ACOUSTIC MEMS ARRAY EMBEDDED IN A SCALABLE REAL-TIME DATA ACQUISITION AND SIGNAL PROCESSING PLATFORM

BY

MARCOS DE AZAMBUJA TURQUETI

Submitted in partial fulfillment of the
requirements for the degree of
Doctor in Philosophy in Electrical Engineering
in the Graduate College of the
Illinois Institute of Technology

Approved _____
Adviser

Chicago, Illinois
July 2010

ACKNOWLEDGEMENT

It would not have been possible to write this doctoral thesis without the support, help and dedication of a number of people. First, and foremost my thesis advisor Dr. Jafar Saniie, who was fundamental in providing guidance for this research. I would like to express my sincere thanks for his patience, encouragement and friendship. My gratitude is also extended to my colleagues and friends, especially Alan Prosser for all the fruitful discussions that enhanced this work so much, and Ryan Rivera that provided greatly appreciated help with his technical expertise. Also, I would like to give my thanks to Dr. Guilherme Cardoso for his suggestions, support, and encouragement, and to Dr. Erdal Oruklu for his time and patience reviewing my work. I am very grateful for the support provided by Fermi National Accelerator Laboratory, which gave me both, the financial support and the opportunity to successfully conclude this work. I would also like to extend my thanks to innumerous people that provided indirect support for this work like: Lorenzo Uplegger, Luciano Piccoli, Sergio Zimmerman, Jeff Andresen and many others whom I can't name here, but who where nevertheless important for the successful conclusion of this work.

I would like to dedicate this thesis to my family, for their constant support of my academic ventures from the beginning to the present. Above all, I would like to thank my wife Aline for her personal support and great patience at all times. To my brothers: André and Mário, and my sister Adriana for their support and encouragement. To my brother in law, Aluisio de Andrade for his time and work helping me review this work.

And finally I would like to give thanks to my parents, Oldevir and Tânia, who encouraged me so much in life and helped me become the person I am today.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

Most sophisticated systems utilizing sound and ultrasound detection and analysis require enhanced signal acquisition and processing due to the increasing demand for real-time, precise, and scalable sound and ultrasound systems. Applications such as 3D source localization, tracking, sonar, and medical imaging require powerful systems that can deliver, in real-time, the complex calculations required by state of the art processing algorithms. Also, in many of these applications the use of multiple sound, or ultrasound transducers that are spatially distributed, or are arranged in arrays is a requirement. Microphone arrays are advantageous compared with single microphones in the sense that they can use the physical spatial information of sound propagation to provide a more accurate picture of the incoming acoustic waves, and therefore yield precise information about its source. Spatially distributed transducers create the need for systems with distributed data acquisition and processing features. Furthermore, to enable the system to support growth and adaptation (for example, to accommodate new developments in signal processing algorithms), the implementation should be flexible without compromising performance.

There are many applications in other fields that make use of transducer arrays such as RADAR, radio telescopes, or tracking systems for high energy physics experiments, they all present similar challenges.

In order to address the above concerns, a novel, flexible, and expandable data acquisition system architecture with a real-time signal processing capability was developed. The design of the architecture has been motivated by careful consideration of the common requirements shared by the many applications described above. The flexible

and scalable nature of the architecture makes possible a wide variety of system solutions that are based on a common set of system components (including hardware, configware, and software). Associated with the data acquisition architecture a specially designed acoustic array was built and embedded on the system to solve problems related to acoustic source localization, and to prove this new data acquisition concept. The architecture presented on this work is called CAPTAN (Compact And Programmable daTa Acquisition Node), and the acoustic array embedded on the system is called AMA (Acoustic MEMS Array).

The CAPTAN architecture is a distributed data acquisition and processing system that can be employed in a number of different applications ranging from test stand data acquisition systems to high performance parallel computing nodes. This architecture has the unique feature of being highly expandable, interchangeable, adaptable, and with a high computational power inherent to its design.

The AMA array conforms to the CAPTAN architecture, and is an acoustic array that employs sound, or ultrasound transducers in two dimensions, fully capable of beam steering on both horizontal and vertical planes. The AMA array takes full advantage of the CAPTAN architecture, and in this way it is fully scalable and planned to work on real-time applications. This unique combination gives the system many possible configurations to be deployed using distributed sensors, composed arrays and 3D arrays where applications such as sound source localization, and acoustic imaging are just a few of many where the system can be used.

This work will also present this system fully integrated, and performing as a multi-source localization and separation system using an algorithm capable of isolating

multiple sound sources and provide estimation of its position. The algorithm that makes it possible for the system to work in this fashion, its capabilities, and the use of unique hardware features of the architecture will be described in detail.

The system ability to separate and locate acoustic sources was tested in a series of specially design experiments where the number of acoustic sources, signal complexity and geometric disposition were the parameters used for judging the performance of this system.

CHAPTER 1

INTRODUCTION

1.1 Objectives

The fields of real-time scalable data acquisition, and high performance transducer arrays are strongly interconnected as most of these transducer arrays demands sophisticated readout systems in order to perform properly. Sound and ultrasound systems such SOund NAvigation and Ranging (SONAR), Sonography, Speech Recognition [Adc96,All00], Robotic Sensing and Sound Source Separation are a few of many systems that can employ transducer arrays and require real-time high speed and expandable data acquisition systems [Nak02,Adc96]. Similarly, arrays that operate within the electromagnetic spectrum or with particle detection are employed in fields such as RAdio Detection And Ranging (RADAR), Positron Emitting Tomography (PET) and Magnetic Resonance Imaging (MRI), require sophisticate data acquisition systems with comparable performance requirements. For these system characteristics such as speed, scalability and real-time signal processing are paramount, demanding highly developed data acquisition and processing platforms [Wei04]. Successfully integrating acoustic arrays with generic data acquisition systems is challenging due to the wide variety of requirements regarding speed, signal processing capabilities and scale that acoustic arrays can present.

Microphone arrays are capable of providing spatial information of incoming acoustic waves, having the ability to capture key information that would be impossible to

acquire with a single microphone, this extra information comes with a price of increased system complexity.

The design of microphones arrays faces several challenges such as number of detectors, array geometry, reverberation, interference issues and signal processing [Buc05]. These factors are crucial to the construction of a reliable and effective microphone array system.

A microphone array is also a challenge for signal processing since a large number of detecting elements will generate large amounts of data to be processed, furthermore applications such as sound tracking and sound source localization, require complex algorithms to properly process the raw data [Ben08]. Challenging environments such as multiple sound sources moving with background noise are especially difficult to deal with when processing in real-time is required, demanding powerful signal processing that the CAPTAN system can provide. It is also important for such a system to be easily scalable since, as demonstrated by [Wei04], and [Ben08], the performance of a microphone array increases linearly with the size of the array.

The system presented in this work is intended to provide a generic data acquisition system that is flexible and expandable, with a powerful real-time signal processing capability which allows it to be interfaced with a wide variety of transducer and transducer arrays. This work will also present the design of an acoustic microphone array embedded on the data acquisition system to solve problems related to acoustic source localization, and to prove this new generic data acquisition concept. The data acquisition and processing architecture presented on this work is called CAPTAN

[Tur08] (Compact And Programmable daTa Acquisition Node) and the acoustic array embedded on the system is called AMA (Acoustic MEMS Array).

The CAPTAN architecture is a distributed data acquisition and processing system that can be employed in a number of different applications ranging from single transducers interface to multi-transducer arrays data acquisition and high performance parallel computing [Riv08]. This architecture has the unique features of being highly expandable, interchangeable, adaptable and with a high computation power inherent to its design.

The AMA array was design to conform to and take advantage of the CAPTAN architecture. It is an acoustic array that employs sound, or ultrasound transducers arranged in two dimensions, fully capable of beam steering on both horizontal and vertical planes. This unique combination of data acquisition system and, the acoustic array gives this particular system many possible configurations such as distributed acoustic arrays, composed acoustic arrays and 3D acoustic arrays where applications such as sound source localization and acoustic imaging are just a few of many that can take advantage of this system.

It is also the objective of this work to present the algorithm that makes this system perform as a multi-source localization and separation platform. This algorithm is capable of isolating multiple sound sources and provides estimation of its distances and position by using independent component analysis [Hyv00]. The algorithm's capabilities and performance as well as the unique CAPTAN architecture features that the algorithm utilizes will also be described in detail.

1.2 Thesis Summary

Chapter 2 presents a literature review on data acquisitions system as well as acoustic arrays basics with a special focus to Independent component Analysis techniques. This chapter also includes the relationship between acoustic arrays and data acquisition systems and the main requirements that this specific type of array demands from a data acquisition system. Beamforming is also presented as a tool proper to be applied to the source location algorithm that this work is proposing. Chapter 3 will present the current state of the CAPTAN system, its architecture and capabilities as well as its possible deployment to acoustic transducer arrays. In chapter 4, the AMA system is presented, design considerations, array performance, capabilities and its and integration to the CAPTAN system are discussed. Chapter 5, the results of this work is detailed where special focus is given to the tests performed with the system and the algorithms devised for this work. Concluding this work, the bibliography consulted is presented.

CHAPTER 2

LITERATURE REVIEW

2.1 Data Acquisition Systems

Data acquisition is the collecting, processing and storing of data from transducers, or other measurement systems. Prior to the use of computerized systems data was often acquired manually or mechanically, and recorded on paper. The advent of electronic based data acquisition allowed users to perform much more complex measurements and processing as well as to store and retrieve data electronically. Most modern electronic equipments have some sort of data acquisition system (DAQ) that enables them to perform its task.

The field of data acquisition is very broad and rich, and systems performing data acquisition can range from tiny systems design to readout temperature information from a single thermometer, to huge data acquisition systems such as those employed to readout particle detectors tracking systems at the Large Hadron Collider at CERN, Switzerland [Men07].

Generally, data acquisition can involve transducers, signal conditioning, analog to digital conversion, signal processing, communications, data storage, and data displaying. The understanding of information theory and its resulting sampling theorem as well as signal processing are very important to the successful design of a data acquisition system [Jer79].

Data acquisition systems can be classified in many different categories, usually the defining characteristics are size, speed, complexity or deployment topology. Data

acquisition size can range from a single transducer to millions of channels and it is one of the main characteristics of a data acquisition system, systems can also have a dynamic size where the number of transducers on the system at any moment is allowed to increase or decrease.

Another very important characteristic is speed, usually the type of transducer utilized will dictate the speed requirements for the data acquisition system, and it can vary in a very broad range, from days to Pico seconds. Closely related with the data acquisition speed is the real-time capability that some systems can have. A real-time system can be very demanding in overall timing requirements, from the point that the signal is digitized, passing through signal processing and data visualization, everything must have the minimal possible delay. Usually a system that requires real-time capabilities will have a signal processing, data visualization, or control feedback that is required to perform with minimal delay related to the real physical process that is being captured or executed.

Data acquisition complexity usually derivates from a combination of factors such as speed and size, also, parts of the system such as signal processing can by itself make a data acquisition system very complex. The deployment topology is related to how the transducers are physically distributed. The transducers, and the data acquisition system can be deployed in the same physical location with transducers embedded on the data acquisition system or physically very close, or they can be deployed along a very wide area, in this case the data acquisition system will be distributed and its parts can be very far apart requiring communication transceivers.

Data acquisition systems can also have at its disposal control systems data will act on transducers or other systems. Figure 2.1 presents a common data acquisition setting where the data acquisition system acquires data through a sensor array and at the same time act on a transducer.

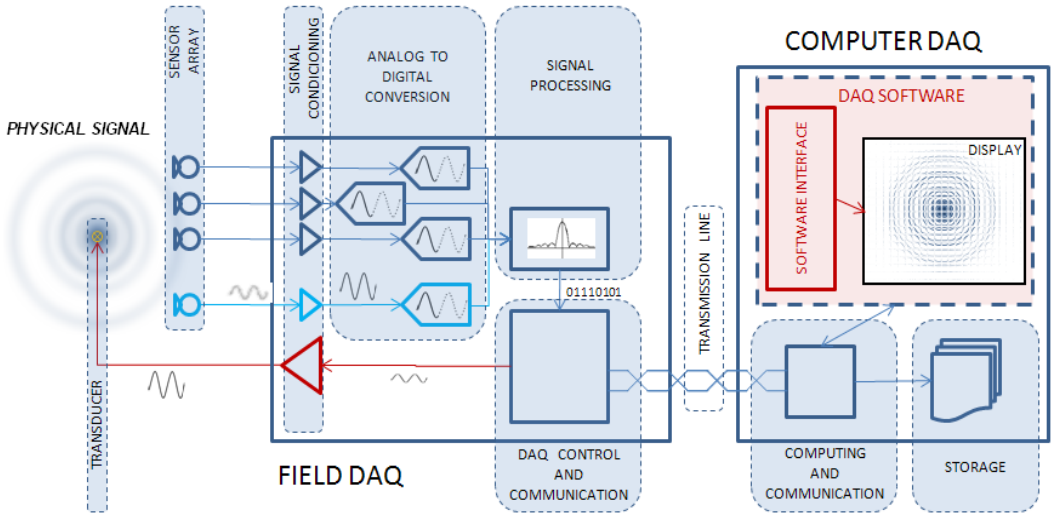

Figure 2.1. Typical data acquition system.

