - x_s)^2 + y_s^2} &= t_{12} \cdot v, \\ \sqrt{x_s^2 + y_s^2} - \sqrt{x_s^2 + (y_3 - y_s)^2} &= t_{13} \cdot v. \end{aligned} \tag{2.6}$$

Calculating the values of $(t_{1j} \cdot v)$ directly from (2.3) and (2.1), the two hyperbolas that describe the system are obtained. A view of this case is shown in Figure 2.2. Note that the hyperbolas intersect at the source's location point P_s.

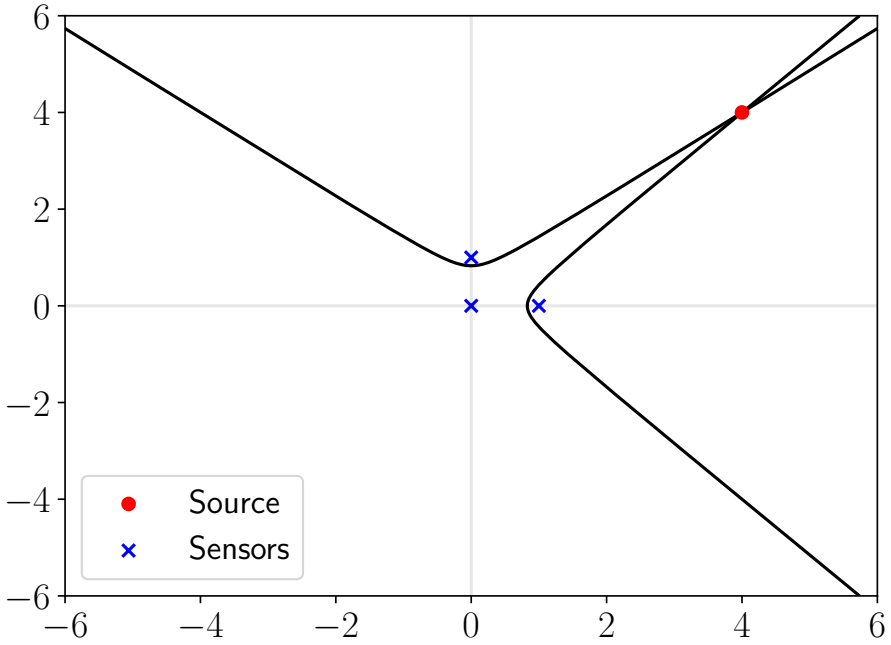


Figure 2.2 SSL example in 2D.

2.1.1 Algorithm for Limited Resources

In the case that the system developed to solve the problem of SSL is computationally limited, there is an interesting approach that can be followed.

From (2.4), one can write

$$D_{js} = D_{1s} - v \cdot t_{1j}. \tag{2.7}$$

After squaring on both sides, rearranging the terms, and dividing by $v \cdot t_{1j}$, we have

$$0 = v \cdot t_{1j} - 2D_{1s} + \frac{D_{1s}^2 - D_{js}^2}{v \cdot t_{1j}}. \tag{2.8}$$

In order to remove the term D_{1s}, we can subtract the equation (2.8) from the case that $j = 2$, so that

$$0 = v(t_{1j} - t_{12}) + \frac{D_{1s}^2 - D_{js}^2}{v \cdot t_{1j}} - \frac{D_{1s}^2 - D_{2s}^2}{v \cdot t_{12}}. \tag{2.9}$$

Establishing the origin of the coordinate system in $P_1 = (0,0,0)$, we have from (2.1) that

$$D_{1s}^2 - D_{js}^2 = -x_j^2 - y_j^2 - z_j^2 + 2x_j x_s + 2y_j y_s + 2z_j z_s. \tag{2.10}$$

The combination of the last two equations provides a set of $N-2$ nonhomogeneous linear equations, given by

$$\begin{aligned}
0 &= x_s A_j + y_s B_j + z_s C_j + D_j, \\
A_j &= \frac{2x_j}{v \cdot t_{1j}} - \frac{2x_2}{v \cdot t_{12}}, \\
B_j &= \frac{2y_j}{v \cdot t_{1j}} - \frac{2y_2}{v \cdot t_{12}}, \\
C_j &= \frac{2z_j}{v \cdot t_{1j}} - \frac{2z_2}{v \cdot t_{12}}, \\
D_j &= v(t_{1j} - t_{12}) - \frac{x_j^2 + y_j^2 + z_j^2}{v \cdot t_{1j}} + \frac{x_2^2 + y_2^2 + z_2^2}{v \cdot t_{12}},
\end{aligned} \tag{2.11}$$

where $j = 3, \ldots, n$.

This system of equations can be solved by a variety of methods, such as Gaussian elimination or singular value decomposition.

2.1.2 Algorithm Based on Maximum Likelihood Estimation

Another way to estimate the location of the source is through the algorithm presented in [12]. This article develops an efficient and non-iterative way to solve the problem of hyperbolic localization, which is the problem described so far. The algorithm performs a MLE, following a different approach.

First, after squaring both sides of (2.1), it is possible to write

$$\begin{aligned} D_{is}^2 &= K_i - 2x_i x_s - 2y_i y_s - 2z_i z_s + x_s^2 + y_s^2 + z_s^2, \\ K_i &= x_i^2 + y_i^2 + z_i^2. \end{aligned} \tag{2.12}$$

Using (2.3) and fixing sensor 1 as the reference leads to

$$D_{is} = D_{i1} + D_{1s}. \tag{2.13}$$

Using this last equation in (2.12) and subtracting the result by D_{1s}^2, obtained from (2.12) with $i = 1$, leads to

$$D_{i1}^2 + 2D_{i1}D_{1s} = K_i - K_1 - 2x_s(x_i - x_1) - 2y_s(y_i - y_1) - 2z_s(z_i - z_1). \tag{2.14}$$

This expression defines a set of $n-1$ equations that can be expressed in the matrix form shown in (2.15), where $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$ and $z_{ij} = z_i - z_j$. Together with (2.16), obtained from (2.12) with $i = 1$, this system allows to estimate the source coordinates.

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = -\begin{bmatrix} x_{21} & y_{21} & z_{21} \\ \vdots & \vdots & \vdots \\ x_{n1} & y_{n1} & z_{n1} \end{bmatrix}^{-1} \times \left\{ \begin{bmatrix} D_{21} \\ \vdots \\ D_{n1} \end{bmatrix} D_{1s} + \frac{1}{2} \begin{bmatrix} D_{21}^2 - K_2 + K_1 \\ \vdots \\ D_{n1}^2 - K_n + K_1 \end{bmatrix} \right\} \tag{2.15}$$

$$D_{1s}^2 = K_1 - 2x_1 x_s - 2y_1 y_s - 2z_1 z_s + x_s^2 + y_s^2 + z_s^2 \tag{2.16}$$

To obtain the final results, we first need to replace x_s, y_s, z_s of (2.15) in (2.16), so that the quadratic equation in D_{1s} will return 2 roots. For each of these roots applied in (2.15), a solution is obtained, so that this algorithm returns two estimates of the source's position.

To solve the problem of having two estimates of the source's position, a cost function based on the method of HLS is used, so that the estimate that has the lowest cost is maintained. This cost function is given by

$$J_{\mathrm{HLS}}(\hat{x}_s, \hat{y}_s, \hat{z}_s) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\hat{D}_{ij} - t_{ij} v)^2, \tag{2.17}$$

where $\hat{D}_{ij}$ is the actual value calculated from (2.3) and (2.1), assuming that the source position is the estimated position $(\hat{x}_s, \hat{y}_s, \hat{z}_s)$.

The described method to determine the best candidate for the source position was presented by [15], where this algorithm is called MLE-HLS and is also compared with other approaches to solve the problem of hyperbolic localization. As it presents high performance, is non-iterative, and has the computational cost of solving a simple linear system and finding roots for a second-degree polynomial, this algorithm also proves to be a good choice for computationally limited systems.

2.2 Geometry

One of the crucial factors in determining the source position is the chosen sensor geometry. To illustrate the importance of a good choice of geometry, some particular cases will be analyzed and evaluated according to the MLE-HLS location algorithm.

Suppose a two-dimensional case that three sensors are collinear and equally spaced. The positions of these sensors are

$$\begin{aligned} P_1 &= (0,0), \\ P_2 &= (1,0), \\ P_3 &= (2,0). \end{aligned} \tag{2.18}$$

From (2.15), the matrix that is composed of the positions of the sensors in relation to the reference can be constructed as

$$M = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}. \tag{2.19}$$

Note that this matrix does not have an inverse, and its pseudoinverse is given by

$$M^{+} = \begin{bmatrix} 0.2 & 0.4 \\ 0 & 0 \end{bmatrix}. \tag{2.20}$$

The fact that M^{+} has the second line with zeros shows that regardless of the position of the source, the algorithm will always return the y_s coordinate as 0. This result can be generalized by saying that the information that sensors in line can obtain in the normal direction to the line is compromised.

Another issue to be observed for this same sensor configuration is the particularity of the TDOA for sources positioned on the x-axis after P_3, that is, $x > x_3 = 2$. For these source positions, the TDOA will always be the same, since the relative distance traveled by the sound between the sensors does not change. In other words, D_{21} and D_{31} will always have the same value. A similar reasoning can be made for sources located on the x-axis but before P_1, that is, $x < x_1 = 0$.

This reasoning can be generalized by saying that sources located in the same direction as the sensors but beyond the line segment that connects them cannot be well located because they have the same TDOA (there may be only a signal difference between the sources on the left and on the right of the sensors).

A three-dimensional case to be considered is that of a sensor geometry that is a square, with positions given by

$$\begin{aligned} P_1 &= (1,1,0), \\ P_2 &= (1,-1,0), \\ P_3 &= (-1,1,0), \\ P_4 &= (-1,-1,0). \end{aligned} \tag{2.21}$$

Also from (2.15), the matrix that is composed of the positions of the sensors in relation to the reference can be constructed as

$$M = \begin{bmatrix} 0 & -2 & 0 \\ -2 & 0 & 0 \\ -2 & -2 & 0 \end{bmatrix} \tag{2.22}$$

This matrix is singular, but its pseudoinverse is given by:

$$M^{+} = \begin{bmatrix} \frac{1}{6} & -\frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{3} & \frac{1}{6} & -\frac{1}{6} \\ 0 & 0 & 0 \end{bmatrix} \tag{2.23}$$

In a similar way to the case of inline sensors, there is no information in the position z_s, which results from the last line of zeros in M^{+}. Another way of visualizing this deficiency is from the central axis of the square, given by $x = y = 0$. It also presents problems in the location of the sources that are in it, since for all points on that axis the distance traveled by the sound will always be the same for all sensors as well as the TDOAs.

From these simple examples, it is possible to illustrate the importance of choosing geometries in the problem of SSL. Often symmetrical geometries can help to detect the direction from which the sound came, but care must be taken with some axes of symmetry to ensure that the system will behave well for sources located in the space of interest.

2.3 Acquisition of Sound Data

In order to be able to perform the desired processing on the audible signal and obtain the TDOA information accurately, it is necessary to be concerned with two procedures, the first is the conversion of the analog signal to digital, and the second deals with the synchronization between the sensors. When converting the analog signal to a digital form, care must be taken with some factors, such as the acquisition circuit and the limitations of the converter used. Synchronism, on the other hand, is crucial to ensure that signals are received simultaneously so that TDOA calculations are accurate.

2.3.1 Acquisition Circuit

Before one can convert the sound signal to the digital domain, we must ensure that it is supplied to the A/D Converter circuit in the correct way, which is done through an acquisition

circuit. The role of this circuit is in

- Ensure that the microphones are correctly polarized;
- Perform an amplification on the received signal;
- Possibly perform filtering of unwanted noise components;
- Supply the signal to the A/D Converter within the allowable voltage range;
- Protect the A/D Converter and the voltage source through protective circuits.

2.3.2 A/D Converter

With the signal in the appropriate voltage range for the A/D Converter, the conversion operation for the digital domain is performed. This converter is capable of discretizing the analog signal to predetermined discrete voltage levels. For instance, for a converter with a voltage range of 0 to 5 V and a resolution of 12 bits, there are $2^{12} = 4096$ levels of quantization, and the voltage resolution is given by

$$\text{Voltage Resolution} = \frac{\text{Voltage Range}}{\text{Quantization Levels - 1}} = \frac{5-0}{4096-1} \approx 1.22 \text{ mV}. \tag{2.24}$$

In addition to the bit resolution and voltage resolution, which allow defining the quantization errors associated with the conversion, some other parameters of this converter are instrumental to impose certain limitations on the design of the SSL system, these are

- The number of channels C of the A/D Converter;
- The maximum number of samples collected per time interval, $f_{s_{\max}}$, of the A/D Converter.

The number of channels C limits the number of microphones that the SSL system can have, limiting the geometry associated with the system. The value of C together with $f_{s_{\max}}$ allows us to calculate the maximum sample rate per channel, which is given by

$$\overline{f_{s_{\max}}} = \frac{f_{s_{\max}}}{C}. \tag{2.25}$$

2.3.3 Circular Buffer

Errors in the synchronization of the sensors insert errors in the calculation of the values of TDOA, which affects the accuracy of the system's location. One way to guarantee synchronism is to use high-performance computing devices, such as FPGA, which together with circular buffer structures establish a well-defined acquisition scheme.

Circular buffers are data structures with the main characteristic of having a fixed size, in which, after filling the last position, the next position to be filled is the initial position. This way, data filling is done in a circular way, and the data structure can be interpreted as a ring.

To illustrate this procedure, consider the Figure 2.3, in which there are two pointers, the begin and the end, which point to the oldest and newest sample inserted in the buffer, respectively. Initially, the pointers are in the same position, and as samples of the audio signal are collected, the end pointer will move one position. When new data arrives and the next position to be filled is the position of the beginning pointer, this position is rewritten and the pointers move one position each.

The use of one independently circular buffer per sensor facilitates the simultaneous acquisition of sound data. The moment a certain stopping criterion is satisfied, the buffers stop receiving new data, and the vectors (starting from the beginning pointer to the end pointer) are provided to the next processing unit.

2.4 Cross-correlation

With the data from the sound source collected, it is necessary to somehow estimate the values of TDOA between the reference sensor and the other sensors. One way to perform such calculations is to use cross-correlation techniques.