In the same figure it is also possible to observe another important property that a data acquisition can have: topology distribution, where part of the data acquisition is with the sensor array and transducer, while another part is far, connected through a communication channel. Many data acquisition systems have this characteristic, usually the part that is not with the transducer is the part that will store, analyze, and present the data. It is also possible to observe on Figure 2.1 the data acquisition system software. Most modern data acquisition systems will have a software part running in a computing system with the mission of providing the user with means to interface with the system [Par03]. Such interface can include access to data visualization, system control, data logging and data analysis. Typical applications that can use the above mentioned

architecture are active RADAR [Hel01], active SONAR, medical ultrasonography, MRI systems and ultrasound material inspection systems [San81, Oru04].

2.2 Acoustic Transducer Arrays

An acoustic transducer array consists of a set of transducers arranged in one, two or three spatial dimensions; its elements can be clustered in a close geometrical arrangement or spatially distributed across great distances. No matter in what configuration the array is set the objective is the same; to sense acoustic pressure waves from multiple points in space. The information captured needs to be processed, the main objective of signal processing in an acoustic transducer array is the estimation of parameters of interest, and depending on the application it can use the spatial-temporal and or frequency information available at the output of the array.

There are many issues that must be taken into consideration when designing an acoustic transducer array, problems such as reverberation, noise, number of elements, and array spatial size need to be carefully considerate [Ben08, Wei04]. It is fundamental to tackle these issues in order to construct a reliable and effective acoustic transducer array system, choosing the right design strategy for the development of the system can also include implementing a practical real-time signal processing integrated system.

Reverberation is the persistence of acoustic waves in a determined region after the passage of an acoustic wave. Reverberations are created when certain geometric conditions arise that will cause large number of echoes to form. Reverberation can occur in the sensor array itself and it can greatly degrade the quality of the signals produced by an acoustic array.

Noise in an acoustic array can be driven by the electronics, specially the conditioning stage of the data acquisition, but also the transducers themselves can be a source of noise. Noise mitigation in an acoustic array is done by careful design of the data acquisition system and proper layout of the acoustic array board.

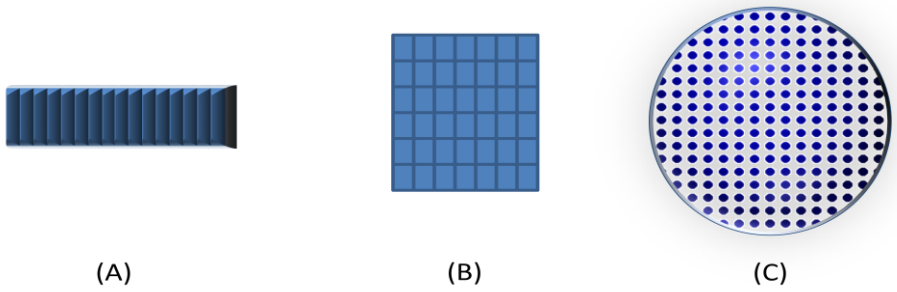

Figure 2.2. Array geometry. (A) Linear array, (B) two dimensional array and (C) three dimensional spherical array.

The geometry of the array can play an important part in the formulation of the processing algorithms [Ben08]. Different applications require different geometries in order to achieve optimum performance. In applications such as source tracking the array geometry is very important to determine the performance of the system [Nak02]. Regular geometries where sensors are evenly spaced are preferred in order to simplify the algorithm development. Figure 2.2 illustrates typical configurations in which an acoustic array can be deployed. Linear arrays are usually applied in medical ultrasonography, planar arrays are often used in sound source localization, and three dimensional spherical arrays are most frequently used in sophisticated SONAR applications. In other applications such as source separation, the geometry of the array is not as important as transducer characteristics such as dynamic range and transducer aperture. The spatial size

of an array is usually determined by what frequency the array will operate and what kind of spatial resolution the application using the acoustic array requires [Wei04].

After the array transforms the acoustic pressure into electric signals, a data acquisition system is necessary to condition, process, store, and display the signals produced by the array. Data acquisition for an acoustic array can be very challenging due to innumerable reasons, but most of the issues are co-related with the design of each specific array such as type of transducer, number of transducer, and array geometry. The application also plays an important role in defining the needs of the array for signal processing capabilities or real-time operation requirements.

Currently the main applications that require acoustic transducer arrays are multiple source localization, source tracking, speech processing [Adc96], echo reduction, noise reduction and SONAR.

Some of the above mentioned applications require advanced techniques to separate mixed signals and other applications require the ability to locate the source of the signals and a few applications require both. This work requires both and implements them by using beamforming for source location and Independent Component Analysis for signal separation.

2.3 Beamforming

Beamforming is a technique used in transducer arrays for signal spatial directivity transmission or reception. This spatial directivity is achieved by the use of interference patterns to change the angular directionality of the array.

When used for transmission of signals the transmitting transducers will be steered, where the amplitude and phase of its individual elements will be controlled to act as one through patterns of constructive and destructive interference. The resulting wave front will have the energy of the array concentrated on the intent direction.

The same way that beamforming is used for concentrating energy to a preferential direction of propagation, when used with transducers receiving waves the array can be steered to increase sensibility to preferential angular direction.

There are generally two categories of beamforming; static and adaptive [Hod80,Cam99]. Static beamforming involves using a fixed set of parameters for the transducer array. The array acts the same independent of changes in the scenario where it is immersed. In this case the individual gain and phase of each element will be dictated only by its geometry and the static directionality requirements of its application. Adaptive beamforming on another hand can adapt the parameters of the array in accordance with changes in the situation in which the array is immersed. Adaptive beamforming can perform substantially better than static beamforming in many situations, noise rejection being one important case, however, adaptive beamforming can be heavily computationally demanding and not worth of implementing on applications that have little gain in employing it.

Transducer arrays that make use of beamforming techniques are also know as phased arrays and can be classified accordingly to the beamforming technique it is using. Regardless of the beamforming technique used, the phased array can also be classified in two categories; time domain or frequency domain beamforming.

Time domain beamforming is based on delay and sum operations [Hod80], delaying the signal from each array element by a certain amount of time and then adding them. Other operations such as multiplication can also be employed during this procedure in order to highlight the desired wave patterns or to insert zeroes and therefore perform filtering.

Frequency beamforming decomposes the incoming signal in frequency bands, generally two different techniques can perform this separation, Fast Fourier Transforms, or multiple filter banks. After the signal is decomposed in different bands time domain beamforming is then applied to each individual bands. This technique allows the phased array to have different directivity toward different frequencies. As with adaptive beamforming this technique will increase the system's computation demands. Applications such as wireless communications can profit from this kind of technique. Both types of beamforming algorithms can be employed either with static or adaptive beamforming.

Beamforming can be used for acoustic or electromagnetic waves, and is currently widely deployed in transducer arrays that make use of either wave types. When employed for acoustic arrays, beamforming techniques are used in applications such as sound source localization, sound tracking, SONAR, ultrasonography and ultrasonic material inspection. When employed to electromagnetic arrays it can be employed in RADAR, telecommunications, medical systems and radio astronomy.

2.4 Independent Component Analysis

Sometimes acoustic signals coming from different sources are mix and it is necessary to isolate them. In applications such as video conference and cell phones, it is necessary to have an algorithm capable of performing such task. Independent Component Analysis was the algorithm of choice for this research due to its versatility and generic use.

Independent component analysis or ICA is a mathematical technique used for extracting hidden parameters that underlie in sets of random variables or signals.

ICA is a type of blind source separation method and common inputs sources are signals originated from audio, images or telecommunications [Sto04].

This technique is based on the assumption that signals from different sources are statistically independent and statistically independent signals can be extracted from mixture signals. Therefore, the condition of source statistical independence must be fulfilled for the successful implementation of this technique.

ICA defines a model for the observed data that requires a large database of samples in order to establish the necessary statistics. The model assumes that the data variables are linear combination of random variables, the random variables are assumed to be non-Gaussian and independent. The goal then becomes to find a transformation in which the components are as statistical independent as possible from each other.

This technique is related to methods such as principal component analysis [Nor06] and factor analysis. The main distinction between ICA and these techniques is that while ICA finds a set of independent sources, principal component analysis and factor analysis finds sets of signals which are uncorrelated. This means that ICA recovers

the original sources while the other two methods only find sets of the signals that can be locally uncorrelated but not necessary globally independent.

A typical example of deployment of the ICA technique is the problem of source separation. When there are mixtures of simultaneous speech signals that have been picked up by a microphones array it is desirable to have the original signals isolated. One way of doing this is through an algorithm implementing ICA. The algorithm would accumulate statistics of the incoming signals, and then analyze them, trying to isolate non-Gaussianity and independent characteristic that the signal can present. Such an algorithm is computationally intensive since it must accumulate and go through the signal samples performing complex operations. In theory the ICA algorithm can distinguish as many sources as independent variables are generated by the linear combination of the sources. In practice this means that we need at least the same number of sensors as sources to correctly separate the signals.

In order to illustrate the way this algorithm works let's consider a simulation where two independent and non-Gaussian signals are being emitted by an audio system, where one is a saw tooth wave, let's call it signal A and the other is a sine wave, let's call it signal (B). The overall arrangement of the experiment can be observed on Figure 2.3, while signals (A) and (B) can be referred by Figure 2.4.

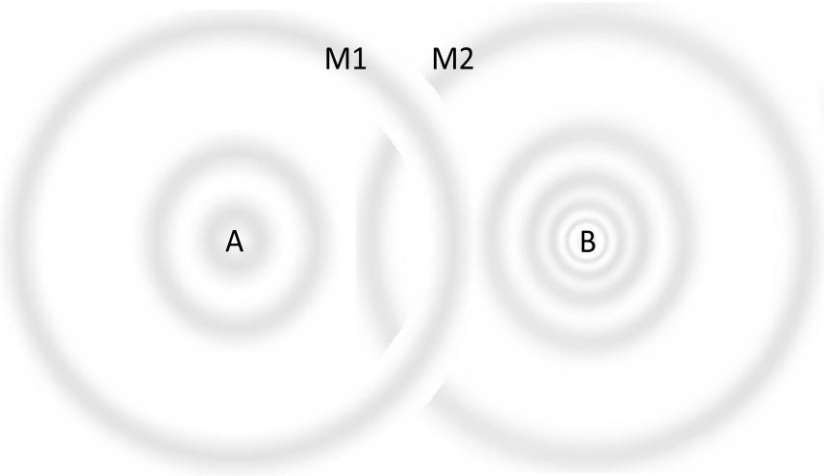

Figure 2.3. The geometry of the experiment. (A) and (B) are the audio sources while M1 and M2 the receivers of the signals.

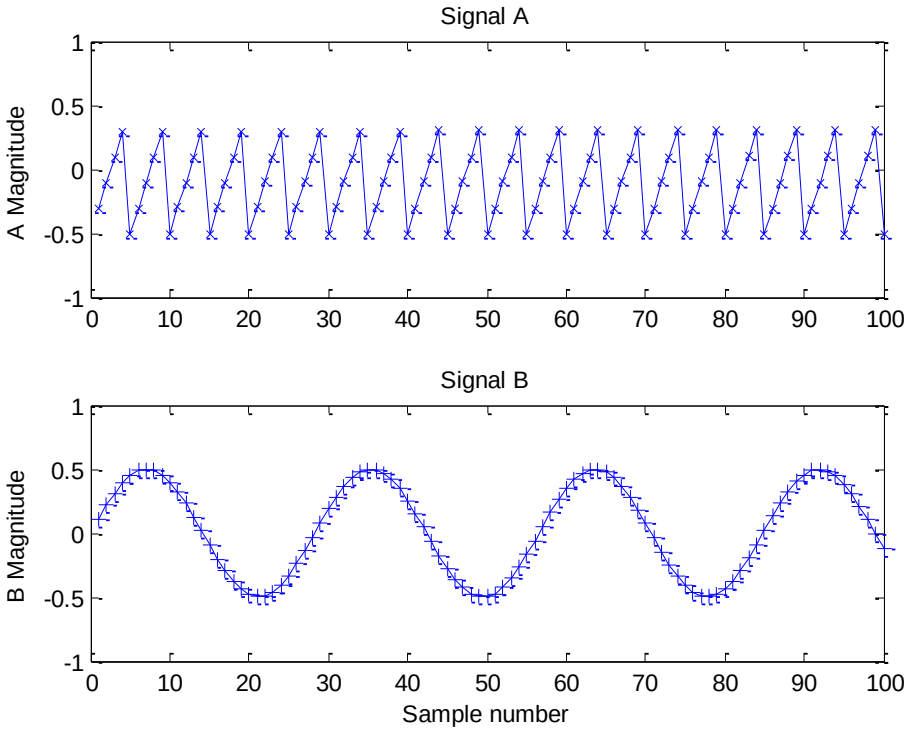

Figure 2.4. Signals emitted by the theoretical audio system. On the top Signal A and on the bottom signal B.