For the sensors i and j, the real discrete signals received by each sensor are represented as s_i and s_j, respectively. These signals are of length L, and the correlation is defined as

$$s_i[n] \star s_j[n] \triangleq \sum_{m=0}^{L-1} s_i[m] s_j[m+n]. \tag{2.26}$$

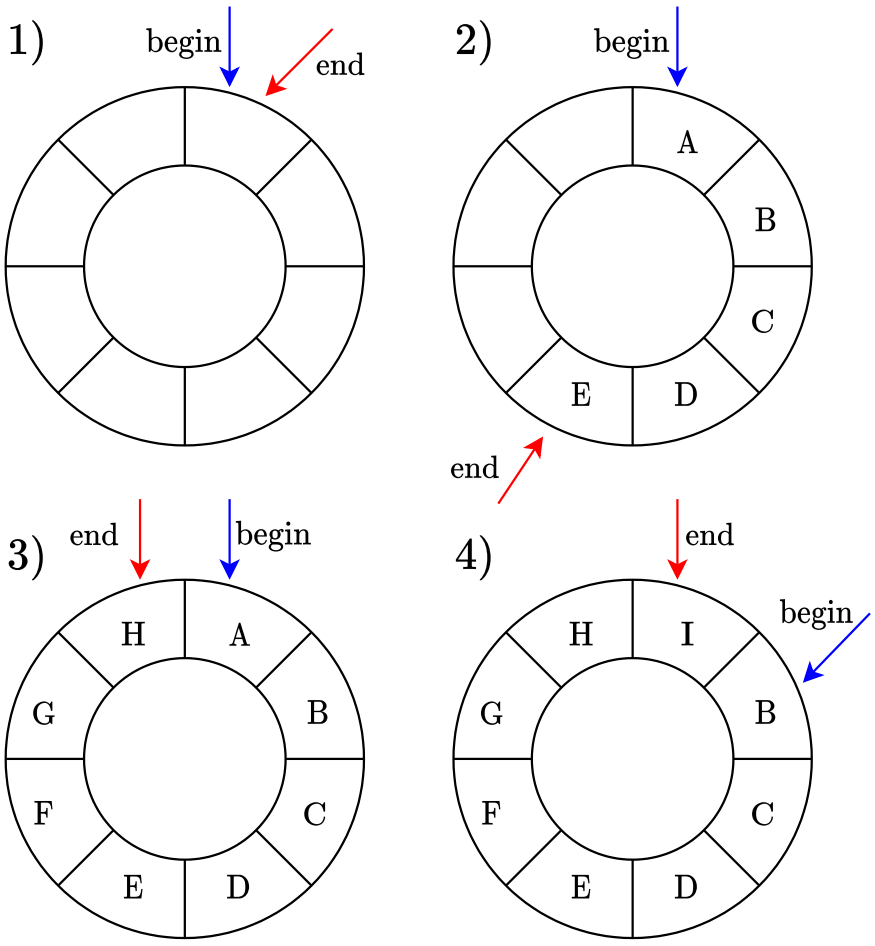


Figure 2.3 Routine of filling a circular buffer.

The TDOA between these signals, t_{ij}, is obtained from the distance from the maximum correlation point to the center of the correlation vector (which is $2L-1$ in size), so that

$$t_{ij} = \frac{\arg\max(s_i[n] \star s_j[n]) - L}{f_s}, \tag{2.27}$$

where f_s is the sample rate of the system. It can be seen then that t_{ij} can only assume values that are multiples of the sampling period $T = 1/f_s$, something that will be explored in the next section.

If implemented as described in (2.26), the complexity of each correlation will be $O(L^2)$. In order to avoid this high computational complexity and take advantage of fast algorithms, the correlation must be performed through Fast Fourier Transform (FFT) algorithms, so that the complexity drops to $O(L\log L)$.

2.5 Sampling Rate

The sampling rate also needs to be well determined, as it influences both the representation of the signal you want to locate and also introduces errors in the location of the sound source. The sensitivity introduced in the estimation of the location of the sound source will be illustrated here.

Suppose t_{ij} is the real TDOA between the i-th and j-th sensors, so that this value is not limited by the discretization performed over time by the sampling rate. After the discretization in period intervals $T = 1/f_s$, the time values available by the system are multiples of T, given by nT. Evidently, the value of t_{ij} will be found between two consecutive values of n, so that if we call that value n', we have

$$n'T \leq t_{ij} \leq (n'+1)T. \tag{2.28}$$

Therefore, the value t'_{ij} that will be chosen by the system must be associated with the value $n'T$ or $(n'+1)T$, which ideally will assume the value closest to t_{ij} . Consequently, the biggest possible error associated with TDOA, in ideal case, will occur when t_{ij} is in the middle of the range in (2.28) (with a value of $(n'+0.5)T$). For this value, t'_{ij} has an equal chance of assuming

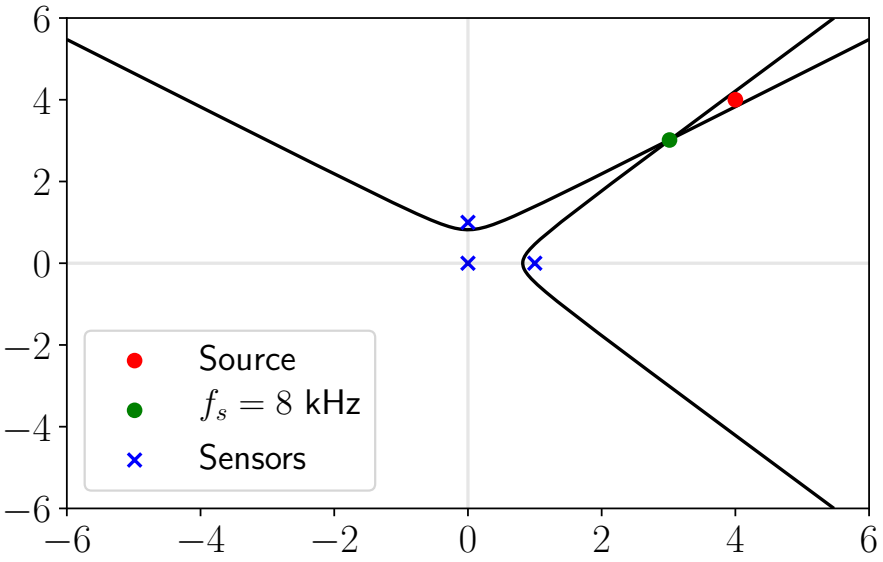


Figure 2.4 SSL example with $f_s = 8$ kHz.

the values $n'T$ and $(n'+1)T$, both having the same error, so that if the first value is chosen the associated error e_t will be given by

$$\begin{aligned} e_t &= |t'_{ij} - t_{ij}| = |n'T - (n'+0.5)T| = \frac{T}{2}, \\ e_t &= \frac{1}{2f_s}. \end{aligned} \tag{2.29}$$

This is the error associated with time, but a more interesting estimate can be found when calculating the distance error e_d associated with this maximum error. This value is simply given by the value e_t multiplied by the speed of sound v, so that

$$e_d = e_t \cdot v = \frac{v}{2f_s}. \tag{2.30}$$

For the values of f_s equal to 8 kHz and 16 kHz, the values of e_d are 2.12 cm and 1.06 cm, respectively. The localization errors that such values can insert into the system are illustrated in Figures 2.4 and 2.5, in which the same array geometry and source position shown in Figure 2.2 were used.

The higher the sampling rate, the smaller the e_t error associated with TDOA approximations, so that the measurements would be more accurate and the system would behave better due to

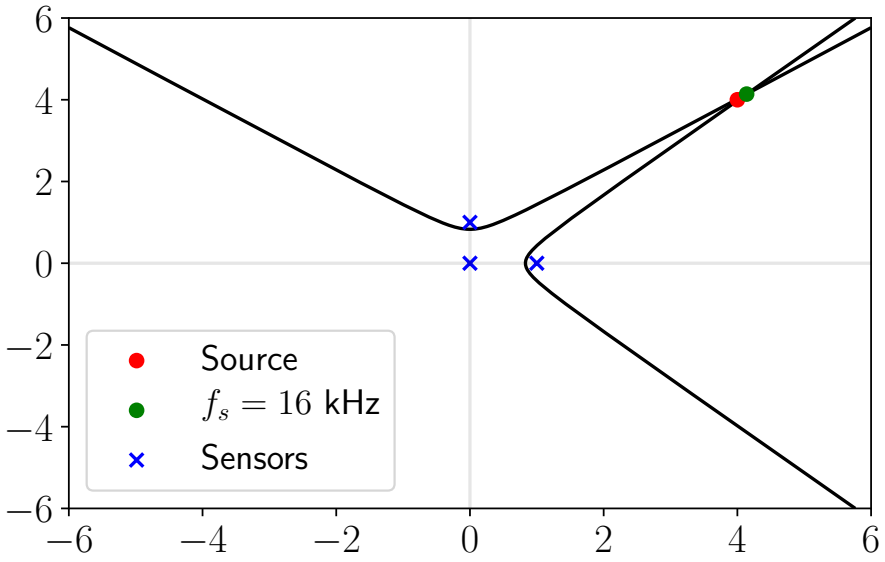


Figure 2.5 SSL example with $f_s = 16$ kHz.

this smaller source of error. Of course, increasing the sampling rate will increase the processing time required by the system, as a larger number of samples will be obtained.

2.5.1 Considerations about the distance between the sensors

It is also interesting to note that for two sensors at a distance of d, the TDOA between them will assume a maximum value of d/v. This maximum value will be associated with an estimated TDOA given by $n'_{\text{max}}T$, so that a decrease in T (increase in the sample rate) will result in a n'_{max} increase, that is, the range of samples that deal with the delay between signals increases, indicating an superior time resolution range available by the system, as a consequence of the higher sampling rate .

In a similar way, a decrease in the distance between the sensors will result in a lower value of maximum TDOA between them, which decreases the range of temporal resolution of the system, since the value of n'_{max} will be smaller. Thus, it is noticeable that if a shorter distance between sensors is desired, one of the factors to be considered is the increase in the sampling rate, so that the system does not lose so much in terms of the time resolution range promoted by

the value $n'_{\max}$.

CHAPTER 3

Sound Classification

Creativity is intelligence having fun.

— ALBERT EINSTEIN

Sound classification is one major part of this work. To determine the SSL, you have to first identify aspects of this sound and judge if the signal received is the desired one. So, it is important to have a robust acquisition method in order to have an accurate signal representation. This acquisition part relies on two important physical devices in the implementation: the sensor positioning, and the A/D Converter. The sensor positioning is represented by a set of microphones, which serve as transductors, converting the sound vibrations to electric signals. As these signals are analogical, we have to digitize them to apply digital signal processing techniques as shown in Figure 3.1. A/D Converter is the hardware responsible for this important conversion.

3.1 Impulsive Signal

In particular, our system is designed to process impulsive signals. Based on [30], the ideal impulsive signal, known as Dirac's delta function, or impulse, is defined "formally" as a function δ that

$$\int_a^b f(t)\delta(t)dt = f(0), \tag{3.1}$$

provided $a < 0$, $b > 0$, and f is continuous at $t = 0$. The main idea in this definition, is that δ impacts over a very small interval, such that $f(t) \approx f(0)$.

Given that, there are some important properties to be mentioned:

- $\delta(t) = 0$ for $t \neq 0$;

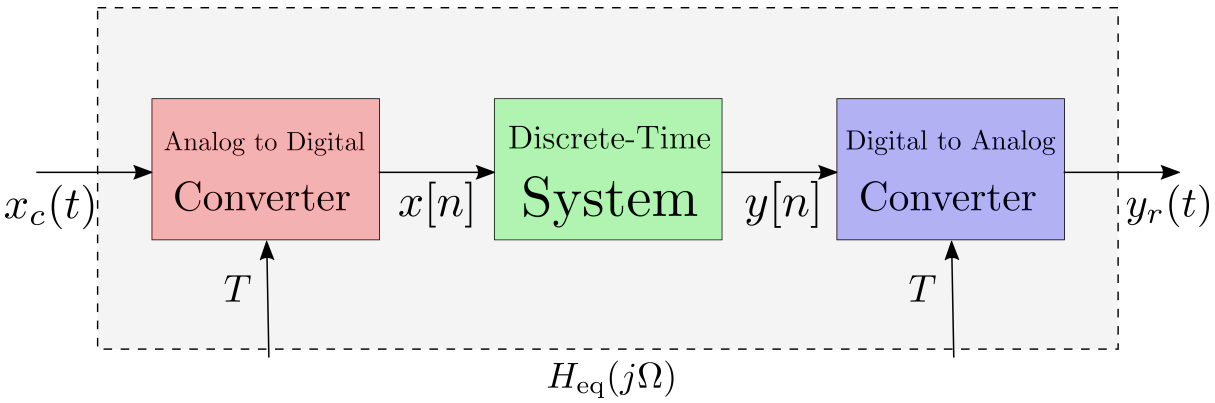


Figure 3.1 General digital signal processing procedure. The input $x_c(t)$ represents the analog (continuous) signal. After the conversion with sampling period T, the input is digitized to $x[n]$ prepared to be processed. Then, the discrete-time system returns the processed array $y[n]$. Some applications may need a last conversion to output an analog signal, $y_r(t)$. The entire procedure can be compared to an equivalent continuous system with impulse response $H_{\text{eq}}(j\Omega)$.

- We define, roughly, that $\delta(0) = \infty$, or simply as not defined;
- $\int_a^b \delta(t)dt = 1$, if $a < 0$ and $b > 0$;
- $\int_a^b \delta(t)dt = 0$, if $a > 0$ or $b < 0$.

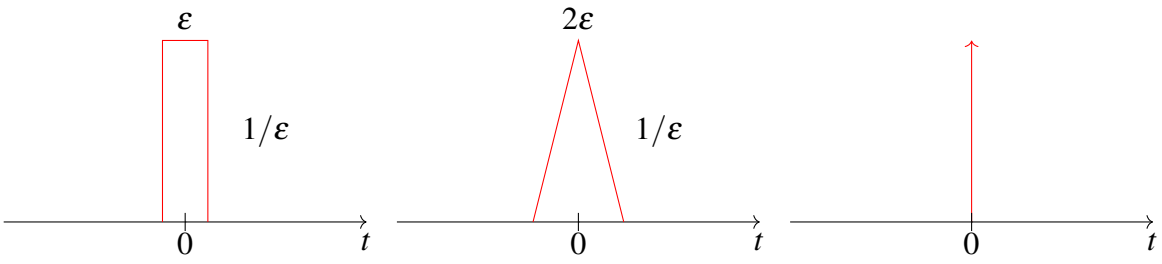
Further discussions and properties related to Dirac's function can be found in [31].

Intuitively, the impulse function is an idealization of a signal that

- is very large near $t = 0$;
- is very small away from $t = 0$;
- has integral equals to 1.

For instance, some examples of signals that follow these three rules are shown in Figure 3.2, as well as the Dirac's delta idealization.

In real-world applications, car crashes and gunshots are good examples of impulsive signals. One important note is that the definition is more flexible here. The environment is commonly noisy, so the amplitude of the signals is not zero at any time. Some approaches may need filtering



(a) Rectangular-shaped (b) Triangular-shaped (c) Dirac's delta

Figure 3.2 Plot of some impulsive signals and the Dirac's delta idealization in the limit $\varepsilon \to \infty$ on frequency domain.

the noise as a first part of the processing. In our case, the signal coming from the A/D Converter has an offset DC component, and impulsive signals, such as gunshot sounds, are distinguished in their high frequency components, so we have to design a properly high-pass filter.

Also, the minimum amplitude to define whether a signal is sufficiently large, and thus, impulsive, can be determined depending on the environment. Urban areas are noisier than rural ones, so it suggests the first to have a higher threshold than the second. Adaptive methods to discriminate thresholds for each environment are recommended.

3.2 High-Pass Filter

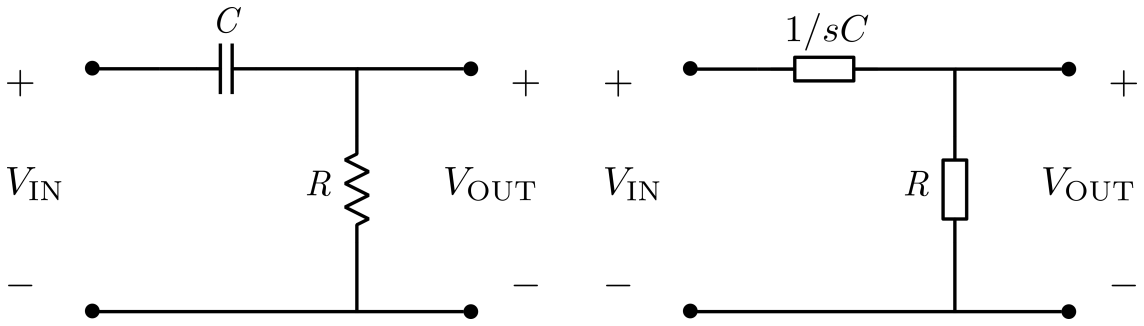
As we are dealing with digital signal processing, we have to implement a digital high-pass filter. This can be done by looking at an analog filter and doing some equivalences.

The simplest analog high-pass filter is the passive first order, shown in Figure 3.3a. By its equivalent Laplace transform circuit in Figure 3.3b one can show that its transfer function $H_{\text{HP}}(s)$ is

$$H_{\text{HP}}(s) = \frac{V_{\text{OUT}}(s)}{V_{\text{IN}}(s)} = \frac{sRC}{1+sRC}. \tag{3.2}$$

Note that $H_{\text{HP}}(s \to \infty) = 1$, and $H_{\text{HP}}(s \to 0) = 0$, which characterizes a high-pass filter.

To clearly see the transformation from an analog to a digital filter, one can write Kirchhoff's



(a) Time domain (b) Frequency domain

Figure 3.3 Circuit diagram of a first-order passive high-pass filter

Laws and the definition of capacitance from the Figure 3.3a

$$\begin{aligned} V_{\mathrm{OUT}}(t) &= RI(t), \\ Q_C(t) &= C(V_{\mathrm{IN}}(t) - V_{\mathrm{OUT}}(t)), \\ I(t) &= \frac{\mathrm{d}Q_C}{\mathrm{d}t}, \end{aligned} \tag{3.3}$$

where $Q_C(t)$ and $I(t)$ are the charge stored in the capacitor and the loop current at time t. Rearranging the equations in (3.3):

$$V_{\mathrm{OUT}}(t) = RC\left(\frac{\mathrm{d}V_{\mathrm{IN}}(t)}{\mathrm{d}t} - \frac{\mathrm{d}V_{\mathrm{OUT}}(t)}{\mathrm{d}t}\right). \tag{3.4}$$

The Equation (3.4) can be discretized by assuming a sampling period T. The input and output samples are now $x[n]$ and $y[n]$ respectively. This way,

$$y[n] = RC\left(\frac{x[n] - x[n-1]}{T} - \frac{y[n] - y[n-1]}{T}\right), \tag{3.5}$$

and now we have the difference equation (3.6)

$$y[n] = \alpha y[n-1] + \alpha(x[n] - x[n-1]). \tag{3.6}$$

The parameter $\alpha = \frac{RC}{RC+T}$ is related to the cutoff frequency f_c by the Equation (3.7)

$$f_c = \frac{1-\alpha}{2\pi\alpha T}. \tag{3.7}$$

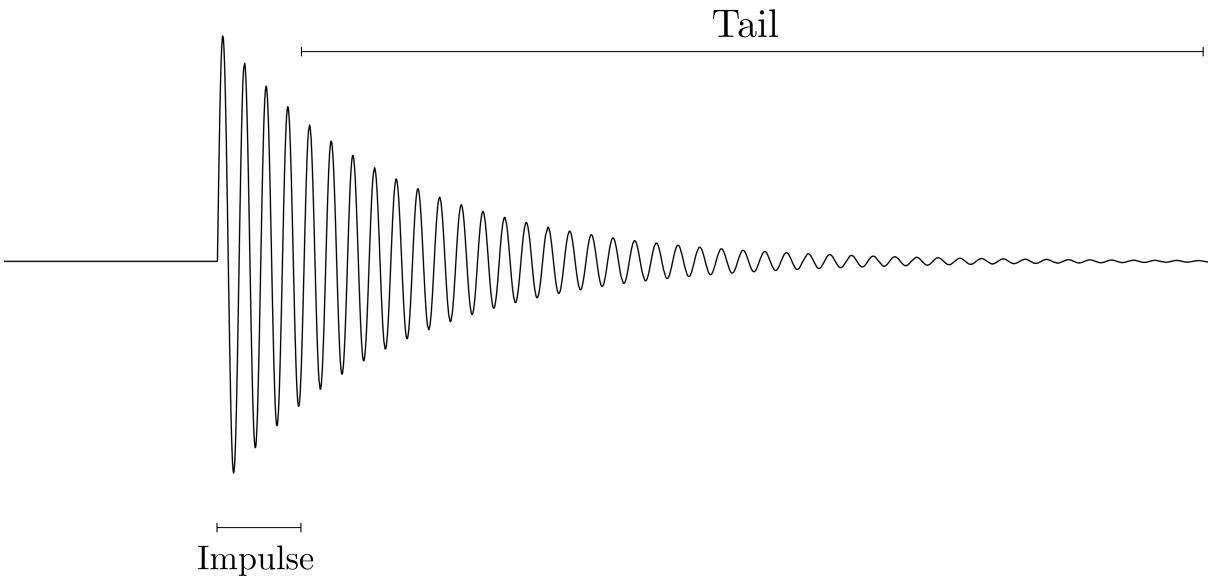


Figure 3.4 A general impulsive signal representation on time domain and its definitions.

3.3 Windowing

To classify the signal, it is important to define a fixed length to the input. In speech recognition, this definition may be challenging, once a simple conversation has different pauses and lengths depending on how expressive the emitter is. As this problem is to classify impulsive signals, a common approach is to find where the amplitude exceeds a threshold and clipping it by a fixed window. This window starts a bit before the impulse and finishes somewhere on its "tail". See in Figure 3.4 a representation of an impulsive signal on time domain and these definitions.

For instance, a signal sampled with a sampling rate of 4 kHz will have 4,000 samples if the window is 1 s. Typically, this window is chosen in the range of 0.1 to 0.5 s, which gives a range of 400 to 2,000 samples in this example. The higher the number of samples, the higher will be the digitized representation, but also the harder will be to build a feasible classifier. It is worth mentioning that the sampling rate must be at least twice the highest frequency component of the sound. Due to the Nyquist-Shannon sampling theorem, this is the boundary a sampling can have to perfectly recover the original signal in ideal conditions.

3.4 Feature Extraction

In the context of classification, we always want to maximize the information we have about the input but also limiting it to a feasible amount of data. Some feature extraction techniques were developed to provide a well-suitable balance in this aspect, especially in sound applications. The MFCC is one popular for music and speech recognition.

The major idea of MFCC is to extract the most important characteristics that can distinguish the sound. The process of calculation involves five components: 1) frequency windowing 2) take the Discrete Fourier Transform (DFT), 3) mel-scale the frequencies over triangular overlapping windows, 4) take the logarithm at each of the mel frequencies, and finally 5) take the Discrete Cosine Transform (DCT). Each step is described in the Figure 3.5

3.5 Neural Network

After all the preprocessing described in the previous sections, the final step is the classification algorithm. In this work, we used a neural network.

Neural network (Figure 3.6) is a relevant artificial intelligence algorithm based on how the brain learns tasks. Its architecture is composed of an input layer, an output layer, and some hidden layers. Each of these layers has a number of elements where some mathematical computations are done, called neurons, and interconnections that transmit these results across layers, the axons. The process of learning is called supervised. We first train the network with inputs and outputs already known, and after trained, the outputs will now be calculated according to the final iteration of a matrix of weights. The learning method follows two steps: forward propagation and backpropagation.

The forward propagation, as the name suggests, is the many calculations that are propagated from the input layer towards the output layer. Each neuron has weight and bias elements, defining weight w and bias b vectors per layer. In this propagation, each layer computes an inner product of w and an input vector x (which is the output of the previous layer) and adds b to this

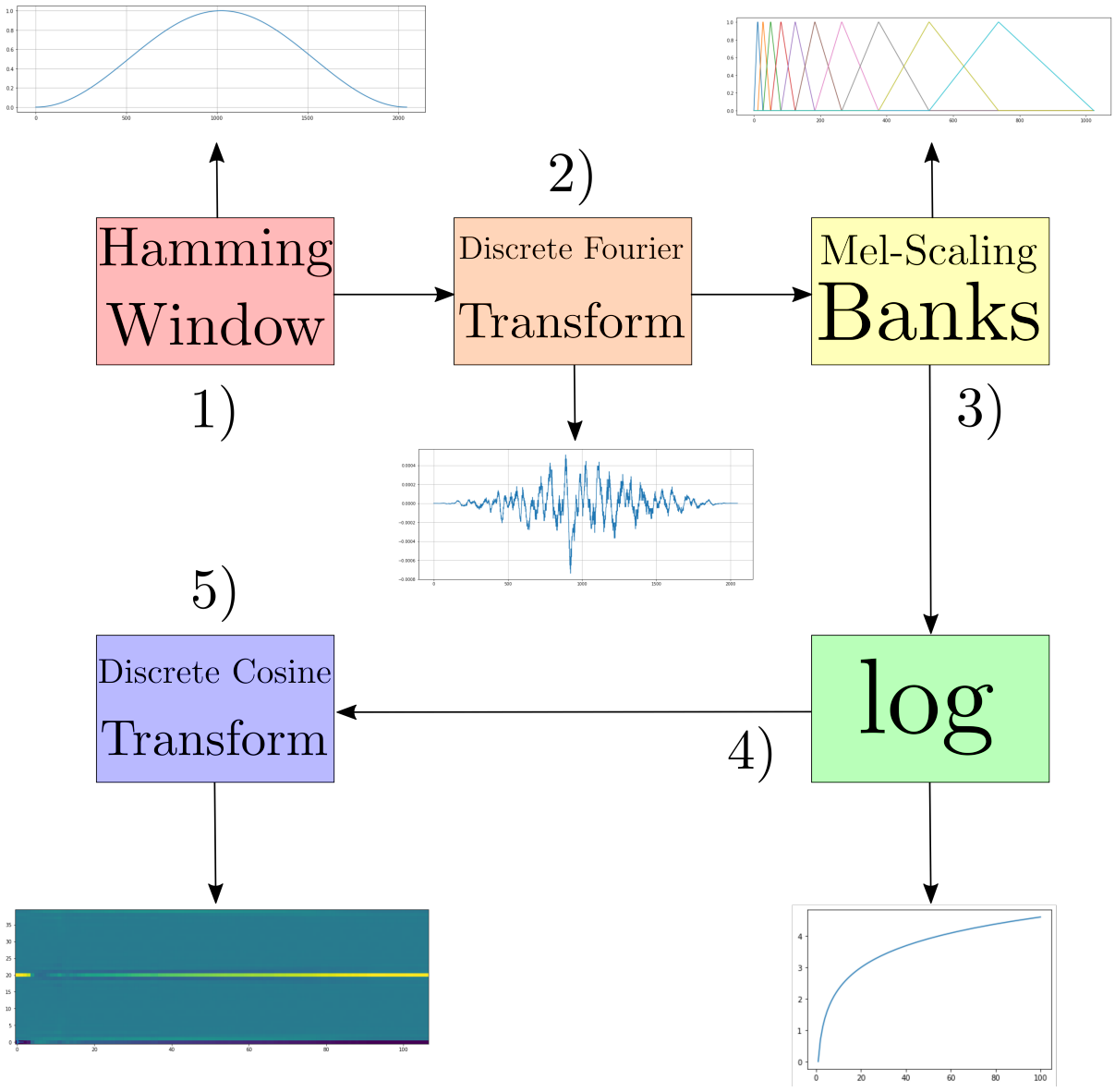


Figure 3.5 MFCC procedure. 1) The signal is multiplied by a Hamming window. This window is different to the one described in Section 3.3. When converting a finite signal on time domain, its frequency response is infinite and vice-versa. This window forces, in a tricky way, our frequency response to be finite. 2) We take the DFT to convert the signal to the frequency domain. 3) We apply triangular filters on a mel-scale to extract frequency bands. It spreads the higher frequencies and shortens the lower ones. 4) We take the logarithm of the mel frequencies. It sounds useless in a first view, but it transforms multiplication into addition, part of the computation of the cepstrum. 5) Finally, we could take the inverse Fourier transform, but it is better to take the DCT. It returns only the real components (the imaginary ones are not important) and also decorrelates the highly correlated coefficients outputted by the banks.

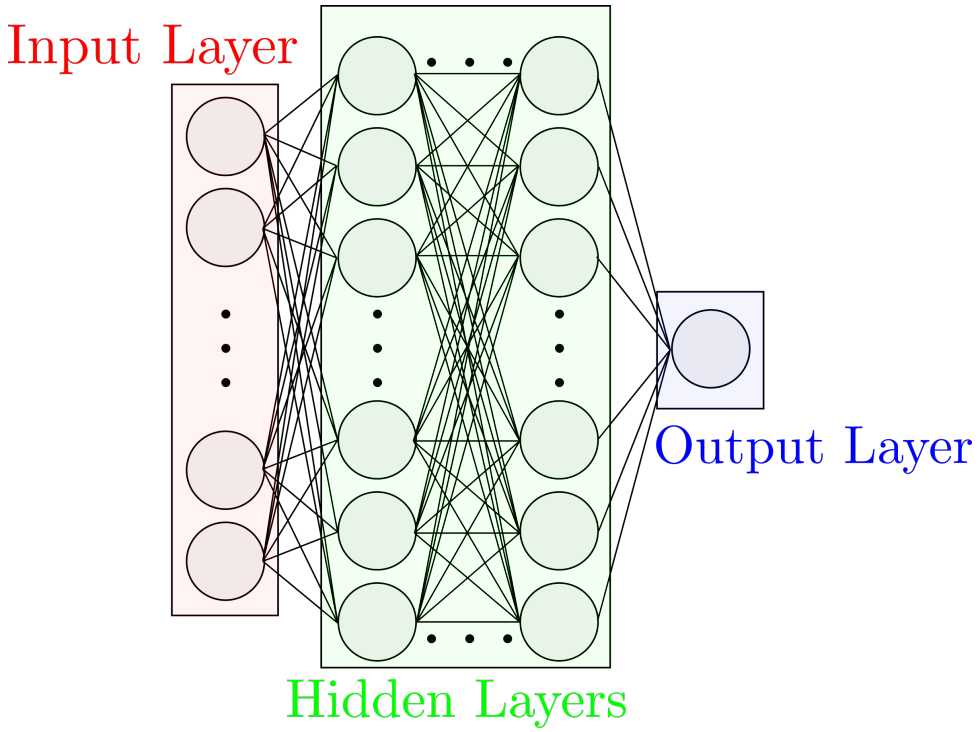


Figure 3.6 A neural network architecture, highlighting the three kinds of layers.

result.

$$y = w \cdot x + b \tag{3.8}$$

Before sending y to the next layer, we apply an activation function f. It is instrumental for this function to be non-linear. Without a non-linear activation function, a neural network, no matter how many layers it had, would behave as a single-layer perceptron. Thus it will not be useful to work on non-linear problems. One frequent activation function is the sigmoid σ, the Equation (3.9) shows its definition. The last output returns the probability of the input to be the desired target. If closer to 1, it is classified as "good", otherwise, as "bad". Good and bad are defined by the training. For instance, in gunshot classification, good can be a gunshot and bad a car crash sound.

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{3.9}$$

The backpropagation is related to the training part. It is a procedure of maximizing the similarity between predicted outputs and real outputs. The dataset that contains all the training samples is assumed as a general set that sufficiently represents what the neural network needs

to understand to classify further cases. Through gradient-based learning, the backpropagation algorithm updates the weight and bias vectors of all layers following the rule in Equation (3.10):

$$\begin{aligned} w &:= w - \mu \nabla J, \\ b &:= b - \mu \nabla J. \end{aligned} \tag{3.10}$$

The gradient is calculated by the chain rule, computing a series of multiplications related to the derivative in each layer. Also, it is scaled by the learning rate (μ) hyperparameter. The way to measure how close the neural network is to the optimized configuration is by minimizing the cost function J. This cost function can be many functions, but a common one is the squared error, shown in Equation (3.11). It calculates the "distance" between the real output (y) and the predicted one ($\hat{y}$).

$$J = \frac{1}{2}(y - \hat{y})^2. \tag{3.11}$$

CHAPTER 4

Methodology and Development

Science is not only a disciple of reason but also one of romance and passion.