As these signals propagate through the air they will combine, and for the sake of illustration let's assume that they went through a linear combination and were captured by two microphones set at different positions in space that captured the combined signal.

$$\begin{cases} \boldsymbol{M_1 = 0.2A - 0.5B} \\ \boldsymbol{M_2 = 0.9A + B} \end{cases} \tag{2.1}$$

The signal captured by the first microphone is called M1 and by the second one M2. These signals were defined by the linear transformation above and can be better visualized on Figure 2.5.

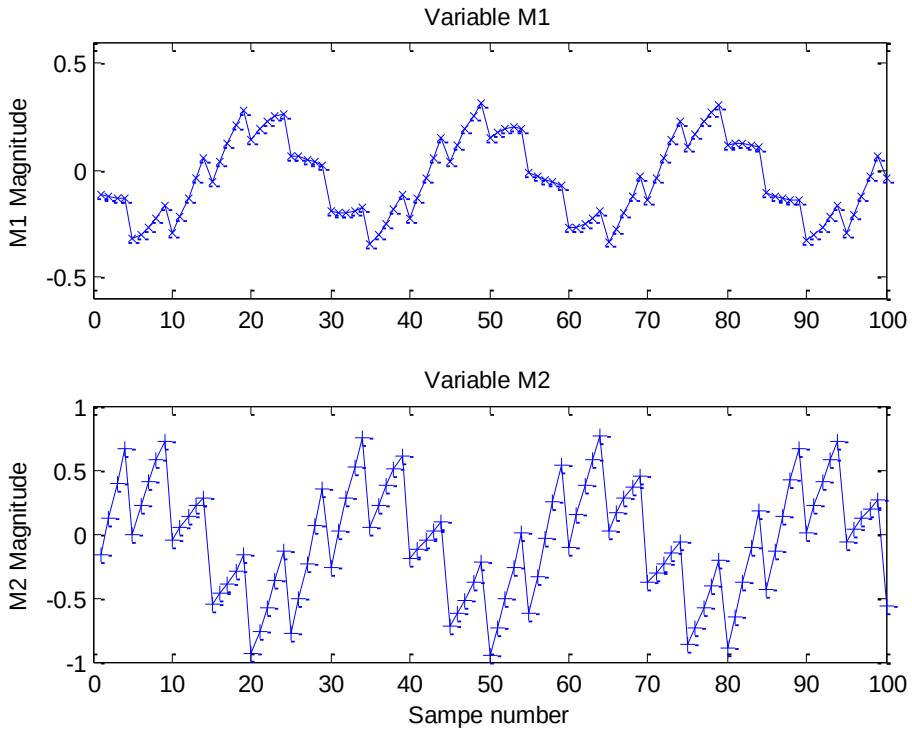

Figure 2.5. Signals resultant from the linear combination they went through. On the top the signal captured by microphone M1 and on the bottom the signal captured by microphone M2.

As the figure illustrates the signals are now quite different from the original signals. In fact one of the most important differences is that any linear combination of two independent random variables will be more Gaussian than the original variables themselves. This assertion comes from the central limit theorem [Ber41] and it is one of the bases for the ICA algorithm.

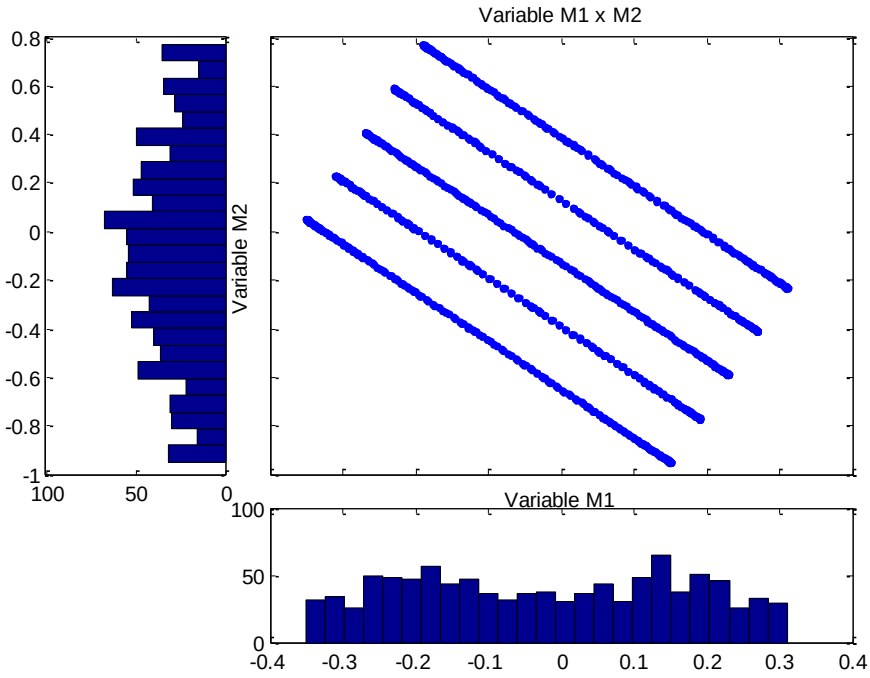

Figure 2.6. Histograms of variables M1 and M2, and the plot of M1 versus M2 where M1 is the abscissa and M2 the ordinate.

The histogram of the variables M1 and M2 is plot on Figure 2.6. Having the central limit theorem in mind it's already possible to infer that if there is a way to produce new variables derived from M1 and M2 that are less Gaussian than M1 and M2 we will be one step closer to the original signals.

The first step in the application of the ICA algorithm is the preprocessing. Preprocessing has as its goal to prepare the data for the efficient application of the algorithm and usually is composed of two steps, centering and whitening.

Centering is the first preprocessing of the data for the ICA algorithm and it's simply center the incoming variables, in the present example the variables M1 and M2 are already centered, but if they were not, centering would consist in subtracting the mean vector for each variable making them a zero-mean variable.

After centering the ICA algorithm usually requires the whitening of the data. The whitening process decorrelates the input signals as much as possible and equals their variance to one. This is required to get one step closer to the original signal, since the linear transformation added correlation between the variables M1 and M2.

The whitening process is a linear transformation where the covariance matrix of the input signals is equalized to the identity matrix. In this case it will be given by

$$\boldsymbol{cov}(\boldsymbol{WM}) = \boldsymbol{I} \tag{2.2}$$

Where (W) is the new variable with the whitened data, (M) the variables containing the data and (I) the identity matrix. The transformation can be accomplished by several different methods; the most used being Principal Component Analysis [JOL02], Figure 2.7 shows the new variables (W) obtained by the whitening of the variables (M). The variance of both axes is equal after the whitening, and the data projection on Figure 2.7 is zero. That means that to obtain the original data all that is left to do is rotate the (W) array till it matches the original values.

The core of the ICA algorithm is what performs this axis rotation, this is not a trivial process, and there are two major processes that must be tackled before this can be achieved. The first is how much you need to rotate the axis and the second is how you rotate it. The two processes are solved by iterative algorithms running in parallel and they are achieved by rotating the axis and minimizing Gaussianity of the projection at the same time. The two most popular methods used to give an indication of the Gaussianity of the distribution are Kurtosis, and Negentrophy. The reason for minimizing the Gaussianity is that when random variables are linearly combined the result will be more Gaussian that the original variables, therefore by minimizing the Gaussianity of the transformed variables we get closer to the original variables.

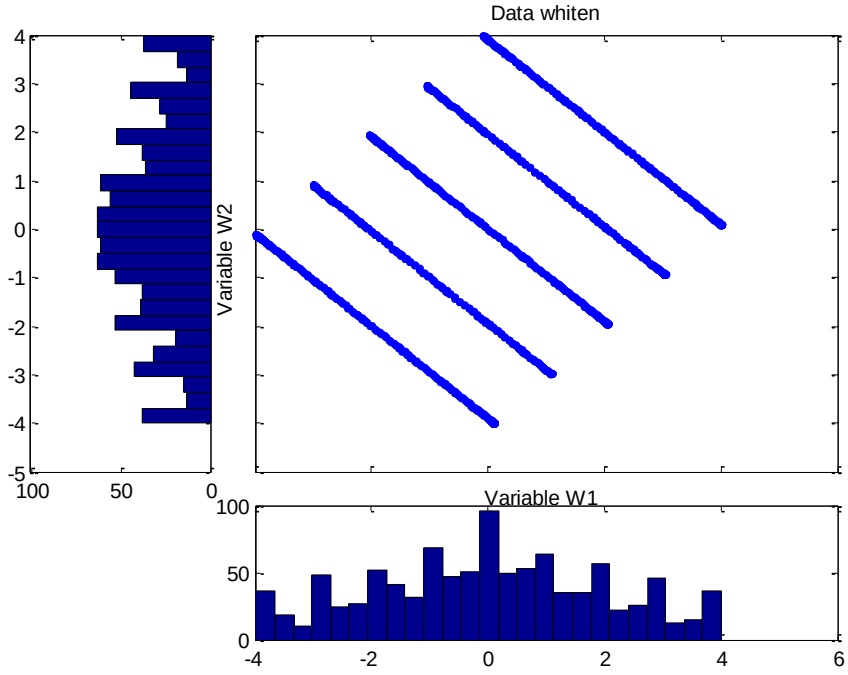

Figure 2.7. Data whitening. Histograms of variables W1 and W2 and the plot of W1 versus W2 where W1 is the abscissa and W2 the ordinate.

This process was applied to the array (W) using the ICA algorithm [Hyv00], and two new variable where generated (AR-Recovered A) and (BR-Recovered B). The results can be observed on Figure 2.8 and Figure 2.9.

Although a very powerful tool the ICA algorithm when applied to complex sets of acquired data has many limitations. The first one can be observed when comparing Figure 2.4 with Figure 2.9, the relative magnitude of the signals is lost. Also, the algorithm does a blind separation and has no idea which signal was (A) and which one was (B). It is also challenging sometimes for the algorithm to converge since it is an interactive method it can be trapped on local minimums and provide non optimal solutions.

Another limitation of the algorithm that is especially important to this work is the fact that the algorithm will generate as many solutions as the number of variables it is fed.

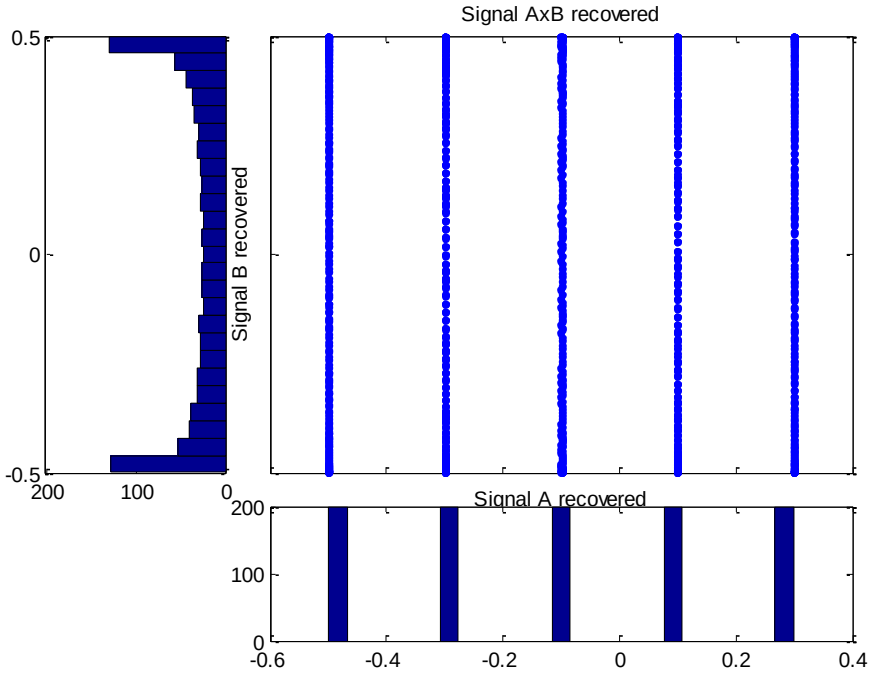

Figure 2.8. Histograms and plot of the recovered signals. Histograms of variables AR and BR, and the plot of AR versus BR where AR is the abscissa and BR the ordinate.

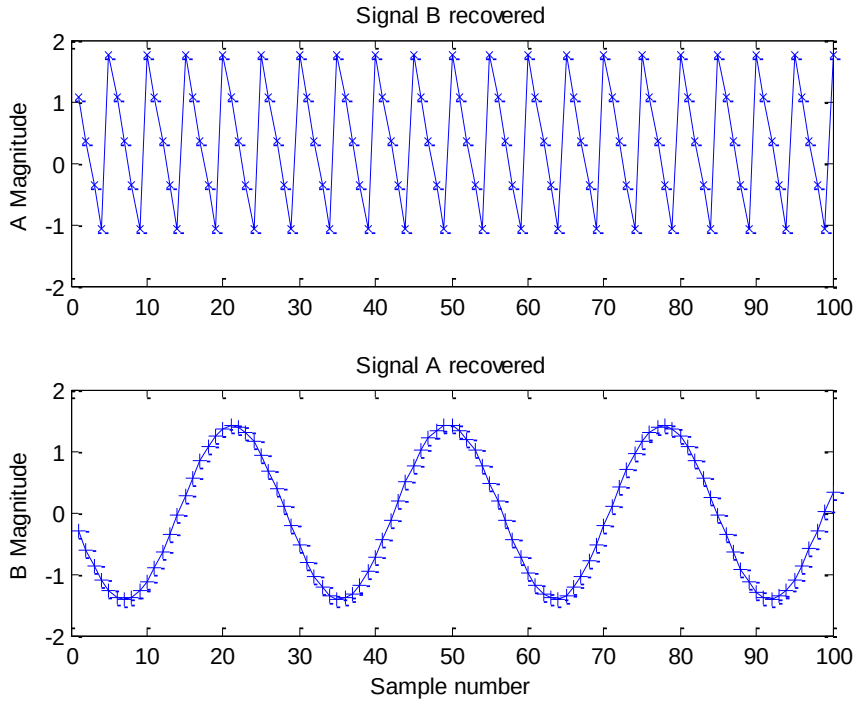

Figure 2.9. Signals recovered by the ICA algorithm. On the top, the recovered Signal BR and on the bottom AR.

Therefore even if there is only one signal present in a real event, the algorithm will generate as many variables as the number of inputs it was fed. In an ideal case the variable would be all identical, but in real applications they will differ due to noise and non-linearity, and then special application dependent techniques are necessary to eliminate false results.

Also, the algorithm tolerates at most one Gaussian variable otherwise it can't converge, Figure 2.10 illustrates a case where there are two Gaussian variables, the algorithm would keep rotating the axis but as the distributions are symmetrical the Gaussianity would always be the same.

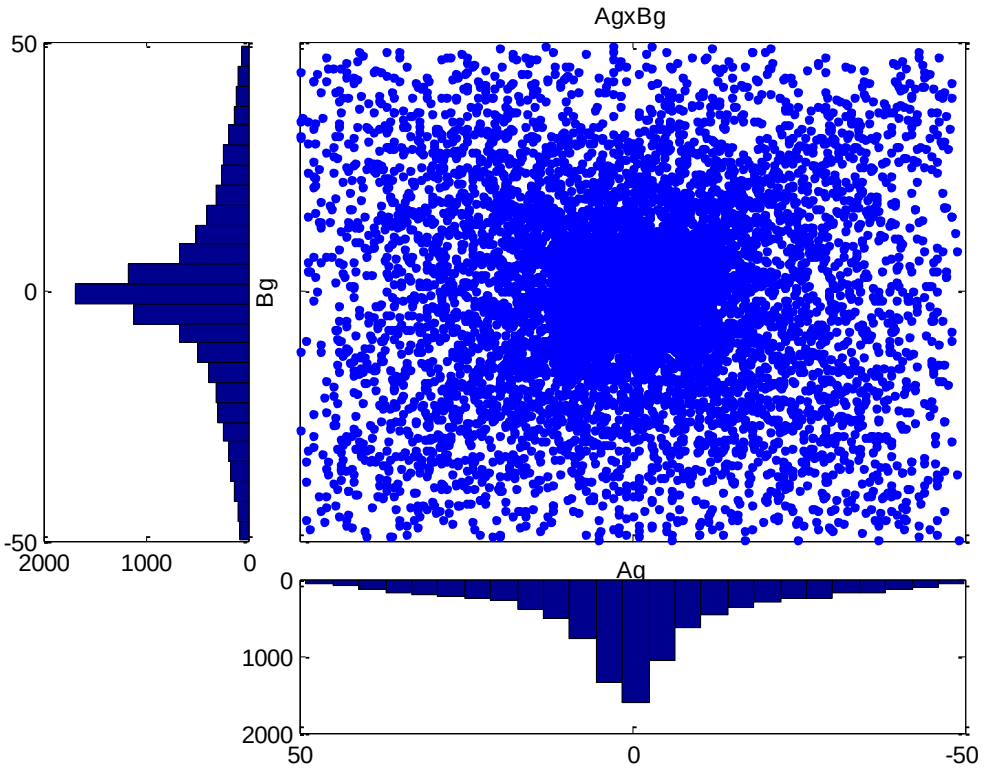

Figure 2.10. Gaussian variables can't be recovered due to symmetry.

If the user is well aware of the limitations of the algorithm it can be a very useful tool for certain types of applications.

2.5 Conclusions

In this chapter, a brief review of basic concepts regarding data acquisition systems and acoustic transducer arrays was presented. Initially an overview of data acquisition systems was presented with focus on the basic parts of a data acquisition system and the diverse properties that a data acquisition system can provide to different applications, where it is demonstrated that different applications may require very different capabilities from a data acquisition system. Special focus was given for the necessary characteristics that a data acquisition system may need to interface with a transducer array.

This chapter also presents a review on acoustic transducer arrays; special attention was given to the main design challenges of this kind of array, and its many possible applications. Also, the integration of these two systems was briefly discussed where the flow of information from its source until its end was illustrated. Finally a review of beamforming and Independent Component Analysis was provided for a better understanding of these techniques as they are utilized on this work.

CHAPTER 3

THE CAPTAN ARCHITECTURE

3.1 Introduction

The CAPTAN system is a distributed architecture based on structures known as system nodes [Tur08]. A node is a stack of boards connected together by a vertical bus in which every board in the same node has access to the vertical bus, and therefore are accessible to each other. There are no limits to the number of nodes that can work together in a system since nodes are networkable, the only limits are for the number of boards that an individual node can support. The system architecture can be separated in three different layers, the node layer, the network layer, and the application interface layer.

The CAPTAN node architecture is the part of the CAPTAN system known as node layer and it supports two types of data paths, namely, the intra-node and the inter-node data paths. The intra-node communication is achieved by means of the vertical bus that connects all the boards in the same node. The inter-node communications can be implemented by two different paths, the horizontal bus and the Gigabit Ethernet Link (GEL) [Iee99]. Another key feature of the node architecture are the core boards which provide the backbone of the node system, and form the central part of this hardware. In addition to the core boards (also known as primary boards), there are secondary boards and user boards.

The network layer is formed by commercial routers, network cards, computers and the specially design GEL card that is the CAPTAN's node communication gateway.

The network layer size has no limits and can be as big as the user application demands. The communications protocol used on the network layer is Ethernet UDP and every CAPTAN node will have at least one IP associated with it. Since CAPTAN nodes can support up to ten GEL boards, a single CAPTAN node can have up to ten IPs associated with a particular node.

Finally, the software [Riv08] is an integral part of the CAPTAN system; the CAPTAN Network Manager (CNM), the CAPTAN Node Controller (CNC) and the CAPTAN Data Acquisition User Interface (DAUC) are integral parts of the CAPTAN system, and exist in support of the three different layers of the CAPTAN architecture. All the software resides in the network layer computers. The third layer, the application interface layer is a purely software layer and is implemented by the DAUC software. This software can be implemented by one or more computers from the CAPTAN network layer.

3.2 CAPTAN Hardware

The CAPTAN proprietary hardware is composed of four core boards and one network board. General use computers, network cards, routers, switches, and CAPTAN user boards can also be part of the CAPTAN system. The CAPTAN system in a few particular applications can exist without having a general use computer as the system manager, but most of the time the CAPTAN system requires at least one general use computer operating with the CAPTAN software installed.

The four boards currently considered to be primary boards include the Node Processing and Control Board or NPCB, the Data Conversion Board or DCB, the Power

Electronics Board or PEB and the Mass Memory Board or MMB. The CAPTAN system also has secondary boards, these are the boards that don't have direct access to the vertical bus, the Gigabit Ethernet Board or GEL is considered a secondary board. Users are free to design both primary and secondary boards as long as the architecture design rules are obeyed.

3.3 The Node Processing and Control Board (NPCB)

The NPCB is a primary board that processes information collected by the system or forwards it to another node or computer. The NPCB can handle information from the data buses as well as the system buses. There must always be one NPCB in a node that is responsible for driving the system bus. The NPCB also has two local buses. One is a high speed local bus typically connected to the Gigabit Ethernet Board (GEL). If not used for the GEL, this bus can be used as a general purpose local bus. The second secondary bus is 32 bits single-ended or 16 bits differential general use local bus. The NPCB also provides the gigabit open-air communication system. The system can also be connected to a POF (Plastic Optical Fiber), but if this is done the open-air communication capability is lost.

The primary component of the current version of the NPCB board is a Virtex-4 [Xil08] Field Programmable Gate Array (FPGA). Seven different versions of this FPGA are supported by the NPCB including the FX12 series devices that possess an embedded power PC hard core.

When stacking NPCB boards there is a tradeoff between maximum speed on the vertical bus and maximum number of boards. Table 3.1 shows the maximum bus clock

speed obtained experimentally with one node (configured with different numbers of boards) for which no errors were observed in intra-node communications on the vertical bus. The bus was tested in a loopback configuration in both single-ended and differential modes. When in differential mode, these values were experimentally obtained with double termination with the exception of a two boards stack.

Table 3.1. Maximum bus clock speed guaranteed

Number of Boards on the Node	Maximum Vertical Bus Clock Speed in MHz (single)	Maximum Vertical Bus Clock Speed in MHz (differential)
2	200	340
3	150	280
4	120	240
5	80	200
6	66	150
7	33	125
8	-	100
9	-	66
10	-	33

The single ended version of the test works with the top NPCB generating a 64 bit word in an FPGA counter and presenting the data to a 64 bit wide data bus. The data is then transmitted on this bus to the last board in the stack. The last board loops back the information through a second 64 bit wide bus and the information is presented back to the top NPCB. At this point the data is compared and if the received data does not match the transmitted data, an error is recorded. The electrical protocol used for this test was LVCMOS (Low Voltage Complementary Metal Oxide Semiconductor) operating at 1.8V.

The differential test works in the same way, except that the buses were 32 bits wide and there were 100 Ohms terminations in the node placed on the receivers at the

two extremes of the stack (one at the top and the other at the bottom). The electrical protocol used for this test was LVDS (Low Voltage Differential Signaling) working at 2.5V. In both single-ended and differential mode tests, if no errors were recorded for a transmitted data payload of 10GB, the test was considered to have passed.

Simulation was used in order to help the analysis of the data integrity on the bus and it matched within 20% of measurements. The basic parameters measured and compared with simulation included eye pattern opening, delay within a bus and propagation time. The board to board propagation time is a very important parameter and its measurement provided a mean of 80ps board to board propagation time without adding delays due to the buffers that vary depending of the FPGA grade being used.

It is also is important to mention that the normal power consumption of this board is on average is 1.5W when not providing power to any secondary board, and having on average 25% of the I/Os switching at 200 MHz rate. This number is a reference number as there are many factors that can bring this number up or down.

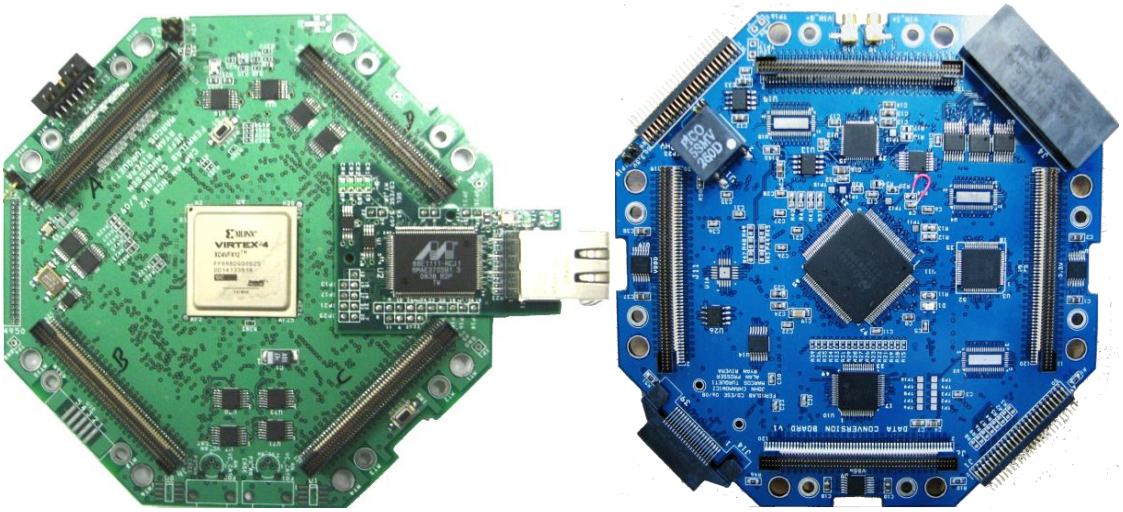

Figure 3.1. On the left in green the picture of the NPCB board attached to the GEL board. On the right in blue the picture shows the DCB board, in black its four lateral connectors are visible on the angled edges of the board. On the top of both boards it is possible to observe in black and silver the electronic components.

The main contributing factors for this are: configware type, GEL utilization, devices supported by the secondary bus, speed of operation and number of I/Os in use.

Figure 3.1 depict a picture of the NPCB and DCB boards, on this figure it is possible to observe the four top connectors that compose the vertical bus and the two secondary buses on the laterals of the card. In the center of the NPCB board it is also possible to observe the FPGA that is the focal part of the board.