—STEPHEN HAWKING

The system was developed following the schematic shown in Figure 4.1. It is separated into five modules: 1) acquisition module, 2) digital signal processing, 3) neural network, 4) localization algorithm, and 5) mobile communication. This chapter presents detailed discussions about iOwlT's implementation. The whole implementation code can be found in the GitHub repository [32].

4.1 Acquisition Module

The acquisition module is divided into the physical system's implementation (umbrella geometry) and the preamplifier circuit.

4.1.1 Umbrella Geometry

To ensure good accuracy in obtaining the arrival angles of the audible signal, a symmetrical base made up of 5 microphones was chosen in the iOwlT system. To avoid symmetry problems with the normal axis to that base, a central microphone with a difference in height from the base was used as a reference, as shown in Figure 4.2. The mathematical definition of the umbrella geometry is described in Table 4.1.

The geometry is similar to an umbrella. It has a pentagonal base plane with five legs of 1 m long and a central upward vertical thin bar with another base plane. These five legs have a microphone attached at each end responsible for the localization procedure. The central upper

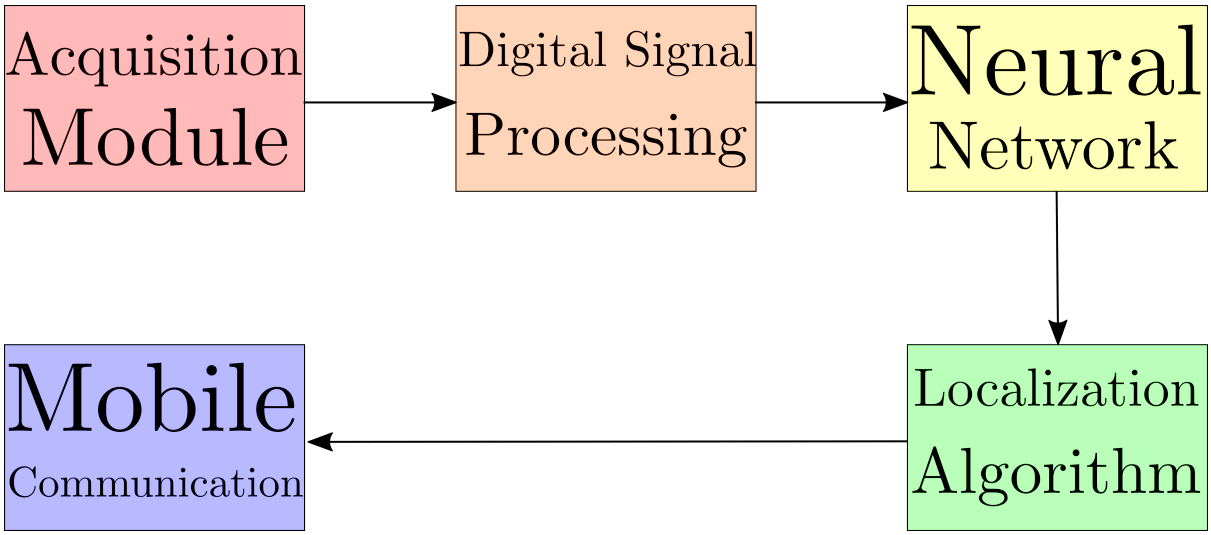


Figure 4.1 iOwlT's system overview.

Sensor	x	y	z
1	+0.00	+0.00	+0.00
2	+1.13	+0.00	−0.55
3	+0.35	+1.08	−0.55
4	+0.35	−1.08	−0.55
5	−0.92	−0.67	−0.55
6	−0.92	−0.67	−0.55

Table 4.1 Mathematical definition of the umbrella geometry.

Figure 4.2 Umbrella geometry used in iOwlT system.

base plane has another microphone, which is related also to the classification part.

Plane bases and legs are made of Medium Density Fiberboard (MDF) wood, the vertical bar is made of rigid steel, and the cables connecting the upper base and the legs are made of flexible steel. Also, the principal base has a fit for standard camera tripods, this way we can adjust the height easily and accurately.

4.1.2 Preamplifier Circuit

The system has six 1.5 m lavalier microphones to capture the environmental sounds. Each of these microphones has a P2-male output with insufficient signal amplitude to be connected to the A/D Converter. So we designed a preamplifier circuit (Figure 4.3).

The project was made using the DE10-Nano FPGA, which A/D Converter has a maximum input voltage of 5 V. The preamplifier circuit can be divided into four parts. The first one is the

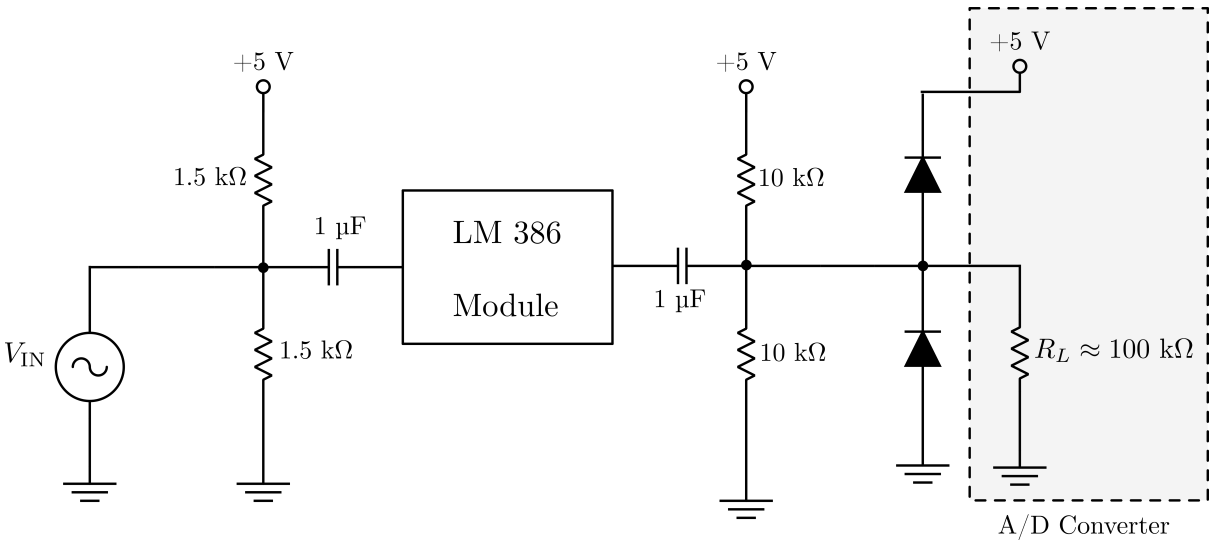


Figure 4.3 Preamplifier circuit.

voltage divider with resistors of 1.5 kΩ responsible for polarizing the microphone. The second one is the LM 386 Module, which amplifies the microphone (AC component) signal. The third one is the voltage divider with resistors of 10 kΩ, responsible for adding a DC component to spans the signal in the range of 0 to 5 V. The last one is the diode pair, responsible for protecting the DE10-Nano board. See its final implementation in Figure 4.4.

4.2 Digital Signal Processing

4.2.1 Analog to Digital Conversion

The DE10-Nano FPGA has the LTC2308 A/D Converter chip, which has 8 channels ($C = 8$), voltage range from 0 to 5 V and 12 bits of resolution. The maximum rate of samples collected per second is given by $f_{s_{\max}} = 500$ kSPS. The resolution in volts and the maximum sample rate per channel are then calculated as

$$\text{Voltage Resolution} = \frac{5}{2^{12}-1} \approx 1.22 \text{ mV}, \tag{4.1}$$

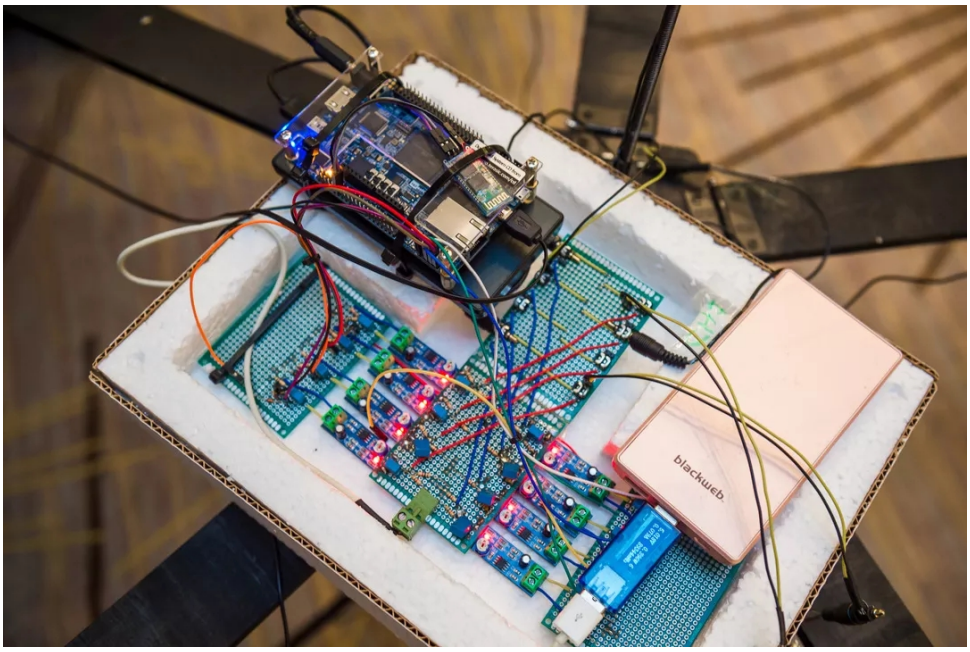


Figure 4.4 Final implementation of the preamplifier circuit.

$$\overline{f_{s_{max}}} = \frac{500\text{ k}}{8} = 62.5\text{ kHz}. \tag{4.2}$$

The control module of A/D Converter on the FPGA was obtained directly from the University Program IP core present in Quartus software, through a component available to control A/D Converter for DE-series boards. The configuration of this module is done directly, just needing the DE-Series board name, the A/D Converter clock frequency, and the number of channels used. An example for a DE0-Nano-SoC with A/D Converter clock frequency of 12.5 MHz, and 2 channels is shown in the Figure 4.5.

This module returns a control code for the A/D Converter that provides a number of outputs equal to the number of channels given, so that whenever a new sampling is performed each of these outputs returns a 12-bit vector with the result of the quantized sample.

The sampling rate chosen to operate the system was $f_s = 16$ kHz, which showed a good compromise between good signal representation and localization accuracy.

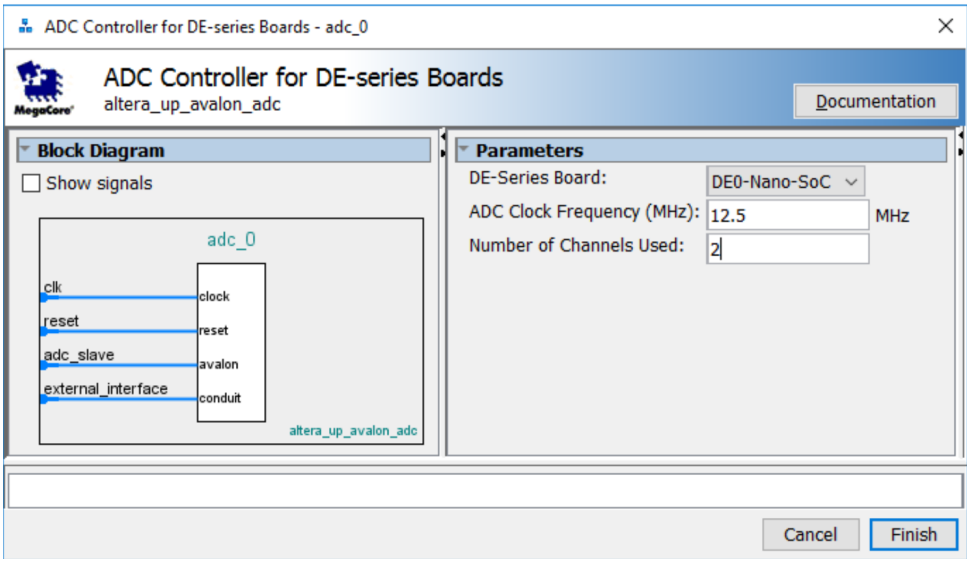


Figure 4.5 A/D Converter controller configuration example.

4.2.2 Digital Filtering

A simple first-order high-pass filter was used in order to eliminate the DC component of the signal coming from the A/D Converter and also other lower frequency components. It was designed to have a cutoff frequency $f_c = 100$ Hz at a sampling rate of $f_s = 16$ kHz.

Since all the arithmetic used was based on signed 12-bit integers, the FPGA would not be able to interpret floating-point operations. The parameter α, that is between 0 and 1, had to be chosen using an integer ratio. This way, the α multiplication operation could be estimated by multiplying by an integer followed by a division by another integer. The parameter α was then chosen as

$$\alpha = \frac{24}{25} = 0.96. \tag{4.3}$$

Therefore, the cutoff frequency of the filter is given from (3.7) as

$$f_c = \frac{1-\alpha}{2\pi\alpha T} = \frac{1-0.96}{2\pi \cdot 0.96} \cdot 16\text{ k} \approx 106\text{ Hz}. \tag{4.4}$$

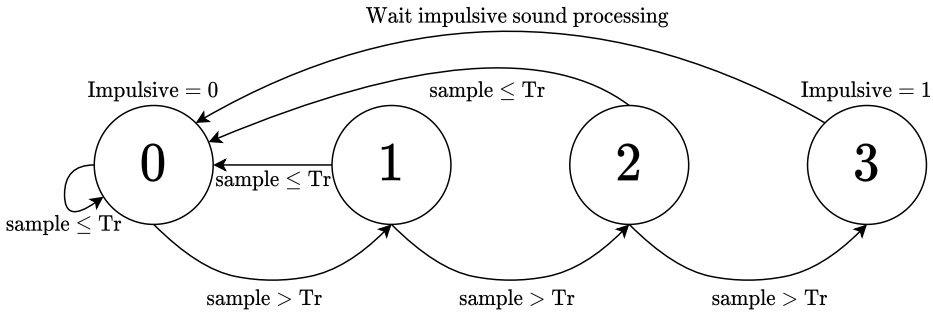


Figure 4.6 Threshold finite state machine.

4.2.3 Threshold

The impulsive signal is detected through a simple peak detector, as illustrated by the state machine in Figure 4.6. The operating logic is simple: if three consecutive samples have a value greater than the threshold Tr, then the signal is said to be impulsive and the processing is done, otherwise, the system is stuck between stages 0, 1, and 2, and the sound is not identified as impulsive.

The impulsive sound only needs to be detected by a single microphone, so the central microphone, the reference, was chosen to perform the detection of the impulsive sound.

4.2.4 Circular Buffer

For each of the six microphones in the system, there is an associated circular buffer of size equal to 8,000, as a windowing of 0.5 s was used at a sample rate of 16 kHz. This entire structure is designed inside a double access RAM implemented in FPGA, so that it could be accessed both by the FPGA part and by the ARM processor that the DE10-Nano board has.

Similar to the A/D Converter controller, it is possible to build a shared memory between ARM and FPGA using an existing tool in Quartus. The use of this tool is illustrated in Figure 4.7. In the case of the iOwlT system, the memory needed to have words of at least 16 bits (the power of 2 immediately higher than the 12 bits of discretization from the A/D Converter) and at least $6 \cdot 8000 = 48000$ memory positions occupied by the circular buffer, plus some extra positions to store flags and other information exchanged between devices. For 48000 memory

positions with 16 bits (2 bytes) each, the equivalent of 96000 bytes is required, which would lead to a memory size of at least $2^{17} = 131072$ bytes, the value used in the system.

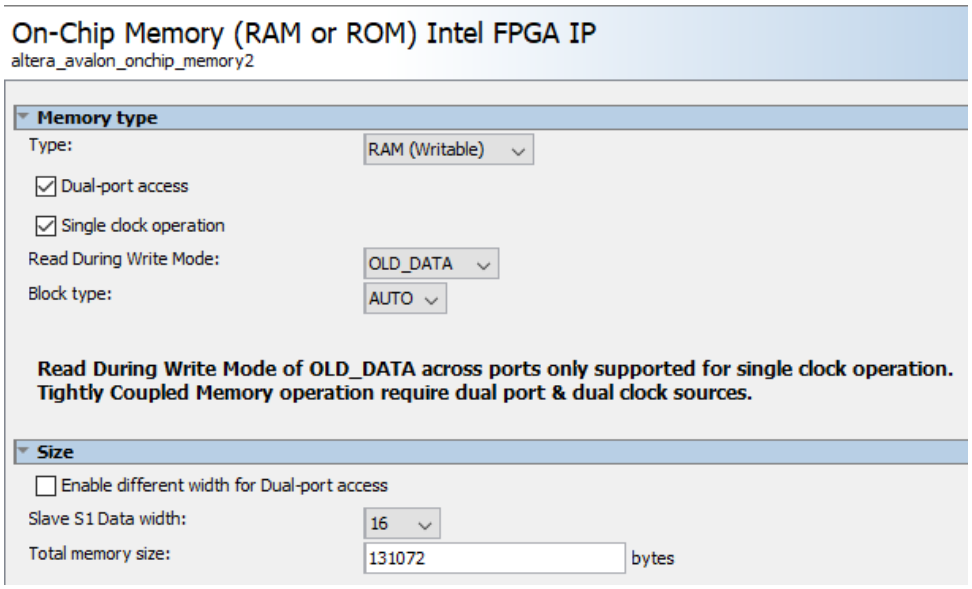


Figure 4.7 2-Port RAM configuration window example.

4.2.4.1 Memory considerations

As explained, the size of the memory depends directly on the number of circular buffers used (given by the number of microphones) and the size of each one, which depends on the window and sampling rate chosen for the signal. Therefore, this memory is configurable and can be changed in size depending on the number of microphones in the system and also on the type of signal you want to detect, which will define the size of the buffers through a window and a sampling rate, necessary to represent the signal.

Also in terms of the size of the implemented memory, the value used corresponds to less than 19% ($\approx 18.4\%$) of the maximum possible value for the chosen FPGA, which has an embedded memory of 5570 kbits.

Figure 4.8 The Brazilian Special Ops training camp in Recife, PE.

4.3 Neural Network

Our neural network is a simple fully-connected multilayer perceptron, with an input layer composed of 637 neurons, two hidden layers composed of 200 and 10 neurons and a last output layer with a single neuron.

It is worth mentioning that the input of 637 neurons is due to the feature extraction from MFCC. The MFCC algorithm was implemented exactly the same way it is done with speech recognition and music applications, further investigations can be done to improve the method for impulsive signals. Before it, the sound vector is a fraction (window) resulting in 8,000 samples.

To train it, we went to the BOPE training camp, where we could record more than 100 shots (see in Figure 4.8). Initially, the idea was to set these gunshots as positive labels, however, we needed to do a live demonstration in China during the Grand Final. So we decide to train the neural network to recognize claps (as positive labels). To negative labels, we used these gunshots, and also we recorded other impulsive sounds as popping bags, glass sounds, etc. First,

this neural network was implemented in Python with Keras library [33]. After trained, we implemented, from scratch, the same neural network (with the last weight and bias vectors) with only the forward propagation part in C language, with support of the Eigen library [34]. This way the ARM could interpret and compute the results, boosting performance.

4.4 Localization Algorithm

The localization algorithm used is the one presented in Section 2.1.1. This algorithm was implemented in C language and runs in the existing ARM processor of DE10-Nano board. Whenever an impulsive sound is identified, the vectors with the sound samples are passed from the FPGA to the ARM, the neural network tests if the reference microphone sound is a desired impulsive sound and if it is the localization algorithm is executed. The necessary correlations for TDOA calculations are all implemented using FFT using the FFTW library [35], and the solution of the linear system of the algorithm is obtained through the Eigen library [34].

4.5 Mobile Communication

Mobile communication was made via Bluetooth with the HC-06 module, shown in Figure 4.9, which is connected directly to the General Purpose Input/Output (GPIO) of the FPGA. At the end of processing an impulsive sound that has been identified as a desired sound by the neural network, the ARM processor sends two flags to the FPGA. The first says that the FPGA can look for the next impulsive sound, and the second activates the Bluetooth controller implemented in FPGA. The Bluetooth controller sends the SSL information in coordinate format to a smartphone connected to the module. This information is present in the ARM processor, and is sent in Unicode format to FPGA through the memory shared between them. The communication is all done at a baud rate of 9,600 bps through the Bluetooth controller on the FPGA, which performs communication via RS-232 protocol.

The Serial Bluetooth Terminal program was installed on a smartphone, which displays the

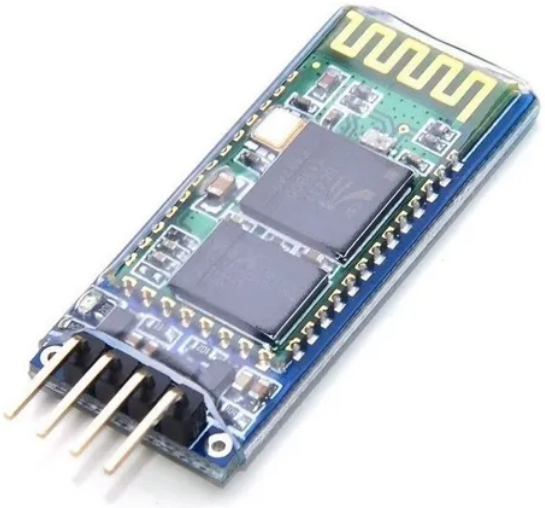


Figure 4.9 HC-06 Bluetooth module.

messages received by the system. An example of the message received is shown in Figure 4.10.

The system does not store the data of the detected and located signals, it is only concerned with sending them to the mobile device. However, if necessary, you can save this data from the operating system that works in the ARM, saving it into the memory of the SD card connected to the FPGA.

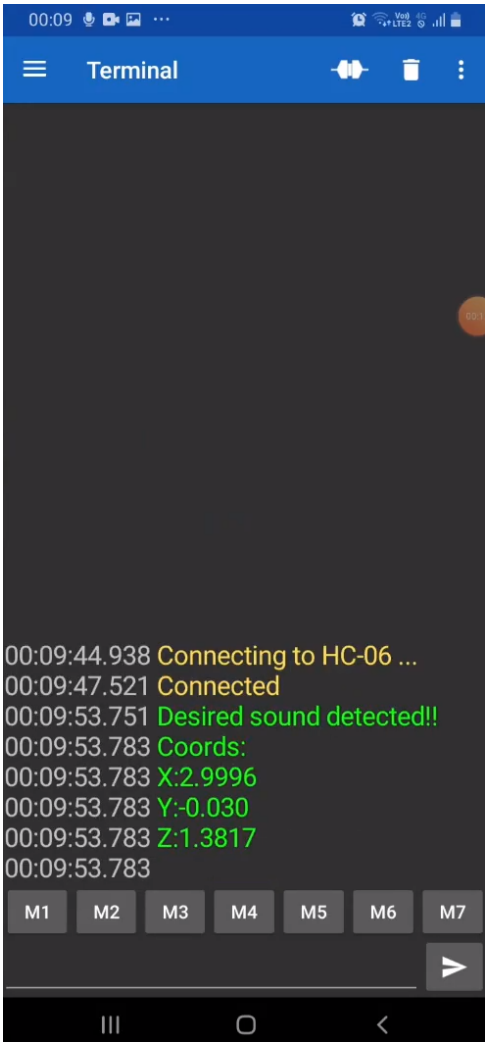


Figure 4.10 Example of message sent by the system.

CHAPTER 5

Results and Discussion

I often spoke like a clown but I never doubted the sincerity of the audience that smiled.

—CHARLES CHAPLIN

5.1 System Performance

The system performance can be summarized in the results related to the neural network and the localization algorithm.

5.1.1 Neural Network

The neural network presented a total performance of 91.38% in the classification of claps. With 89.74% accuracy when classifying as a clap when it is truly a clap and 94.74% when classifying as not clap, when it is truly a not clap. The confusion matrix is shown in Figure 5.1.

5.1.2 Localization Algorithm

The localization algorithm evaluation is divided into two tables. Table 5.1 presents the performance of evaluating the angle direction of the sound source, and Table 5.2 its distance. These distance errors are calculated by taking the module of difference between the predicted value and the real value and dividing it by the real value. The errors associated with the direction angles were calculated in a similar way, only with a change in the normalization factor, where 90^{o} is used, which is the largest variation that exists in a quadrant. On average, the performance of the tables shown are 97.21% and 88.32% on determining impulsive sound direction and

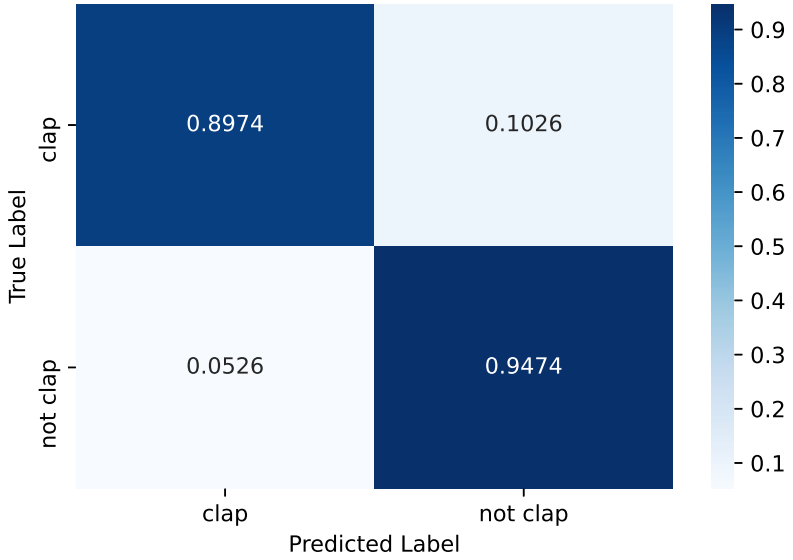


Figure 5.1 Neural network clap classification confusion matrix.

Quadrant	5 m	10 m	15 m	20 m
Quadrant – I	1.63%	2.33%	3.21%	4.45%
Quadrant – II	0.79%	1.68%	2.54%	3.02%
Quadrant – III	1.01%	3.12%	3.88%	4.74%
Quadrant – IV	1.52%	2.16%	3.44%	5.12%

Table 5.1 Angle direction error.

position, respectively. The experiment was conducted in an open area, without any barriers and interference.

The tests presented extended up to 20 m away from the sensor geometry, so that above that distance the system started to show larger errors, mainly in distance.

5.1.3 InnovateFPGA 2019 Design Contest

"The InnovateFPGA is a global FPGA design contest where teams from around the world compete as they invent the future of Artificial Intelligence with Terasic and Intel. The competition

Quadrant	5 m	10 m	15 m	20 m
Quadrant – I	8.63%	10.11%	13.57%	15.12%
Quadrant – II	5.63%	7.77%	8.81%	12.24%
Quadrant – III	7.09%	9.21%	11.95%	17.47%
Quadrant – IV	8.00%	12.34%	17.37%	21.54%

Table 5.2 Distance error.

is open to everyone including students, professors, makers, and industry." The competition was divided into two phases with a total of 270 submissions.

The first one was regional, where we competed against 40 teams from North, Central, and South America. Our system ranked 2nd, earning both the Silver Award and the Community Award (elected, by the community, as the best project regionally).

The second phase, the Grand Final, was held in Tianjin, China, in December of 2019. In this phase, the best 11 teams demonstrated and presented their projects to a committee of experts in the field of FPGAs and artificial intelligence. Our system ranked 2nd, earning the Silver Award. See in Figure 5.2 the final celebration.

In summary, iOwlT: Sound Geolocalization System got three international awards among 270 projects, including some developed by graduate students at top-ranked schools (University of Illinois at Urbana-Champaign, University of Pittsburgh) and researchers at well-known companies (Microsoft, Intel).

5.2 Discussions

The system had a better performance at determining the direction of the sound, with a 5.12% error in the worst-case scenario.

The biggest error in the distance when compared to the direction can be understood through the hyperboles (or hyperboloids) that describe the multilateration system. The farther away the sound source is, the closer it gets to the asymptotes of these hyperbolas, so that small errors in

Figure 5.2 Silver Award at InnovateFPGA 2019 Design Contest.

obtaining TDOA can lead to estimates of the source that are closer to or further away from these asymptotes, generating a nonlinear error of distance that increases with the distance from the source. The fact that errors at large source distances occur for points close to the asymptotes, which have a fixed associated angulation, means that an error associated with long distances does not affect angulation errors as much.

5.2.1 The Echo Problem

Some of the effects that contribute to these localization errors are errors obtained due to: the TDOA digitization process, small inaccuracies in the measurement of the microphone coordinates and possible echo effects that may arise depending on the environment. The first error factor has already been considered in this work and the second concerns human inaccuracies in the construction and measurement of the equipment. The third problem, that of the echo, is challenging, and a comment about it is relevant, since in urban environments it is common to have many barriers in which the sound can be reflected and the echo appears.

The echo mimetics the sound emitted by the source in another position, creating a pseudo-source emitting the same sound. It deals in confusion in the localization method system, as it uses only the phase difference of the sound to doing calculations.

One possibility to amenize the echo problem is to treat the signal with adaptive filters. In more open areas, the echo is not a problem, so the addition of those filters is a decision that can be done based on the application.

5.2.2 Optimizing Hardware Parameters

Due to limitations of time, we did not try other geometries to compare. After the competition, we continued the development of iOwlT, studying better configurations of hardware parameters. These configurations include evaluating novel geometries, sampling rate, and the number of sensors. Then, we wrote the paper “Optimization of Hardware Parameters on a Real-Time Sound Localization System” which is currently in submission.

This paper contributes with three highlights:

- Propose a model for evaluating configurations of SSL depending on geometric and computational parameters;
- Develop an optimization method to find the best parameters under specific environments, both indoor and outdoor;
- Describe a comprehensive comparison between found configurations and related works, performing up to 33.0% better.

CHAPTER 6

Conclusion

The most beautiful things in the world cannot be seen or touched, they are felt with the heart.

—ANTOINE DE SAINT-EXUPÉRY (The Little Prince)

We have successfully implemented an intelligent cyber-physical SSL system inspired by owls and focused on locating impulsive sounds, as gunshots. The system achieved good results and proved the technical feasibility of the idea.

The use of FPGA was instrumental to develop iOwlT. The system is based on TDOA, thus extremely dependant on the accuracy of their acquisitions. Any measurement error would imply considerable deviations in the result as the speed of sound is large. Also, as the TDOAs are very small, it would be difficult to establish a synchronous and accurate acquisition routine in software. At this point, hardware implementation was critical for the system to work properly. The parallelism and synchronization of hardware-implemented circular buffers underline their importance for the system.

The system can have further improvements. The neural network can be trained with more data, or can even be trained to discriminate different kinds of impulsive sounds, as different guns (rifles, pistols, submachine guns, etc). Following the discussions presented in Section 5.2, we can use the fact we already calculate phase under cross-correlation techniques to implement echo-removal filters. Also, we can testify the results from "Optimization of Hardware Parameters on a Real-Time Sound Localization System" of novel configurations that can bring improvements. Better A/D Converter and FPGA can allow a more robust design. For instance, feature extraction and neural networks implemented in hardware would increase time performance.