The Data Conversion Board or DCB is the primary system board used for data acquisition. This board includes Analog to Digital Converters (ADCs) and Digital to Analog Converters (DACs). There are three different ADCs on the board, providing various conversion speeds. Table 3.2 provides some of the details for the different converters. The 65 Msps ADC includes an operational amplifier in each channel, where gain and offset can be adjusted digitally.

Table 3.2. Summary of the analog to digital converter capabilities of the DCB board

ADC TYPE	Number of Channels	Resolution (Bits)	Sampling (Msps)
ULTRAFAST	2	8	1600
MEDIUM	8	12	65
SLOW	12	32	0.3

Also available on this board are two DACs, one fast and one slow. A summary of the capabilities of the DACs is presented on Table 3.3.

Table 3.3. Summary of the digital to analog converter capabilities of the DCB board

DAC TYPE	Number of Channels	Resolution (Bits)	Sampling (Msps)
FAST	2	12	500
SLOW	32	16	0.3

Access to the ADCs and DACs is provided on three lateral connectors on the board, with exception of the ultra fast ADC that has special dedicated coaxial connectors. The board also provides a bridge to the vertical bus through a fourth lateral connector. This bridge is capable of level translation, and it is meant to allow external peripherals access to the vertical bus.

In addition to the above features, the DCB board also provides five adjustable regulated voltages through the lateral connectors. Table 3.4 provides the capabilities of the voltage regulators available on this board to the lateral connectors.

Table 3.4. Summary of the board regulation capabilities

Regulator Type	Number of Channels	Range (Volts)	Max Current (mA)
Low Voltage Positive	3	1.0 to 5.0	2000
Low Voltage Negative	1	-2.5 to -5.0	200

High Voltage Dual	1	50 to 250	2

To configure and control devices the DCB interfaces with the System Bus by means of four CPLDs (Complex Programmable Logic Device), one for each of the four System Bus quadrant connectors, making available to the parameters such as sampling rate, voltage references, channel gain, offset and voltage levels. Figure 3.1 shows the top portion of the DCB board, the 1.6 Gsps ADC is at the center of the board. It's also possible to observe horizontal connectors that give access to external devices to the ADCs, DACs and indirect access to the vertical bus.

The Power Electronics Board shown on Figure 3.2 is a primary board that provides regulated power supply for the four power channels available to the vertical bus and driving capabilities for controlling external devices such as motors, actuators and relays.

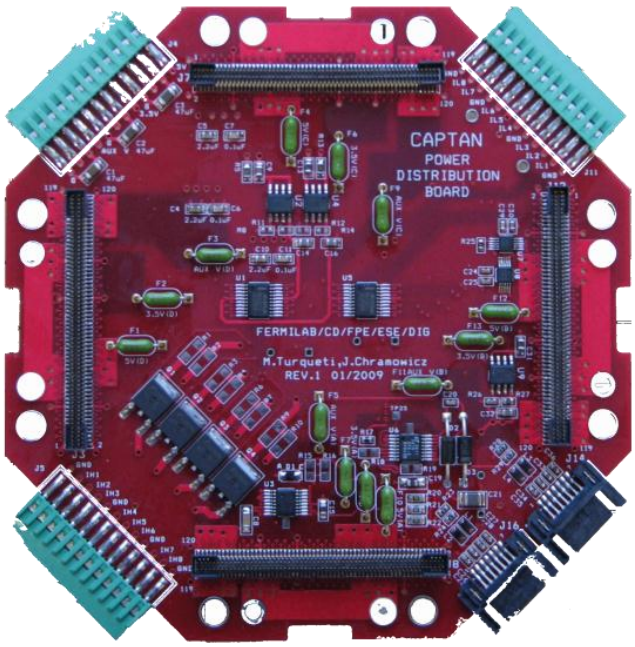

Figure 3.2. In red the Power Electronics Board, on three angled edges of the board its possible to observe the green power connectors rated to different power levels and on the forth edge in black connectors for data over power communications.

The power board is not required for a stack to work, since the vertical bus can be powered externally through any NPCB board without the power board. The power board adds regulated monitored power and protection to the bus. The power board also provides eight IGBT high power drives and eight MOSFET medium power drivers in order to control devices that require high current switching capabilities. This board can provide a maximum combined power of 100W to the vertical bus and additional 20 W for the onboard switches.

The Mass Memory Board is a primary board that provides up to 1 GB of DDR3 memory. The objective of this board is to provide a high speed local mass memory to the system. The design of the board supports standard DDR3 memory cards, and every card is connected to a carrier CAPTAN board that is connected directly to the 64 bits bus, they can only be connected to the bus when this is operating in single ended mode LVTTL (Low Voltage Transistor-Transistor Logic). This board is still under development and more results on the performance will be provided in the future.

The GEL is a secondary board providing gigabit communication between nodes or between a node and a computer. This board is the main network interface of the CAPTAN node and can communicate directly with any computer with 1000BASE-X network capabilities. The board is designed to work with Ethernet protocol 10/100/1000 and to use UDP/IP as the communication protocol.

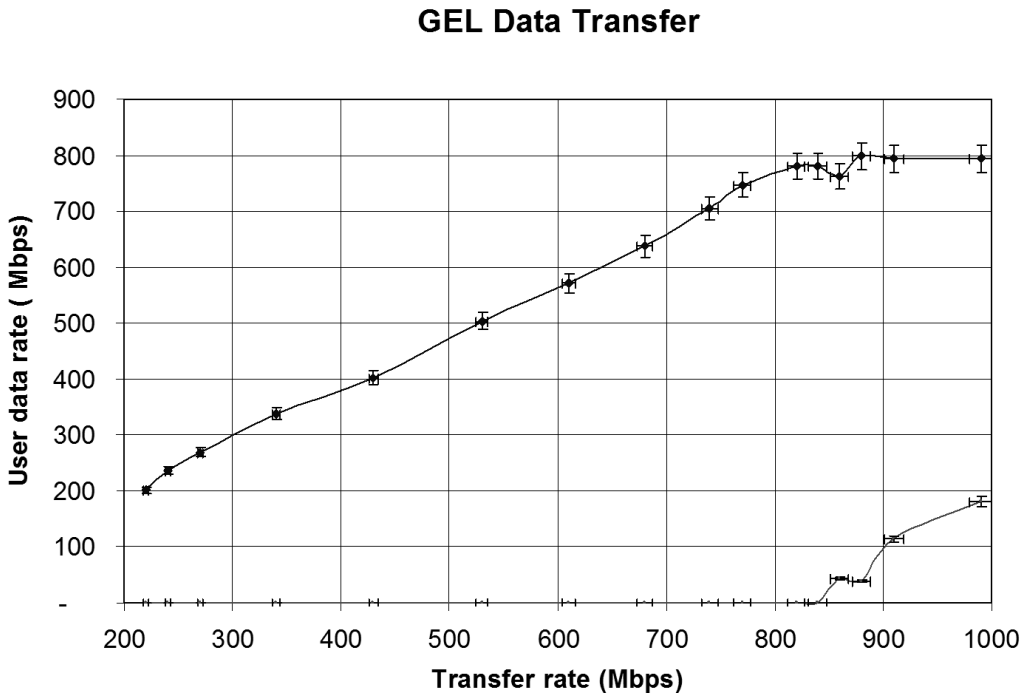

Figure 3.3. This graph shows how fast the user can transfer information before it starts to loose data. Where the horizontal axis refers to the rate that data is being throttle and the horizontal the rate that user data is being transferred. The top curve refers to data successfully transferred, and the bottom curve data lost.

Although the board is capable of connecting using the IEEE 802.3ab (1000BASE-X) [Iee99] protocol it cannot send pure user data at this speed due to the addition of several layers of protocol, maximum packet size limitations and the particular hardware used. The board can however send user data up to 800 Mbps as illustrated in Figure 3.3. The performance demonstrated in Figure 3.3 was obtained using the UDP protocol with the packet size set to 1466 bytes and with the PHY running 1000BASE-X.

The power consumption of the GEL board is 1.25W when running in 1000BASE-X. Any NPCB in a stack can support a GEL board. The GEL board can currently only interface with the node through the NPCB board and the NPCB FPGA configware must

include the gigabit Ethernet controller configware as the GEL board itself contains only the gigabit PHY.

When more customized hardware is necessary user boards can be a very useful tool for specific applications; guidelines are provided for users to design their own boards compatible with the CAPTAN system. Either primary or secondary boards may be designed but secondary boards tend to have less overhead than primary boards. In order to design a primary board there are three main aspects that must be taken into consideration. The first is the vertical bus rules that must be followed. For example, a primary board must have buffers to isolate itself from the vertical bus. Second, power limits must be obeyed (the board cannot consume more than 12 W of power). Finally the designer must follow the mechanical rules for the primary board type, which will guarantee that cooling channel, optical link and bus interfaces will match the existing system.

For the design of a secondary board, the rules are less restrictive as the only mechanical constraint that the user will face is on the lateral bus connection. Also, if the user is utilizing power from the stack drawn through the lateral bus, it cannot exceed 3 W.

3.4 CAPTAN Node Architecture

The CAPTAN node architecture is defined by three main characteristics: the vertical bus, the horizontal bus and the core boards. These three characteristics give this architecture exceptional versatility.

The vertical bus is the main communication pathway allowing a board in the node to communicate directly with another board in the same node, broadcasting information to the other boards on the stack. The vertical bus is designed to deal with large amounts of data at high speed that is transported over four identical connectors which are present on the top and bottom of every CAPTAN core board. A node of two or more boards is assembled by connecting two or more boards together in a stacked arrangement. The optical bus that allows high speed optical communication between boards is also part of the vertical bus. In order to control access to the vertical bus the System Bus was created. The System Bus physically belongs to the vertical bus, but the kind of data that runs through this particular part of the vertical bus is for control only and no user data is moved through this bus.

Electrically the vertical bus is divided in twelve independent data buses (four with 64 pins, another four with 16 pins and finally, four with 10 pins). In addition to the data bus there is a single 48 pins system control bus and one 16 pins SPI system bus. Power is also distributed to the system over these same connectors providing 3.3V, 5.0V, 12.0V and -12V for the entire node. The power provided by the vertical bus is not necessarily well regulated because the architecture pushes the responsibility for voltage regulation towards the individual boards.

Figure 3.4 illustrates how the bus is organized at the connector level, while Figure 3.5 illustrates the localization of the common components of all CAPTAN core boards.

Figure 3.5 also depicts the baseline design of any CAPTAN core board, with its four vertical bus connectors, and provides information about the organization of the node architecture.

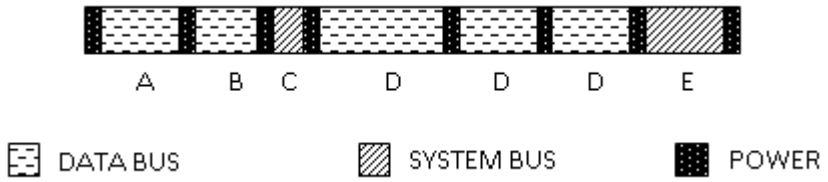

Figure 3.4. This figure depicts one of the four identical buses that run vertically through the board where C and E are pins reserved for the system bus. A, B and D are data buses of widths of 16, 10 and 64 pins respectively.

The design of the four buses reflects a high degree of symmetry making the rotational orientation in a stack largely unimportant. Boards can be added to a stack in one of four different orientations. This provides the means to make the best use of board resources in a stack without limitations imposed by the presence of other boards in the stack.

The vertical bus can work in two modes, single-ended mode and differential mode. Since the buses are independent, mixed mode configurations are possible providing flexibility for the operation of the individual boards in the stack. When working in differential mode, the standard supported is LVDS and the number of bits that the specific bus can carry falls by the half of the number of pins available to the bus.

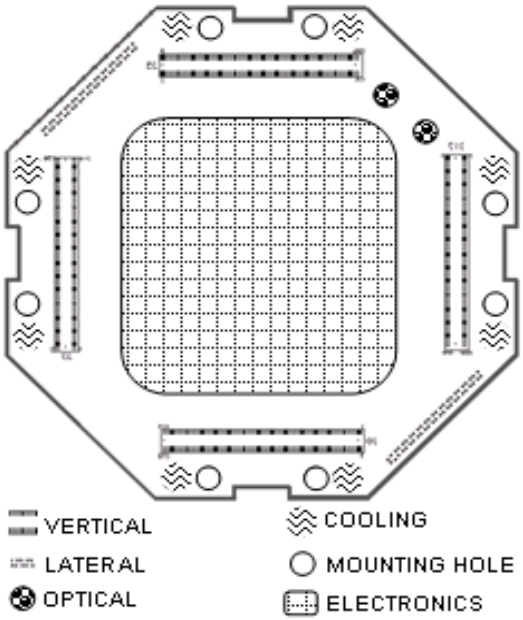

Figure 3.5. This figure depicts the baseline design of any CAPTAN core board with its four vertical bus connectors, vertical bus, lateral bus (horizontal), electronics pack, optical drivers, cooling channels and mounting holes.

Due to the desire to support high speed data communications with robust signal integrity, the CAPTAN system was designed to work primarily with differential signals so its core boards and the bus are designed to support LVDS. The possible vertical bus LVDS configurations for the CAPTAN system are implemented using 32, 8 and 5 bits buses.