Bibliography

[1] Eric Knudsen and Masakazu Konishi. “Mechanisms of sound localization in the barn owl (Tyto alba)”. In: *Journal of Comparative Physiology* 133 (Jan. 1979), pp. 13–21. DOI: 10.1007/BF00663106.

[2] Eric Knudsen. “Instructed learning in the auditory localization pathway of the barn owl”. In: *Nature* 417 (June 2002), pp. 322–8. DOI: 10.1038/417322a.

[3] C. Zhou et al. “A Robust and Efficient Algorithm for Coprime Array Adaptive Beamforming”. In: *IEEE Transactions on Vehicular Technology* 67.2 (2018), pp. 1099–1112.

[4] Changkyu Choi et al. “Real-time audio-visual localization of user using microphone array and vision camera”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 1935–1940. DOI: 10.1109/IROS.2005.1545030.

[5] J. C. Chen, R. E. Hudson, and Kung Yao. “Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field”. In: *IEEE Transactions on Signal Processing* 50.8 (2002), pp. 1843–1854. DOI: 10.1109/TSP.2002.800420.

[6] F. Grondin and F. Michaud. “Noise mask for TDOA sound source localization of speech on mobile robots in noisy environments”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 4530–4535.

[7] C. Knapp and G. Carter. “The generalized correlation method for estimation of time delay”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.4 (1976), pp. 320–327. DOI: 10.1109/TASSP.1976.1162830.

[8] Jan Scheuing and Bin Yang. “Correlation-Based TDOA-Estimation for Multiple Sources in Reverberant Environments”. In: *Speech and Audio Processing in Adverse Environments*. Ed. by Eberhard Hänsler and Gerhard Schmidt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 381–416. ISBN: 978-3-540-70602-1. DOI: 10.1007/978-3-540-70602-1_11. URL: https://doi.org/10.1007/978-3-540-70602-1_11.

[9] Khalaf-Allah. “Performance Comparison of Closed-Form Least Squares Algorithms for Hyperbolic 3-D Positioning”. In: *Journal of Sensor and Actuator Networks* 9.1 (Dec. 2019), p. 2. ISSN: 2224-2708. DOI: 10.3390/jsan9010002. URL: http://dx.doi.org/10.3390/jsan9010002.

[10] Pinghui Wu et al. “Time Difference of Arrival (TDoA) Localization Combining Weighted Least Squares and Firefly Algorithm”. In: *Sensors* 19 (June 2019), p. 2554. DOI: 10.3390/s19112554.

[11] Yiteng Huang et al. “Real-time passive source localization: a practical linear-correction least-squares approach”. In: *IEEE Transactions on Speech and Audio Processing* 9.8 (2001), pp. 943–956. DOI: 10.1109/89.966097.

[12] Y. T. Chan and K. C. Ho. “A simple and efficient estimator for hyperbolic location”. In: *IEEE Transactions on Signal Processing* 42.8 (1994), pp. 1905–1915.

[13] R. A. Hooshmand, M. Parastegari, and M. Yazdanpanah. “Simultaneous location of two partial discharge sources in power transformers based on acoustic emission using the modified binary partial swarm optimisation algorithm”. In: *IET Science, Measurement Technology* 7.2 (2013), pp. 112–118.

[14] Marko Kovandžić et al. “Near Field Acoustic Localization Under Unfavorable Conditions Using Feedforward Neural Network For Processing Time Difference Of Arrival”. In: *Expert Systems with Applications* 71 (Nov. 2016). DOI: 10.1016/j.eswa.2016.11.030.

[15] José Fresno et al. “Survey on the Performance of Source Localization Algorithms”. In: *Sensors* 17 (Nov. 2017), p. 2666. DOI: 10.3390/s17112666.

[16] Guillermo Robles et al. “Antenna array layout for the localization of partial discharges in open-air substations”. In: *Proc. Int. Electron. Conf. Sens. Appl.* Vol. 2. Nov. 2015. DOI: 10.3390/ecsa-2-E008.

[17] J. Hu et al. “Geometrical arrangement of microphone array for accuracy enhancement in sound source localization”. In: *2011 8th Asian Control Conference (ASCC)*. 2011, pp. 299–304.

[18] B. Yang and J. Scheuing. “Cramer-Rao bound and optimum sensor array for source localization from time differences of arrival”. In: *Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 4. 2005, iv/961–iv/964 Vol. 4.

[19] Haitao Liu, Thia Kirubarajan, and Qian Xiao. “Arbitrary Microphone Array Optimization Method Based on TDOA for Specific Localization Scenarios”. In: *Sensors* 19 (Oct. 2019), p. 4326. DOI: 10.3390/s19194326.

[20] G. Liu et al. “A Sound Source Localization Method Based on Microphone Array for Mobile Robot”. In: *2018 Chinese Automation Congress (CAC)*. 2018, pp. 1621–1625.

[21] Jwu-Sheng Hu and Chia-Hsing Yang. “Estimation of Sound Source Number and Directions under a Multisource Reverberant Environment”. In: *EURASIP J. Adv. Sig. Proc.* 2010 (Feb. 2010). DOI: 10.1155/2010/870756.

[22] Anthony Badali et al. “Evaluating Real-time Audio Localization Algorithms for Artificial Audition in Robotics”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*. Dec. 2009, pp. 2033–2038. DOI: 10.1109/IROS.2009.5354308.

[23] J. Hu et al. “Simultaneous localization of mobile robot and multiple sound sources using microphone array”. In: *2009 IEEE International Conference on Robotics and Automation*. 2009, pp. 29–34. DOI: 10.1109/ROBOT.2009.5152813.

[24] J. Nikunen and T. Virtanen. “Time-difference of arrival model for spherical microphone arrays and application to direction of arrival estimation”. In: *2017 25th European Signal Processing Conference (EUSIPCO)*. 2017, pp. 1255–1259. DOI: 10.23919/EUSIPCO.2017.8081409.

[25] Jean-Marc Valin et al. “Localization of Simultaneous Moving Sound Sources for Mobile Robot Using a Frequency-Domain Steered Beamformer Approach”. In: *Proceedings - IEEE International Conference on Robotics and Automation*. Vol. 2004. Jan. 2004, pp. 1033–1038. DOI: 10.1109/ROBOT.2004.1307286.

[26] Daniel et al. “Atlas da violência 2020”. In: *IPEA* (2020).

[27] The Global Burden of Disease 2016 Injury Collaborators. “Global Mortality From Firearms, 1990-2016”. In: *JAMA* 320.8 (Aug. 2018), pp. 792–814. ISSN: 0098-7484. DOI: 10.1001/jama.2018.10060.

[28] Peerapol Khunarsa, Chidchanok Lursinsap, and Thanapant Raicharoen. “Impulsive Environment Sound Detection by Neural Classification of Spectrogram and Mel-Frequency Coefficient Images”. In: *Advances in Neural Network Research and Applications*. Ed. by Zhigang Zeng and Jun Wang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 337–346. ISBN: 978-3-642-12990-2. DOI: 10.1007/978-3-642-12990-2_38. URL: https://doi.org/10.1007/978-3-642-12990-2_38.

[29] *iOwlT: Sound Geolocalization System – InnovateFPGA 2019 Design Contest submission*. URL: http://www.innovatefpga.com/cgi-bin/innovate/teams.pl?Id=AS026 (visited on 03/25/2021).

[30] Stephen Boyd. *Lecture 1 – Signals*. 2003. URL: https://stanford.edu/~boyd/ee102/signals.pdf.

[31] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & Systems (2nd Ed.)* USA: Prentice-Hall, Inc., 1996.

[32] Matheus Farias and Davi Moreno. *iOwlT: Sound Geolocalization System*. 2019. URL: https://github.com/matheussfarias/iOwlT-Sound-Geolocalization-System.

[33] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[34] Gaël Guennebaud Benoît Jacob et al. *Eigen*. 2006. URL: https://eigen.tuxfamily.org/index.php?title=Main_Page.

[35] Matteo Frigo and Steven G. Johnson. *FFTW Library*. 2005. URL: http://www.fftw.org/.

APPENDIX A

Original Work

AS 026
iOwlT: Sound Geolocalization System

Team members：Davi Almeida, Gabriel Firmo, Matheus Farias

Organization: Universidade Federal de Pernambuco

Instructors: Daniel Filgueiras, Edna Barros

I. High-level Project Description

Acoustic systems of location and event identification have several applications in the everyday world, being present in security systems, earthquake recognition, sonar and various types of man-machine iteration.

Shooting sound mapping techniques began to be implemented in the last decades, even though it has been a problem of interest since the mid-First World War. In addition to military practices and environmental protection (e.g. detection of hunters in forbidden areas), this mechanism can be used in urban areas, providing instantaneous data to the local police or collecting data for further study of violence in certain areas.

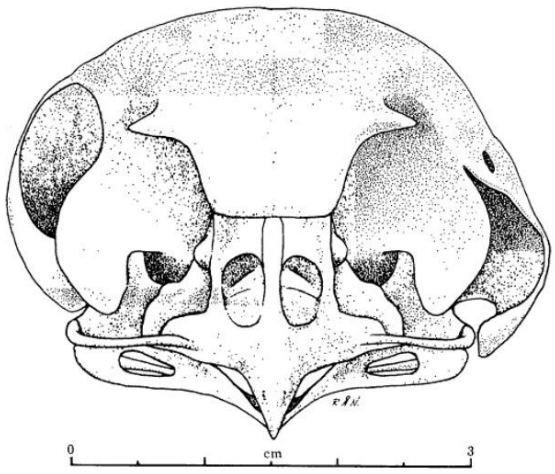
Aiming to recognize and map specific types of sounds, an idea of an intelligent and self-adaptive system was developed based on the functioning and learning of the auditory system of species of owls.

Owls are animals that possess a powerful hunting ability during the night, and to accomplish such a feat, as at night the sight is naturally more overshadowed by the absence of light, the owl has to use other benefits of evolution to improve accuracy of predicting the location of your dinner, one of them is the sound.

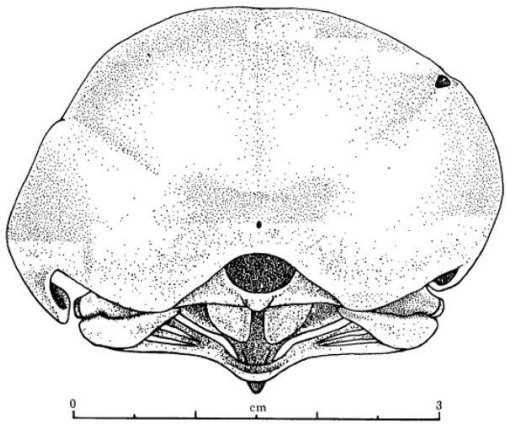
Experiments conducted by neurobiologists Eric I. Knudsen and Masakazu Konishi[1] have been able to prove, using barn owls as the species of study, that this species of owl is able to locate a prey being immersed in a totally dark room, only using the sound emitted by its prey.

A barn owl

The great evolutionary advantage present in this species is related to the considerable asymmetry that exists between their ears, it is known that the left ear is positioned around centimeters below the right ear, and with this difference of height, the owls can receive the information of the emitter with phase shift. From this difference, it becomes possible to accurately measure the location of its target, much like the triangulation positioning process, widely used in telecommunications engineering with telephone networks, or even in satellites. A very interesting video produced by the BBC[2] demonstrates the whole hunting process of this species.



Front vision of barn owl skull



Back vision of barn owl skull

Owls that locate their prey using a sharp hearing aid are not born[3] with this technique already well developed, thus necessitating an apprenticeship to adapt to their own physical characteristics (skull diameter, height difference between the ears, etc.) that can vary significantly in the same species, beyond that, the owls have on the side of the head, channels of rigid feathers that can regulate the passage of sound. Thus, these animals have a very efficient adaptive control, allowing that the accuracy in the location prediction maintains high even when dealing with different environmental conditions or physiological differences inherent to the species.

The technique of finding the coordinates of an unknown source from delays in reception of the signal in receivers distributed in a known manner in space is part of a technique called **multilateration**, which has no trivial solution. It is possible to show with algebra that in an N dimensional space N+1 receivers are needed, with known positions, to uniquely determine the coordinates of an unknown source.

Taking a case of easier visualization, there are 3 known receptors $\mathbf{R_1}$, $\mathbf{R_2}$ and $\mathbf{R_3}$ and a target **T** with unknown location in an x-y plane.

$$R_1 : (x_1, y_1),\ R_2 : (x_2, y_2),\ R_3 : (x_3, y_3),\ T : (x, y)$$

When **T** emits a sound, the receivers detect the signal at different time. Without loss of generality, consider that $\mathbf{R_1}$ will receive the information first in a time **t**, $\mathbf{R_2}$ in $\mathbf{t+dT_1}$ and $\mathbf{R_3}$ in $\mathbf{t+dT_2}$.

To calculate the distance between **T** and the i-receptors we have:

$$d_i = v \cdot t_i$$

Where **v** is the sound velocity and $\mathbf{t_i}$ is the time of arrival of the signal from **T** to the i-th receptor

$$d_1 = v \cdot t$$

$$d_2 = v \cdot (t + \Delta t_1) = d_1 + v \cdot \Delta t_1 = d_1 + \Delta s_1$$

$$d_3 = v \cdot (t + \Delta t_2) = d_1 + v \cdot \Delta t_2 = d_1 + \Delta s_2$$

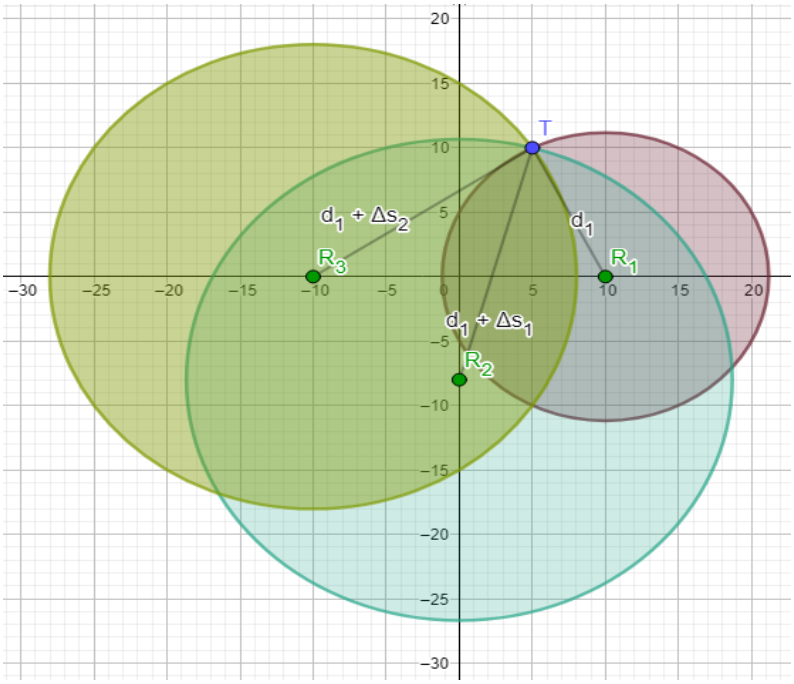
Centered at each of these receptors one can draw the circles $\mathbf{C_1}$, $\mathbf{C_2}$ and $\mathbf{C_3}$:

$$C_1 : (x - x_1)^2 + (y - y_1)^2 = d_1^2$$

$$C_2 : (x - x_2)^2 + (y - y_2)^2 = (d_1 + \Delta s_1)^2$$

$$C_3 : (x - x_3)^2 + (y - y_3)^2 = (d_1 + \Delta s_2)^2$$

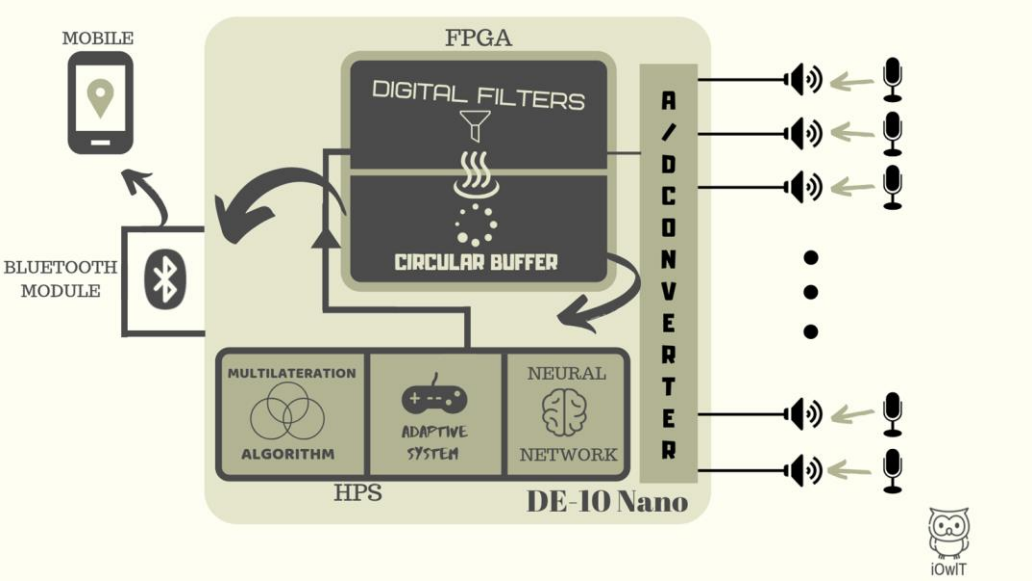
Drawing the circles in an x-y plane, we have:



The only unknown variables to this system of equations are **x**, **y**, and $\mathbf{d_1}$. For purposes, it is possible to solve this system by applying direct minimization techniques, otherwise, in the real case, with noises and inaccuracies, these circles do not have an intersection and we need to define cost functions with numerical algorithms (e.g. gradient descent) that minimizes the error and find and approximate value for T.

II. Block Diagram

Figure bellow is the system block diagram which can be divided into three parts: The Acquisition Circuit, FPGA and HPS (ARM).



2.1 Acquisition Circuit

This is an amplifier circuit that was designed to obtain the signal received by the microphones in voltage form varying in the range allowed by the *DE-10 Nano* A/D Converter, which receives the signal.

The *DE-10 Nano* board has an analog-to-digital converter of only one output, so the signals picked up by the sound detectors pass through a 8:1 multiplexer, observing the datasheet, it is noted that this mux takes a total of 3µs to switch, and therefore the largest possible delay in the acquisition of the signals, i.e. considering 8 sound detectors, is 3x7 = 21µs, as the audible frequency is in the range of 20Hz to 20kHz, using the Nyquist theorem, the sampling rate for the set of observed signals is 40kHz, and this results in 25µs, so the analysis of the signals is practically simultaneous, leading to an increase of considerable performance as well.

2.2 FPGA

The FPGA will have 4 essential modules, the A/D converter controller, that do the communication with the A/D converter and control the sampling rate of the signal, the Digital Filters module, that will process the sound digitalized by the A/D converter and remove low frequency noise, using precalculated parameters

adapted to the type of noise and type of impulsive sound that the system will recognize, the Circular Buffer module, that stores the sound signals in circular buffers to after send to the HPS, and the Bluetooth communication, that simply guarantee the communication between the cellphone and the equipment.

2.3 HPS (ARM)

The HPS will have 3 modules, the Adaptive System module, that will change some threshold parameter to adapt the solution to the environment, the Multilateration Algorithm, which is the correlation operation combined with the

effective measurement of where is the sound emitter using the multilateration technique and the Neural Network, which is responsible to determine if the sound is the desired sound or not.

III. Intel FPGA Virtues in Your Project

3.1 Adapt to changes

Processing of sound signals made in the FPGA is supported by an adaptive threshold system, that will vary depending on the distance of the sound source, type of sound (being more general than gun shots) and ambient noise. This will result in a change in the threshold variable which will be adapted to help in the sound recognition. Therefore, the iOwlT system is adaptable to this feature.

The present project can be used not only embedded on a police car. Depending on the use of the technology, the iOwlT system may well be positioned in static strategic positions, such as on traffic lights, an interesting application would be to identify a possible earthquake imminence, since the onset of a seismic shake is determined much earlier by sound signals of high intensity but with very low frequency, being audible to animals like horses but not to humans. Such sonorous signals could be identified by the iOwlT system, and therefore there would be a longer preparation time for the coming earthquake.

3.2 Boost Perfomance

The iOwlT system, using FPGA technology, can performs the multilateration algorithm with an outstanding precision, using the idea of a circular buffer to be

the data structure that stores the sound received by each microphone. As the microphones receives the signal with a phase difference, it's possible to see very clear the phase difference by counting the pivot index difference with correlation. As discussed in FPGA section of Block Diagram, the almost simultaneity of signal analysis contributes to increase considerable performance as well.

The sampling rate control system and real-time audio capture is of great importance for sound recognition and phase difference calculation between microphones. Having this circular buffered system implemented in hardware is guaranteed that the system will function properly. This same implementation would be very difficult to perform on a normal microprocessor system due to severe time constraints.

3.3 Expands I/O

The analog inputs of the *DE-10 Nano* board will mostly be occupied by sound detectors, although 4+1 detectors would solve the problem of precisely determining the target (1 to be the reference), as a form of security, adding extra microphones does not increase the cost considerably and ensures the reliability of the signal that will be further processed.

The output of the multilateration will result in the location of the sound event, such output will be sent by the Bluetooth module to the connected mobile phone, so that the location of the event can be shown in the application to easily see in a cellphone.

IV. Functional Description

Following the natural flow of the project, in order, there is the acquisition of sound data by the A/D Converter, the Digital Filters, Neural Network and in the last the Multilateration Algorithm.

4.1 Digital Filters

At this point, the received data goes through digital filters, whose purpose is to clear the received signal from possible low frequency noises. To perform such cleaning, a high pass digital filter is applied using the expression below that characterizes a moving average filter. Since the amplitude of the A/D converter signal has an offset (DC component), this filter also removes this component, centering the signal in zero, something that facilitates the after processing.

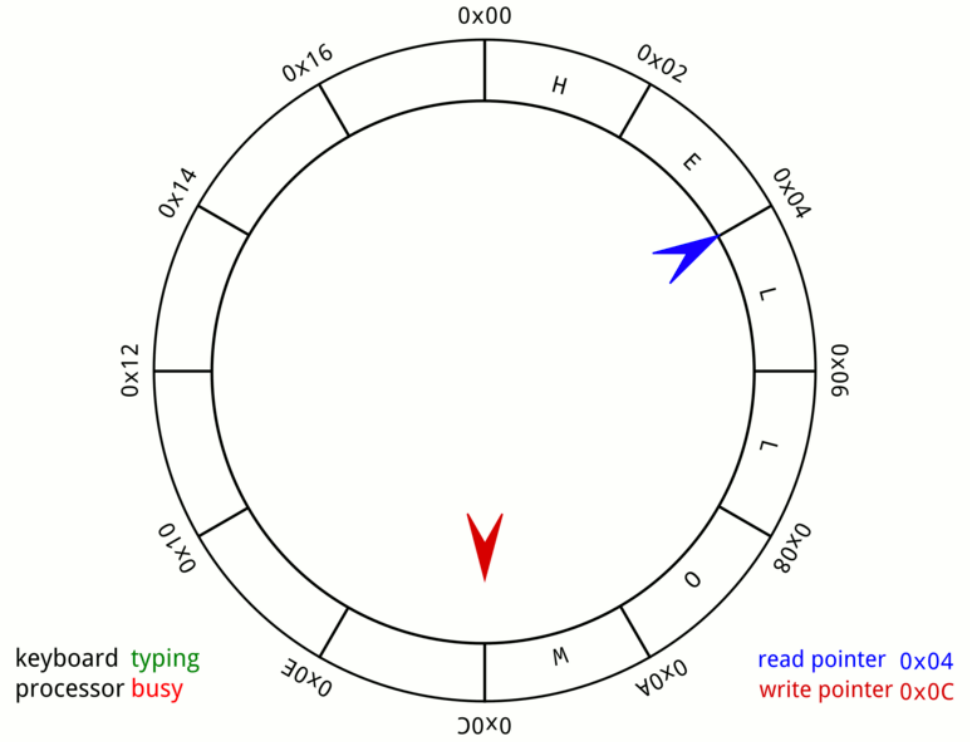
$$y[k] = \alpha(x[k] - x[k-1]) + \alpha y[k-1]$$

In the above expression, **y** represents the output vector of the filter, **x** the input vector and **α** the filter parameter, precalculated based on the sampling rate of the system and the desired cutoff frequency.

Once filtered, the data is sent to circular buffers, where there is one circular buffer for each microphone.

4.2 Circular Buffer

Circular buffers are data structures that are defined by a pivot, where the first data that was placed on the structure is located, and its tail, which is the last data, as shown in the figure below.

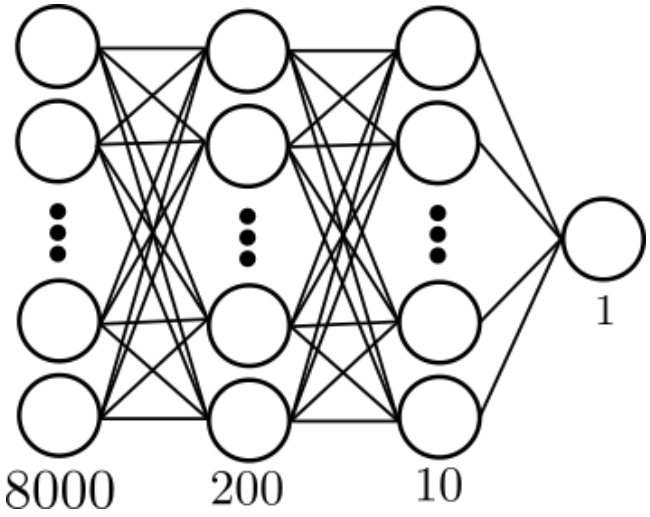


If some signal that is stored in the circular buffer pass the threshold barrier, this one is sent to the HPS, and the sound recognition is done with the help of a Neural Network.

4.3 Neural Network

Before determining the cross-correlation between the signals to obtain the phase difference between then, one must first know if these signals are really a desired sound, and for this, neural networks are used. Firstly, so that the system does not keep processing the neural network all time, a threshold based on the impulsivity of the signal is used. Since the received signal is considered impulsive, the signal is processed by the neural network and is determined whether the signal is the desired sound or not.

The neural network architecture used in the project is 4-layer MLP (input + 2 hidden layers + output), with neurons of each layer in the order: input, 200, 10, 1. The input layer quantity neurons depends on both sampling rate, feature extraction techniques and the average time that defines the signal (window time). An example of an MLP architecture for the system with 16 kHz sampling rate, window time of 0.5 s and no feature extraction is illustrated in the figure below.

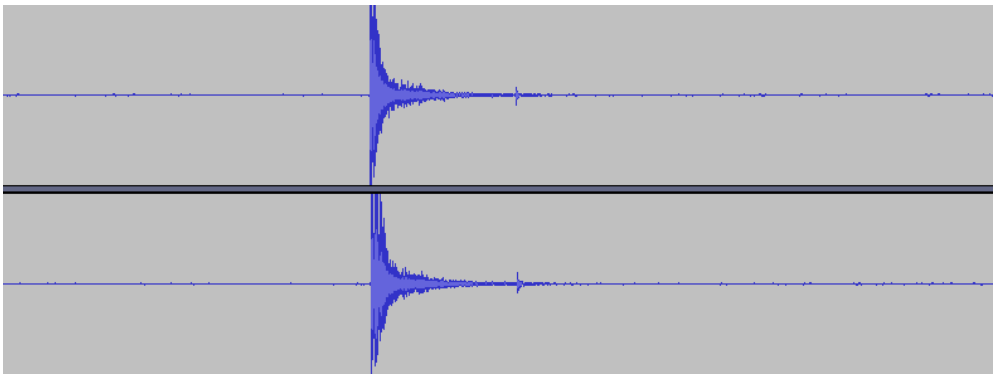


It is also important to highlight that due to the flexibility of the neural network, the system can be trained to identify other sound events (if trained correctly).

If the Neural Network identify the candidate signal as a desired sound, the system executes the multilateration algorithm, to locate the possible source of the sound.

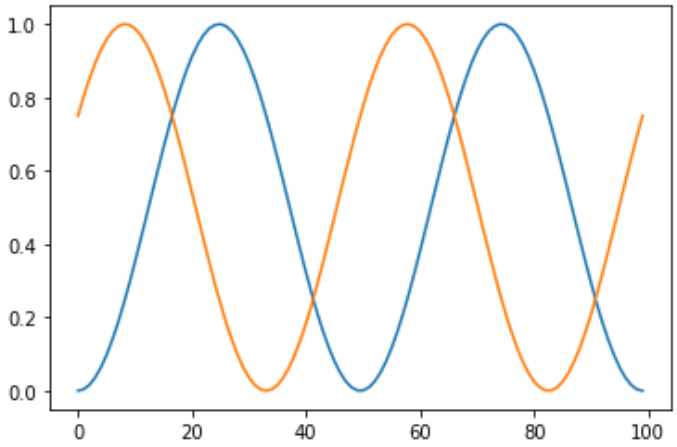
4.4 Multilateration Algorithm

In iOwlT, since the microphones are arranged at a well-defined distance, the signal is received by the microphones at different time. Two sound waves of a gunshot sound between two system microphones were recorded, and Audacity[4] software was used to show then. Is virtually imperceptible the lag between the sounds, as shown in the figure below.

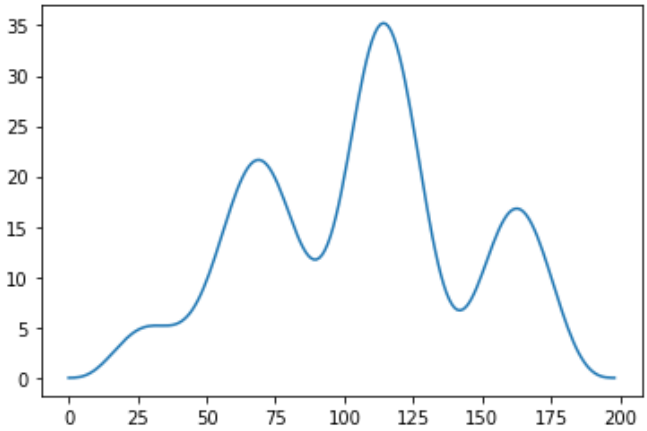


Two gunshot waves recorded with two system microphones

However, the idea behind the circular buffer is that two microphones will have their pivots shifted from a given number of samples that can be found through signal similarity analysis methods, the use of the FPGA at this time is crucial because one slight acquisition delay can lead to a considerable calculation error, since the speed of sound is high. In this case, the method used is the cross correlation.



Example of two sine signals with phase shift



The cross correlation of the sine functions

It is observed that by applying the method, a peak is obtained in the operation, and the distance from that peak to the center represents the *N* amount of samples shifted between the analyzed sine waves, and it is now possible to defined the lag time **τ** using the sampling rate SR. In iOwlT system, the SR is 16 kHz. In the example above, the phase shift is observed seeing that the peak is not in the center.