The single-ended mode offers a wide range of operation, where each bus can be configured separately to operate as LVCMOS or LVTTL. The use of this mode is desirable when the communication speed between boards is not an issue. Single-ended mode supports 64, 16 and 10 bits data buses.

The optical bus is optional for the vertical bus intra-node communications. It is a bidirectional system of high speed lasers for open-air communications capable of providing direct intra-node connections. It is capable of transferring information at up to

1 Gbps in a serial fashion regardless of where in the stack the board is located or how many boards are on the stack.

There are special requirements that every core board must follow to allow open-air stack communications. In order to repeat the laser signal the optical transceiver must be placed in a specific location on the board, or a physical window for the laser to go through must be provided in the local illustrated by Figure 3.5.

The System Bus consists of the system control bus and the system SPI bus. This bus is also distributed in four quadrants across the four inter-board connectors. In addition to the control of the CAPTAN stack, the System Bus also supports the programming of devices on the boards of a stack. The controller of the System Bus is implemented by a NPCB board. Any CAPTAN node is required to contain such a NPCB board. Any NPCB in the node can serve as the node controller but only one node controller can be active at any given time.

The primary task of the system control bus is to carry messages within the node regarding the status of the data buses, ensuring that access to each bus is granted in a safe fashion. Priorities and policies governing bus access are specified on the System Bus controller, a configware block resident in the NPCB.

The system SPI bus is used to distribute the configware needed to configure programmable logic devices residing in any of the boards forming the node. The configware for these devices is delivered from the FPGA on the NPCB. Other signals on the system control bus include a 33MHz reference clock and the node hardware reset signal. The System Bus has many pins reserved to accommodate growth of the architecture.

The 48 pins of the system control bus are named as GENERAL_00P_00S to GENERAL_23N_47S, while the SPI System Bus is named as JTAG_TDI#, JTAG_TDO#, JTAG_TCK# and JTAG_TMS# where # ranges from 1 to 4.Currently the possible implementations of the System Bus are listed below:

Table 3.5. Currently System Bus Possible Implementations

Type	System Control	SPI System	Termination Type
Custom	N/A	N/A	Static
Type 1	Yes	No	Static
Type 2	Yes	Yes	Static
Type 3	Yes	Yes	Semi-Dynamic
Type 4	Yes	Yes	Dynamic

When the System Bus is implemented as CUSTOM, the user will configure it for its own specific architecture implementation, when that is the case the only rules that apply for custom boards are the maximum bus power and the use of the correct vertical bus connector pin out. It is also important to observe that the System Bus must be implemented with single ended signals.

When configured as TYPE 1, the SPI System Bus is inexistent, and the System Control is configured as a bare minimum, in this case the bare minimum is a direct control line to the stack, a general reset line and a system clock. In this configuration access to the bus is granted through a direct point to point connection from the System Bus Controller to the board requiring access in a two wire communication flag fashion; with pins type P coming from the bus controller and granting access to that board for the bus and type N going to the controller and requesting access.

This system limits the number of devices connected to twelve data buses available on the vertical bus to up to 23 devices, with every device having its own pair of bus

control lines. This implementation of the system control is simple and fast. It also greatly reduces the complexity of the System Bus Controller implemented on the NPCB. Table 3.6 illustrates the way that the System Bus is configured for configuration TYPE 1.

Table 3.6. Type 1 System Bus Configuration

Pin Name	Function
GENERAL_00P_00S	System Clock REF
GENERAL_00N_01S	Reset
GENERAL_01P_02S … GENERAL_23N_47S	Direct Bus Access
JTAG/SPI	N/A

The System Bus TYPE 2 implements the SPI System Bus and the System Control. The solo task of the SPI System Bus is to download the configware to the others NPCB and DCB boards on the stack. This is done through the NPCB that has the System Controller implemented. The configware is downloaded in a cascaded fashion, with the SPI controller downloading initially the configware to board one, then to board two and incrementing in this fashion up to the last board on the stack. The SPI system first broadcast in the SPI bus which board will be programmed and the device on the board then sends the bit stream to the bus.

For the TYPE 2 implementation of the System Bus, the System Bus Control is implemented in a similar fashion, but with the difference that now there are 17 pairs for the direct bus access mode and 11 lines for the indirect bus access mode.

Table 3.7. Type 2 System Bus Configuration

Pin Name	Function
GENERAL_00P_00S	System Clock
GENERAL_00N_01S	Reset
GENERAL_01P_02S … GENERAL_18P_36S	Direct Bus Access
GENERAL_18N_37S … GENERAL_22P_44S	Indirect Bus Access Upstream
GENERAL_22N_45S … GENERAL_23N_47S	Indirect Bus Access Downstream
JTAG/SPI	JTAG Programming Only

The indirect access grants access to eight of the data buses, the four 32/64 bits buses and the four 8/16 bits.

A modification to the bus status of the available data bus is granted every 12 clock cycles. This means that every 12 clock cycles of the system clock, the eight buses above mentioned can be reassigned. Table 3.8 provides the pin assignment of the System Bus for the indirect bus access mode pins.

The way that the indirect bus access works is that the board doing the request for access to the bus must place its request on the bus on the right clock cycle, the controller keeps a counter counting up to 12, this counter is counting the clock cycles on the reference clock and every time it reaches 12 it reset itself. All boards must be in sync with this counter and put the request for the bus in its designed clock cycle, for convenience there can be only 12 boards on the system, this translates in one available slot for a bus request every 12 clocks. The bus request is presented through the pins

BUSREQB0, BUSREQB1 and BUSREQB2 working together as a 3 bits bus, where the information about the request for the bus will be presented. The information presented is what bus is being requested.

The limitations of the indirect bus are that transfers for the data access for the data bus must be multiples of 12 system clock cycles. Also only one bus per board can be switched every 12 clock cycles. It's important to mention that this mode can be used in association with the direct mode.

Table 3.8. Type 2 indirect bus pin assignment

Pin Name	Function
GENERAL_18N_37S	BUSAA(0)
GENERAL_19P_38S	BUSBB(1)
GENERAL_19N_39S	BUSCC(2)
GENERAL_20P_40S	BUSDD(3)
GENERAL_20N_41S	BUSA(4)
GENERAL_21P_42S	BUSB(5)
GENERAL_21N_43S	BUSC(6)
GENERAL_22P_44S	BUSD(7)
GENERAL_22N_45S	BUSREQB0
GENERAL_23P_46S	BUSREQB1
GENERAL_23N_47S	BUSREQB2

The configurations of system control bus TYPE 3 and TYPE 4 are novel and currently are under study, they use full reconfiguration, and partially reconfiguration, respectively. This allows the terminations on the vertical bus to be dynamic rather than static when in differential mode. This new techniques can enhance the maximum board to board communication speed on the vertical bus since the termination will always be in the ideal spot. TYPE 3 would use a similar architecture as TYPE 2 with the addition that when a board to board operation is required the configware can be reloaded automatically

with the proper termination. This operation takes a maximum of 25 milliseconds and all the registers and memory on the device would be wiped clean, this significantly limits the usefulness of this type. TYPE 4, in contrast, makes use of partial dynamic reconfiguration, supported by the VIRTEX 4 FPGA through the use of the Frame Address Register (FAR) [Xil09]. This allows the terminations to be changed without prejudice to the device memory or registers, it is also much faster, and it can be implemented for one bus in less than 100 microseconds. Studies for practical implementation of this mode are currently under way.

The horizontal bus is a local bus, connected only to the electronics of the board itself and does not connect directly with the vertical bus without a bridge. The main reason for the existence of the horizontal bus is to support the connection of secondary boards that can collect data or provide signal conditioning for data. The horizontal bus can also be used for node-to-node communication, either directly, or through a secondary card bridging two nodes.

The configuration of the horizontal bus varies greatly, but the baseline for the digital boards is 32 pins for data communications (differential or single ended), 12 pins for control signals, clocks and two power supplies. The bus when implemented on a DCB allows also support for analog signal input or output.

3.5 System Constraints

One of the main features of the CAPTAN architecture is that it empowers the user to develop a wide variety of systems. In the core of this architecture are the system boards that can be divided into core (or primary) and secondary boards. At the same time

the architecture is made in such a way as to encourage users to design their own boards in cases where the system boards cannot fulfill the requirements of the specific application. This is done by providing the user with the foot print of a blank board and the rules that must be followed in order to successfully integrate this new board into the system.

When stacking boards to build a node, there will be constraints, primarily due to limitations of system performance, and power delivery, limiting the maximum number of boards that a node can contain.

For example, with every board added to the stack the maximum communication speed that the vertical bus can support decreases due to the effects of cross-talk, signal reflections, capacitive loading, and different delays between lines of the same bus. The maximum number of boards due to these constraints is not easy to predict because it depends on the types of boards being stacked. Experimentally, a limit on the number of boards in a stack is seen to be somewhere between 8 and 12 boards, depending on the types of primary boards which make up the stack.

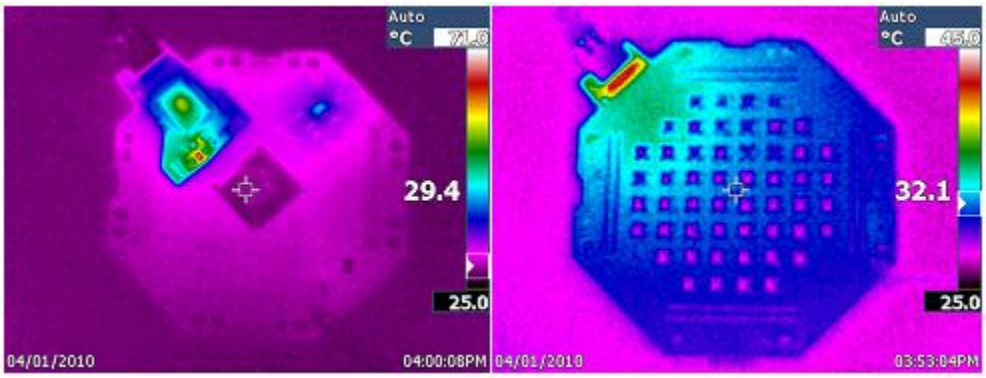

Figure 3.6. This figure depicts the thermo profile of a stack of two boards, on the left the NPCB and on the right the Acoustic MEMS array.

Power available to operate the stack is limited by the maximum current that the vertical bus can handle and it yields a limit of roughly 100W. This limit is for the whole node, so the number of maximum boards due to power consumption depends on how much power each board consumes, but in practice due to voltage drop in the vertical bus the node should have no more than 10 boards.

Individual boards are limited to a maximum power consumption of 12W due to the limited cooling capability available to the stack. Figure 3.6 illustrates the case where a sonic array board is stacked on top of a NPCB board. It's possible to observe that the GEL board is the one that dissipates most of the power due to the Gigabit transceiver. The sonic array is practically not influenced by the heat coming from the NPCB.

The cooling of the stacks is performed by convection with air and there is an option to add copper rod bars that run from the top to the bottom of the stack on the cooling channels. There are eight cooling channels; they appear as round openings on the vertices of the CAPTAN boards. The cooling channels are connected to the solid ground copper plane that every CAPTAN board has, this channels can be observed on the edges of the board on Figure 3.6 and Figure 3.7. The rods connecting the cooling channels can then be cooled by liquid coolant.

3.6 Applications

There are a number of possible applications for the CAPTAN system, including data acquisition systems, data processing systems, and mixed applications. The flexible nature of the CAPTAN architecture makes possible topologies which can be individually

suited to the application. Some examples of topologies to support different applications will be illustrated next.

The powerful FPGA of the NPCB makes it possible to configure a CAPTAN system with Gigabit Ethernet service using a single primary board and a GEL board. Depending on the FPGA used on the NPCB board, up to 400 I/O pins that can be configured as 400 LVCMOS or LVTTL signals or 200 LVDS signals are available as user I/O. This option provides ample support for many applications that do not require analog capabilities. One natural application is the construction of test stands for the evaluation of prototype systems. Since the NPCB board can be assembled with an FPGA with an embedded PowerPC core, applications which benefit from embedded software solutions can be supported. As the basic configware that the system requires to be implemented on the NPCB FPGA occupies less than 10% of the FPGA, there are considerable resources available for the implementation of application specific configware such as data compression or digital signal processing blocks.

The stacking of multiple primary CAPTAN boards creates a more powerful node and greatly expands the capabilities of the system. Such a node may be capable of dealing with both analog and digital information if it includes a DCB as one or more of the primary boards residing on the stack. Depending on the number of NPCBs in the stack, up to ten GEL boards can provide networking access to multiple boards in the node. Nodes of this type find applications as test stands systems such as pixel detector readout and devices with analog readout architectures [Men07]. These nodes can be used as laboratory bench test systems for characterizing individual modules for a pixel detector or any other kind of analog transducer.

Table 3.9 shows three of many possible configurations that a multiple board node can assume and some key capabilities that these nodes would offer. The data rates between the boards of a node and between other networked elements (e.g. a personal computer or another CAPTAN node) are listed for different stack compositions (i.e., different combinations of NPCBs and DCBs in a node).

Three different configurations of the node are explored, the first one contains 8 NPCBs boards, the second one 7 DCBs boards and one NPCBs and the third one 4 NPCBs boards and 4 DCBs boards.

Table 3.9. Node performance in different configurations

Total Number of Boards	Number of Ultrafast ADC channels	PowePC Cores	Intra-node data rate transfer (Gbps)	Inter-node data rate transfer (Gbps)
8	0	8	19	6.4
8	14	1	19	0.8
8	8	4	19	3.2

An extra resource that a stack can utilize is its GEL boards for communication within the stack, this can greatly expand data rates within the node.

Figure 3.7 shows an actual stack composed of three NPCB boards, on the top of the stack is a JTAG connector used to download the initial configware to the FPGA. While NPCB boards inserted on the stack must initially be configured with the basic configware, the configware can be updated later over the Ethernet connection. In addition, any DBCs in the stack can also have configware updates applied in-system by using the FPGA in the node as a programming device. The bit stream is transferred over the vertical bus and delivered to a quadrant CPLD.

Figure 3.7. Node configured with ten boards, the color green indicates that it is a NPCB board, on the top in black and silver the active electronic components.

Distributed systems with multiple CAPTAN nodes can be configured using the Ethernet capabilities of the nodes. There is no limit to the number of nodes that can be interconnected making it possible for such a configuration to deal with very large amounts of data at very high speeds.

An example of such a network application is the acquisition, processing and control of composed transducer arrays. Due to the scalable nature of the architecture, this design can be expanded to include thousands of detectors in a straightforward manner to support the data acquisition needs of large systems.

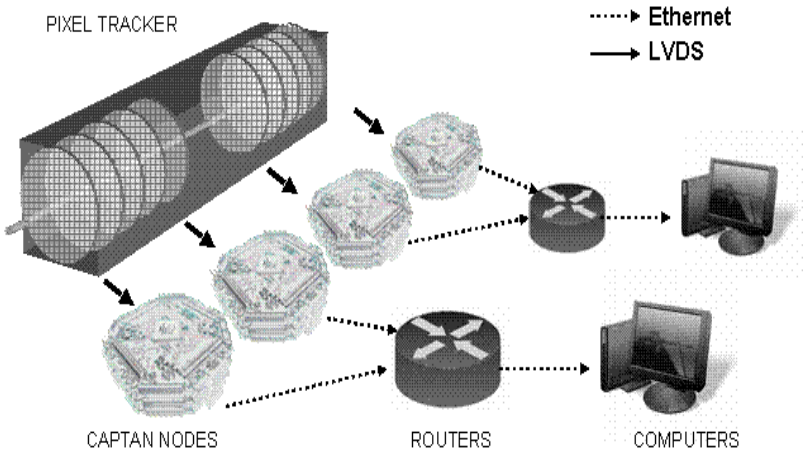

Figure 3.8. Applications such as pixel tracker readout for high energy physics are possible [Upl06].

Figure 3.8 shows such an application, utilizing commercially available components such as routers, switches and Ethernet cards compatible with the CAPTAN system nodes. Such a system makes the best use of cost-effective off the shelf components with application specific CAPTAN components. Software architecture suitable for implementing such networked applications is described in [Riv08].

Another possible application for a networked CAPTAN application is as a computing farm. Such application could support special purpose computational tasks, possibly requiring digital signal processing requirements with more general purpose computing tasks, in a parallel computing implementation, Figure 3.7 illustrates what a parallel processing node looks like.

3.7 Conclusion

The CAPTAN system is powerful and flexible data acquisition system architecture with its roots in sensors R&D. Due to the flexible and expandable characteristics of the system, it can be utilized in a wide range of applications. The CAPTAN architecture is novel in that this systems scales by taking advantage of the node approach for easy growth. It also supports multi-process and distributed computing, and provides added capabilities through convenient interfaces to analog instrumentation.

CHAPTER 4

THE MICROPHONE ARRAY

4.1 The AMA Array

The detection and analysis of sound sources utilizing microphone arrays is a growing field of research due to the evolving needs of fields such as noise suppression, voice recognition, sound tracking and acoustic source localization. Microphone arrays are advantageous when compared with single microphones in the sense that they can use the physical spatial information of sound propagation to provide a more accurate picture of the incoming sound waves and therefore yield valuable information [Ben08]. This chapter describes the broadband array MEMS sound imaging system called AMA.

The AMA board is a user CAPTAN board and as such it obeys all the requirements demanded by the CAPTAN architecture. In addition design issues such as array sensitivity, accuracy and geometry of the array will be discussed, as well as the system's real-time capabilities regarding signal processing and data acquisition. Source separation and localization are the main target applications of this system.

Two boards were developed for this work, the AMA I and AMA II and they can be observed on Figure 4.1. The first one was used as a research platform for optimizing parameters such as microphone separation and array geometry. The second board used the findings from the first and was then employed for source separation and localization.

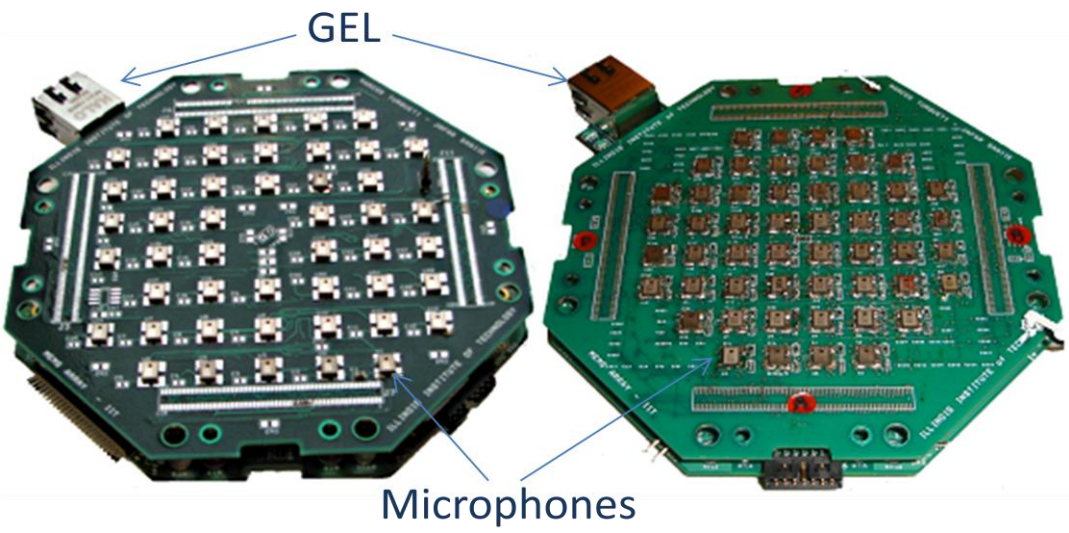

Figure 4.1. On the left the AMA I board and on the right the AMA II, in red are the bus labels, in green are the printed circuit boards, and in silver the microphones and connectors.

4.2 The Microphone Array Board Design

Careful considerations was taken for the design of the microphone array board, basic parameters for the design were extracted from a test board (AMA I), built with the solo purpose of deciding the best topological distribution of sensors, number of microphones, inter-microphone spacing and microphone type.

In order to avoid spatial aliasing, and have a good acoustic aperture, the initial inter-microphone distance was 12.0 mm centre to centre, this spacing made it possible to obtain relevant phase information of incoming acoustic sound waves thus increasing the array sensitivity and allowing spatial sampling of frequencies up to 14 kHz without aliasing, it can be approximated by dividing the speed of sound by the inter-microphone distance and then further dividing the result by two in order to satisfy the Nyquist–

Shannon theorem [Wei04]. On the second version of the board this distance was further decreased to 10.0 mm in order to allow a spatial resolution of up to 17 kHz.

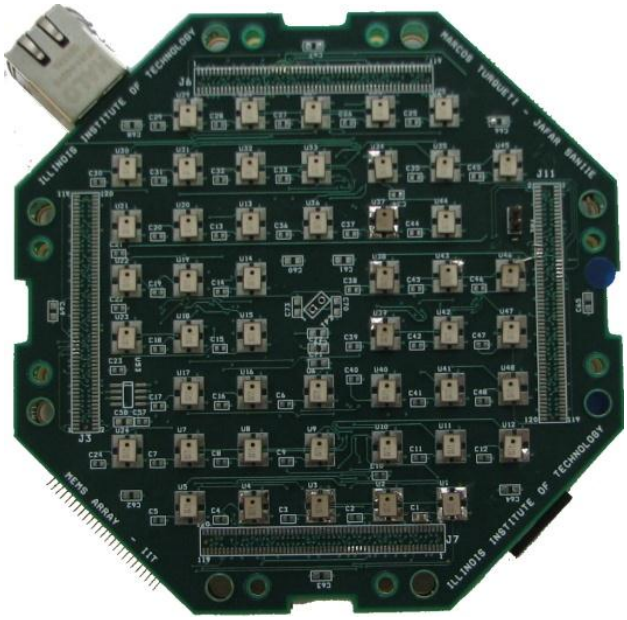

Figure 4.2. AMA I Microphone Array Board. 48 Microphones spaced 1.2 cm centre to centre. In green is the printed circuit board in silver the microphones and connectors.

The number of microphones was a compromise between the size of the board, the electronics needed to deal with the massive amount of data, and the need to achieve a signal-to-noise performance of at least 20dB in order to enhance the system capability to distinguish signals from weak sound sources. The AMA I [Tur10A] board (Figure 4.2) is composed of 48 MEMS microphones distributed in an octagonal fashion having 7 columns and 8 rows with the two central microphones substituted by a transducer and two missing microphones on columns 1 and 7 as shown on Figure 4.2. This was an exploratory geometry chosen in order to allow beam steering on the horizontal and vertical planes [Ben08]. The central loudspeaker has dual function: it can be used for

calibration purposes, or for sonar like applications. When used as calibration element the microphone emits a set of pure frequencies that are then captured by the microphones and used for its calibration taking into account the geometry of the array. When the central microphone is used for active sonar applications a series of pre-programmed pulses are emitted by the central microphone, when these pulses hit obstacles they bounce back to the array thus giving information about the distance and position of the obstacles.

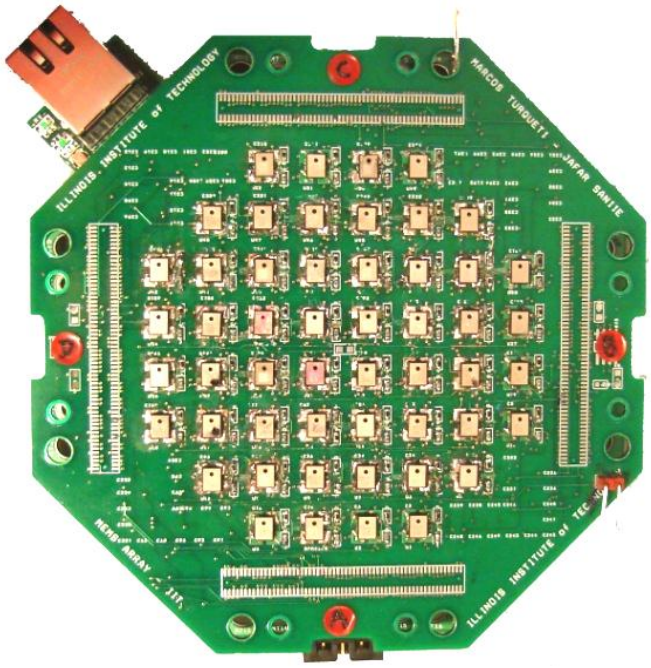

Figure 4.3. Top view of the AMA II acoustic array. The MEMS elements are clearly visible on the top of the printed circuit board (PCB). In red are the bus labels, in green is the printed circuit board in silver the microphones and connectors.

The AMA II (Figure 4.2) board increased the number of microphones to 52, the sensitivity of the array increases monotonically with the number of sensors, and the MEMS microphone chosen for this array (SPM0208) has a sensitivity of 1V/Pa at 1 kHz [Kn06]. These microphones are Omni-directional and when combined on the array they provide a very good acoustic aperture. The increase in the number of microphones

opened the opportunity to give different gains to multiple channels therefore minimizing saturation problems.

In support of the microphones, both boards also provides analog to digital convertors that interfaces with an NPCB board through the four board to board vertical bus connectors. The readout system is based on the CAPTAN architecture [Tur08] and its implementation as a whole will be presented next.

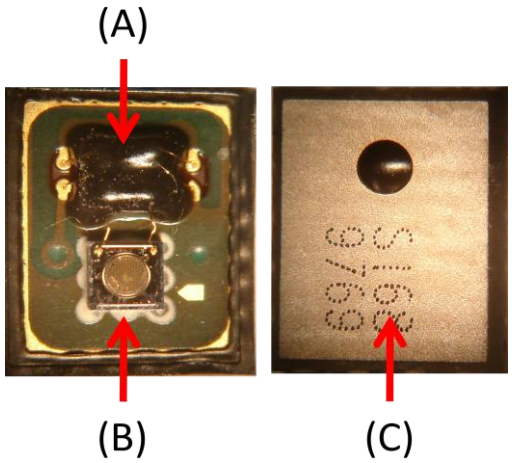

Figure 4.4. MEMS microphone, (A) in black the amplifier, (B) in gold the microphone and (C) in silver the aluminum cover.