With the delays between the microphones calculated using cross correlation, it is now possible to execute the true multilateration algorithm, which consists in the solution of the following system.

$$\begin{bmatrix} \frac{2x_2}{v\tau_2} - \frac{2x_1}{v\tau_1} & \frac{2y_2}{v\tau_2} - \frac{2y_1}{v\tau_1} & \frac{2z_2}{v\tau_2} - \frac{2z_1}{v\tau_1} \\ \frac{2x_3}{v\tau_3} - \frac{2x_1}{v\tau_1} & \frac{2y_3}{v\tau_3} - \frac{2y_1}{v\tau_1} & \frac{2z_3}{v\tau_3} - \frac{2z_1}{v\tau_1} \\ \frac{2x_4}{v\tau_4} - \frac{2x_1}{v\tau_1} & \frac{2y_4}{v\tau_4} - \frac{2y_1}{v\tau_1} & \frac{2z_4}{v\tau_4} - \frac{2z_1}{v\tau_1} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} v\tau_1 - v\tau_2 + \frac{x_2^2+y_2^2+z_2^2}{v\tau_2} - \frac{x_1^2+y_1^2+z_1^2}{v\tau_1} \\ v\tau_1 - v\tau_3 + \frac{x_3^2+y_3^2+z_3^2}{v\tau_3} - \frac{x_1^2+y_1^2+z_1^2}{v\tau_1} \\ v\tau_1 - v\tau_4 + \frac{x_4^2+y_4^2+z_4^2}{v\tau_4} - \frac{x_1^2+y_1^2+z_1^2}{v\tau_1} \end{bmatrix}$$

The multilateration approximate linear system

Where τ_i is the time difference of the i-th microphone related to a microphone set as reference, **v** is the sound velocity and x_i, y_i and z_i are coordinates of each microphone related to a microphone set as reference.

V. Performance metrics / goals

In the iOwlT system, there are three important performance parameters that are analyzed:

5.1 Threshold

The first important parameter to be analyzed is the threshold, as it determines whether the neural network should judge whether the detected impulsive sound is a gunshot sound or not. Therefore, tests were made with impulsive sounds such as firework sounds, plastic bags and shooting itself.

5.2 Neural Network

The second important parameter to be analyzed is the neural network. To observe the behave of gunshot and impulsive sounds, a partnership was made with BOPE (brazilian SWAT), where it was possible to create a considerable

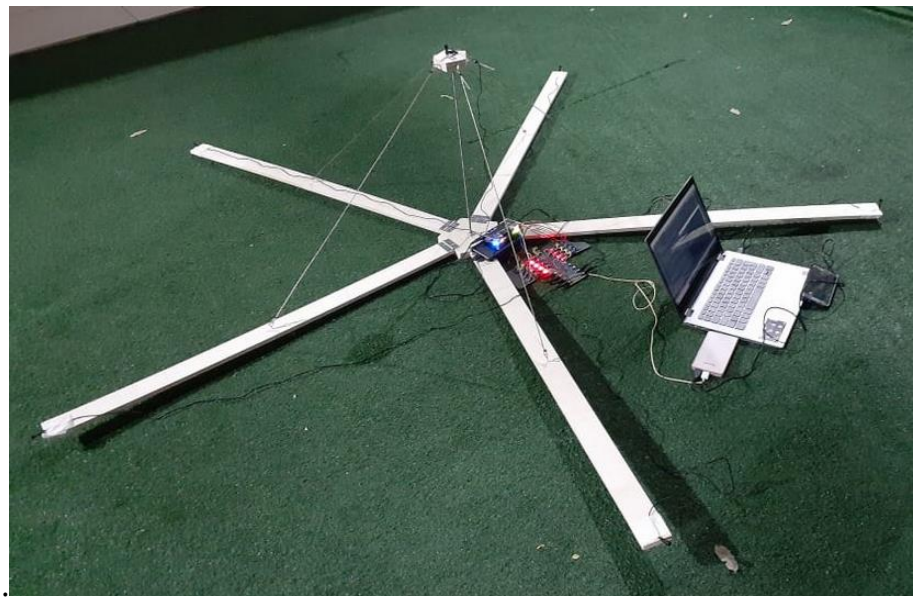
dataset of gunshots. The sounds were recorded in an open environment with a pistol.

Using Holdout validation technique, the neural network took an average perfomance of 91.38%, with the average confusion matrix shown below:

Real/Prediction	Not Shot	Shot
Not Shot	105	12
Shot	3	54

5.3 Multilateration Algorithm

The third important parameter to analyze is the multilateration algorithm. To this, an arrangment of 5-legs umbrella-shaped microphones was constructed, where each leg has a microphone in the end that will be used for the calculation of multilateration, and in the center has a microphone that has the purpose of the threshold and neural network process.



The iOwlT system

The coordinate system origin was defined at the center of the pentagon, and the y-axis as one of the legs. Thus, using a measuring tape, the actual distance values of a sound emitted were compared with the values found by the multilateration algorithm. The main idea was to analyze the system error (both distance and direction) for the four quadrants and varying distances (4 times for each point and took the average), as shown in the table:

	5m	10m	15m	20m
1^{st} Quadrant	1.63%	2.33%	3.21%	4.45%
2^{nd} Quadrant	0.79%	1.68%	2.54%	3.02%
3^{rd} Quadrant	1.01%	3.12%	3.88%	4.74%
4^{th} Quadrant	1.52%	2.16%	3.44%	5.12%

Table of angle direction system error

	5m	10m	15m	20m
1^{st} Quadrant	8.63%	10.11%	13.57%	15.12%
2^{nd} Quadrant	5.63%	7.77%	8.81%	12.24%
3^{rd} Quadrant	7.09%	9.21%	11.95%	17.47%
4^{th} Quadrant	8.00%	12.34%	17.37%	21.54%

Table of distance system error

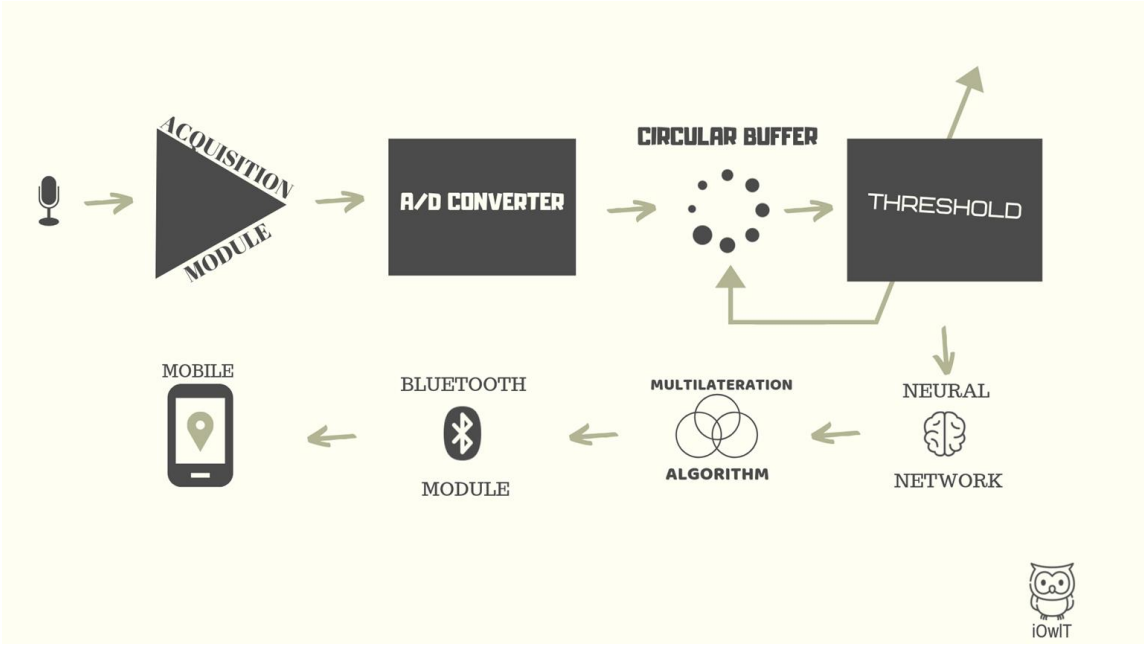
5.4 The echo problem

It is curious to observe that the iOwlT system was very good at determining the direction of the sound detected, with a 5.12% error precision in the worst case, as seen in the table. Otherwise, the error precision to the measurements of distance was bigger, this can be explained with the echo phenomenon.

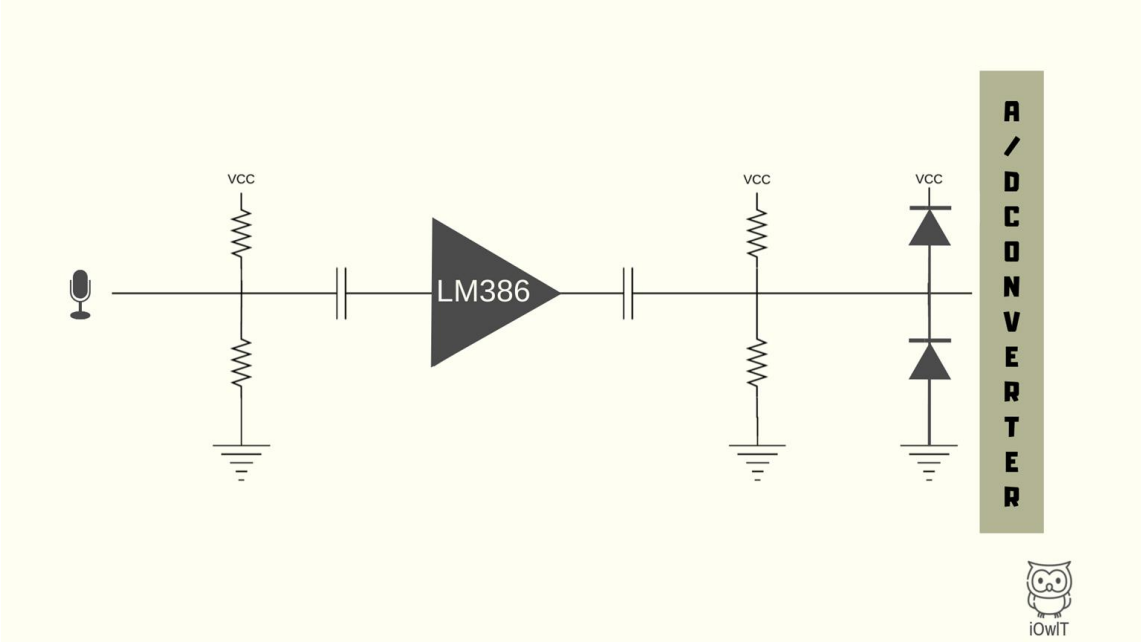
The great problem is that the echo mimetizes the sound emitted by a source in another position, creating a pseudo-source emitting the same sound, which can impose a confusion to the iOwlT system, as it uses only the phase difference of the sound to doing calculations.

One possibility to amenize the error distance due to echo problem is to treat the signal with a filter. In more open areas, the echo is not a problem, so the addition of those filters is a decision that can be done based on the application.

VI. Design Method

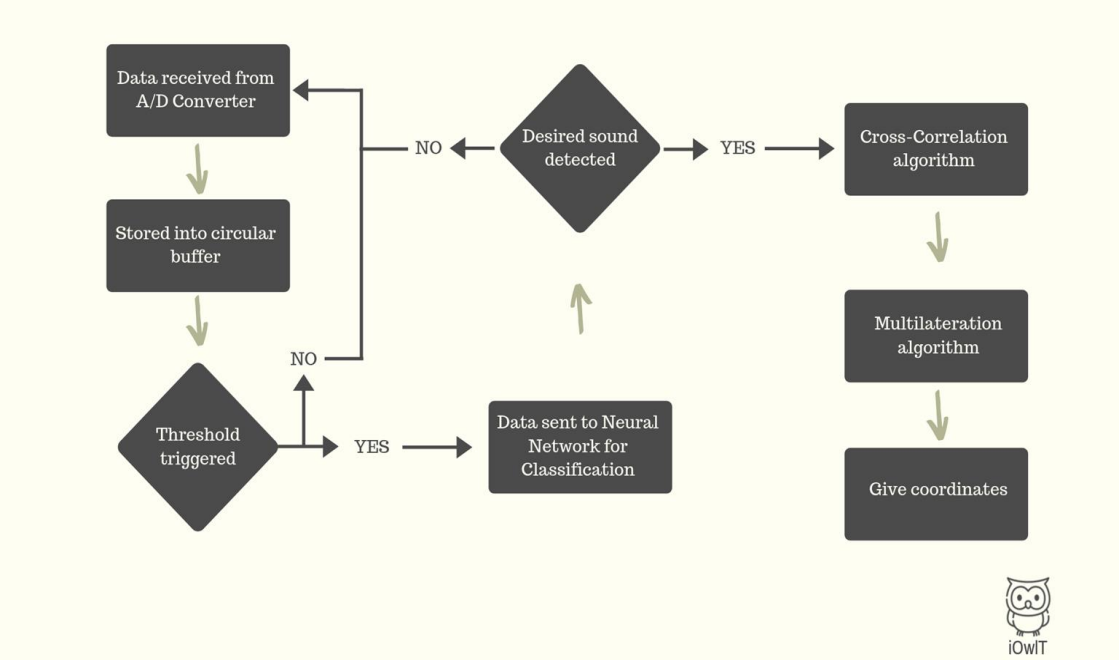


iOwlT system design scheme



Hardware circuit for every microphone

Each of the microphones used required an external FPGA circuit before they could be connected to the A/D converter. This circuit aims to polarize the microphones to allow them to operate, amplify the signal, and offset the signal to the A/D converter conversion range. In addition, there are two protection diodes that prevent a very high voltage from being delivered to the FPGA.



Software flow

VII. Conclusion

The system as a whole achieved good results, the main goal of the project is to demonstrate the technical possibility of locating sound events with sound treatment alone, and certainly iOwlT proved the technical feasibility with good accuracy.

It is important to highlight the fundamental use of FPGA to develop the system. As the system is based on the multilateration algorithm, it is extremely necessary that the phase difference measurement of the microphones be very accurate, since the speed of sound is high and therefore any error in the measurement of time leads to a considerable deviation in the final result of event position. As the phase difference between the microphones is very small, the level of accuracy in time recording of microphone signals would not be possible using a circular buffer structure implemented in software. At this point, hardware implementation was critical to good system performance, the parallelism and synchronization of hardware implemented circular buffers underlines the importance of the system.

Of course, the system can get even better, both in hardware, i.e. hardware that enables a faster A/D conversion could result in a smaller prototype as its size is directly related to time measurement accuracy, or even more logical elements,

for a more robust implementation with feature extraction and FPGA neural networking, and even software enhancements, with more BOPE visits for a wider shot dataset creating a better neural network training set.

VIII. References

[1] Knudsen, E.I. & Konishi, M. J. Comp. Physiol. (1979) 133: 13. https://doi.org/10.1007/BF00663106

[2] How Does An Owl's Hearing Work? | Super Powered Owls | BBC https://www.youtube.com/watch?v=8SI73-Ka51E

[3] Knudsen, E. Instructed learning in the auditory localization pathway of the barn owl. *Nature* 417**,** 322–328 (2002) doi:10.1038/417322a

[4] Audacity. https://www.audacityteam.org/

[5] J. Pak and J. W. Shin, "Sound Localization Based on Phase Difference Enhancement Using Deep Neural Networks"

[6] Renda, William & Zhang, Charlie. (2019). Comparative Analysis of Firearm Discharge Recorded by Gunshot Detection Technology and Calls for Service in Louisville.

[7] Mandal, Atri & Lopes, C.V. & Givargis, T & Haghighat, A & Jurdak, Raja & Baldi, Pierre. (2005). Beep: 3D indoor positioning using audible sound. 2005.