The MEMS microphones are a fundamental piece of the array due to the small size, high sensitivity, and low reverberation. Its frequency response is essentially flat from 1 to 8 kHz and it has a low limit of 100 Hz and a high limit frequency of 25 kHz.

In order to reduce reverberation on the system the microphones are glued with silver epoxy to the copper pads in the board. Figure 4.4 illustrates the MEMS microphone used on this work.

4.3 System Implementation

The final system was implemented using the AMA II [Tur10B] array and is comprised of three different pieces of hardware, the first is the MEMS array board itself that contains the microphones, amplifiers and ADCs. The second is the Node Processing and Control Board (NPCB) that contains the FPGA, the system configware, and the third part is the Gigabit Ethernet Board (GEL); these last two boards are part of the CAPTAN system. Figure 4.5 illustrates the three pieces of hardware that compose the system.

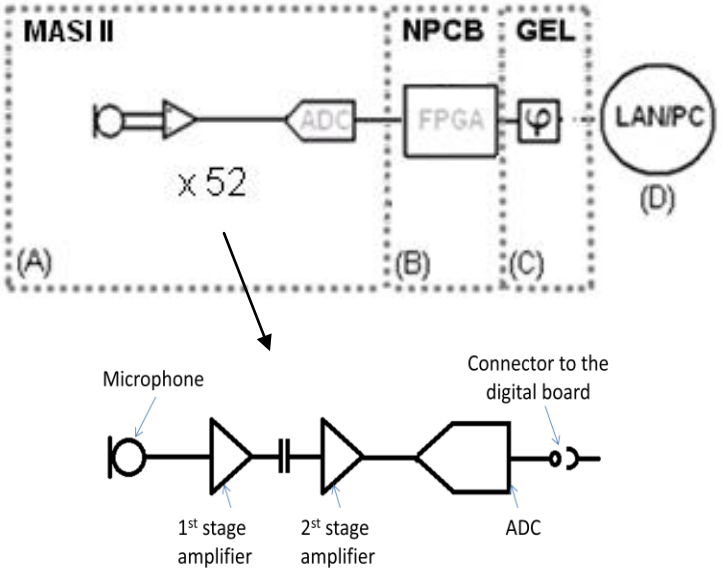

Figure 4.5. On the top, the system block diagram, on the left (A) the microphone array is presented with the amplifiers and ADCs, the signal captured by the microphones is then conditioned by the amplifiers and digitized by the ADCs, the data is then forward to (B) where signal process take place and is then presented to the GEL (C) that make the information available for the network (D). On the bottom, the AMA II electronics in more details.

The board containing the MEMS array is responsible for the data acquisition, each of the 52 MEMS microphones has one front end signal conditioning circuit. This

circuit is presented on Figure 4.6 and has the objective of amplifying and filtering the microphone signal.

The first stage of the amplifier gives a 20 dB gain and is embedded on the microphone case as indicated by the dashed area on Figure 4.5. The second stage of the amplifier gives a further 20 dB gain where the gain can be adjusted by the feedback resistor. The amplifiers also provide a second order high pass filter with corner frequency set to 400 Hz.

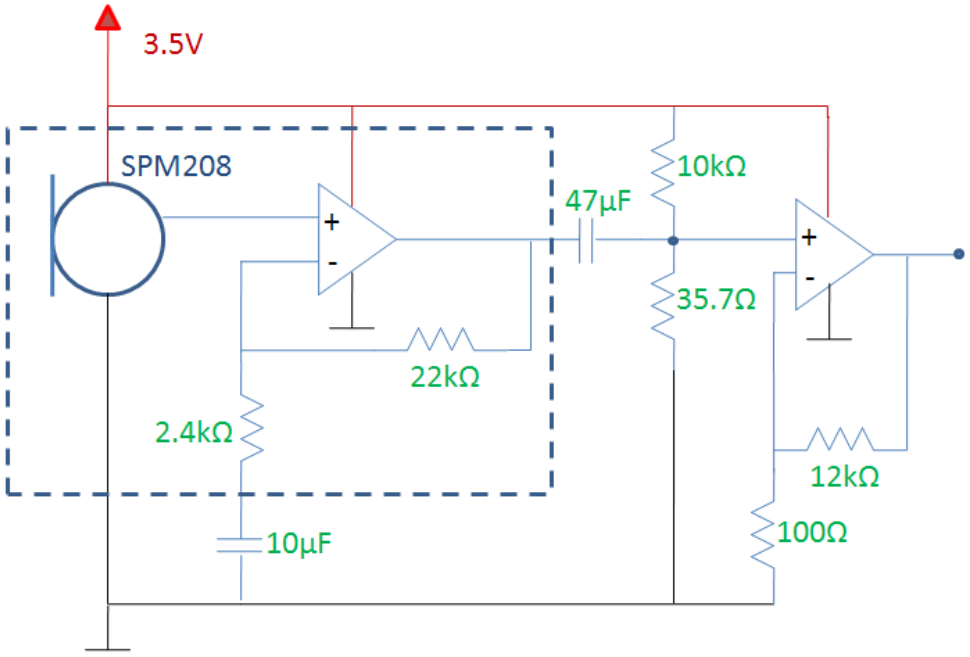

Figure 4.6. Front end electronics of the AMA II array, in solid blue the circuit elements, in red the power network, in green the components values and in dashed blue the embedded MEMS circuit.

This board support two different commercially available MEMs microphones from Akustica, the SPM208 with dynamic range from 100 Hz to 12 kHz and the SPM204 with a dynamic range varying from 10 kHz to 65 kHz, these microphones can be intermixed on the array or used as the solo type. The two different sensor types allow

sound and ultrasound applications such as sound tracking and ultrasound ranging. The MEMS used in this work was the SPM208 and its frequency response is shown on Figure 4.7.

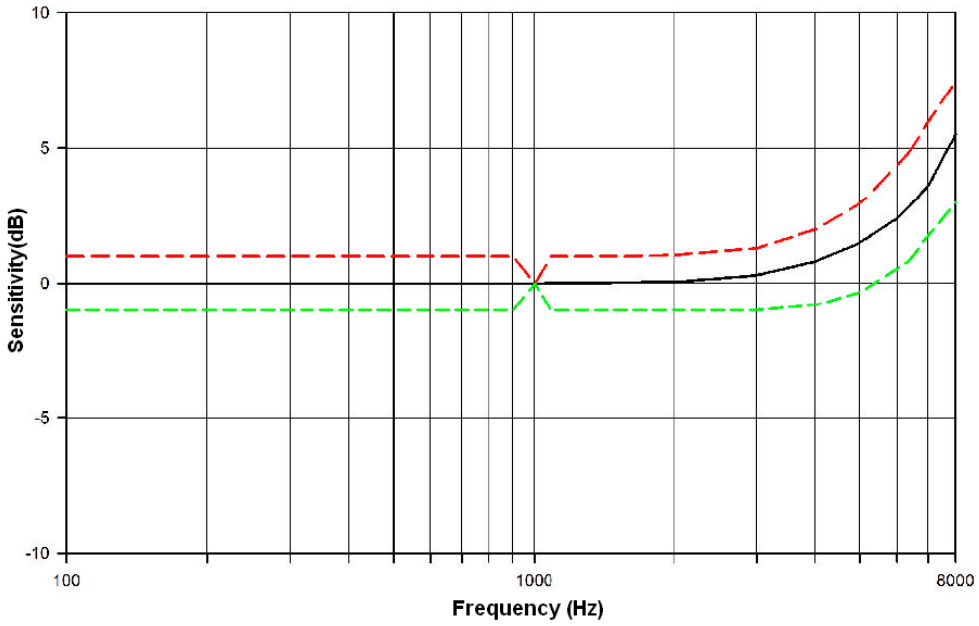

Figure 4.7. SPM208 frequency response. The lines on the top in red and on the bottom in green are the error margin of sensitivity.

After the analog signal is conditioned it is handed to the channel Analog to Digital Converter. Every single channel has its own ADC, an ADC121S101 [Nat10] from national. This is a serial 12 bit A to D converter with maximum sample rate of 1 Msps. The system presented on this work was set to work at 36 Ksps, but it can easily be adjusted in the configware to any sample rate desired up to 1 Msps.

After the ADC digitizes the signal it generates a serial bit stream at 430 Ksps, this bit stream is continuously generated by all 52 ADCs creating an overall data rate of 21.5 Msps handed to the NPCB board through the four vertical bus connectors.

The NPCB board contains a VIRTEX-4 XC4VFX12, all data from the array is stored and processed on it. The FPGA is connected to a 32MB EPROM that contains a specially designed configware for dealing with the data coming from the array, this configware is automatically loaded to the FPGA every time the system is powered up. It is important to mention that the configware plays a central part on the system architecture and it is divided in three distinct modules, Acquisition Control Module, Signal Processing Module and the Ethernet Communication Module.

The Acquisition Control Module is the block that contains the SPI (Serial Peripheral Interface) interface to communicate with the ADCs, this block is responsible for controlling the ADCs by programming their registers, it is also responsible for receiving and formatting the data sent by the ADCs. Data coming from the Acquisition Control Module modules is then sent to the signal processing block, the processing block is application dependent it usually contains DSP modules but can be as simple as a buffer, from this block the data is forward to the Ethernet block.

The Ethernet communication block is part of the CAPTAN architecture, and it is a specially designed configware that formats data to the UDP protocol [Tur08] and sends it to the GEL board. This block does all network communication using UDP, and is a full duplex system being able to transmit and receive data at Gigabit rate. This block sends its data out of the FPGA through eight lines referred to a 125MHz clock, these lines arrive to the GEL card that then electrically format the data to conform to Ethernet physical layer using a PHY (physical layer converter). Once on the Ethernet the information can flow directly to a computer or it can go to a network, each GEL has a unique MAC address and an IP that can be configured by the user.

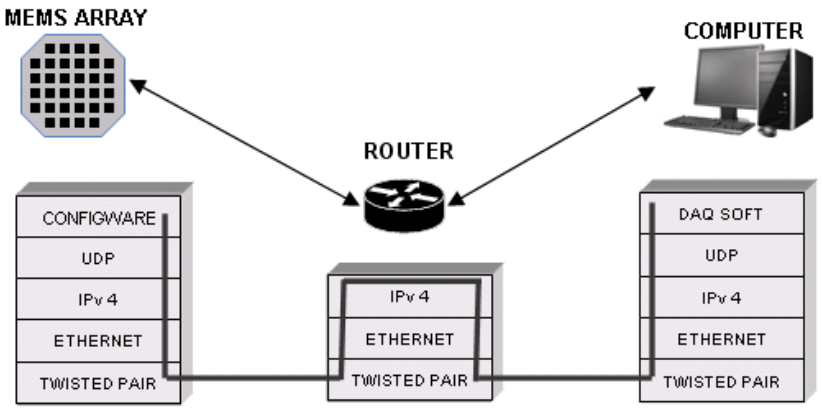

Figure 4.8. Data path and framing. This figure represents the several layers of data encapsulation that allows the data to flow from the sensor array through the Ethernet to the computer.

An integral part of the AMA system is the data acquisition and network manager software. After the information is broadcast to the Ethernet a computer connected to the network is capable of retrieving the data using the CAPTAN data acquisition software. This software contains a custom class created for the MEMS array system, this class is able to interface with and program the array by sending commands through the Ethernet. Commands sent by the software are interpreted by the NPCB and forward to the array. The software is also capable of processing and displaying the data broadcasted by the AMA in real-time due to the wide link bandwidth.

Another important feature of this system is its networkable nature which allows the system to be highly scalable. It opens the possibility of networking the microphone array board creating composed arrays with almost unlimited sizes since every array board can have a singular IP address.

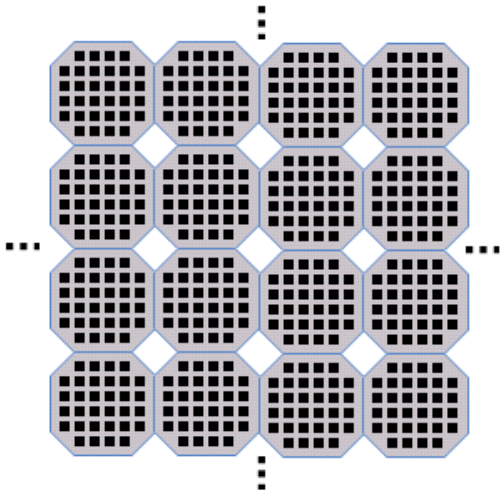

Figure 4.9. Composed array. Tiling arrays open the possibility for creating very big arrays such as the one depicted above.

Due to the octagonal shape of the board it lends itself nicely for tiling, the Ethernet connection is routed through the open squares as a result of the board geometry.

Another important feature derived from the network capability is the possibility of a composed space distributed array. Figure 4.10 illustrates how straight forward it is to increase the number of planes that the system operates in regardless of where the boards are located. The number of boards is basically just limited by the number of router channels available and the maximum speed of the router.

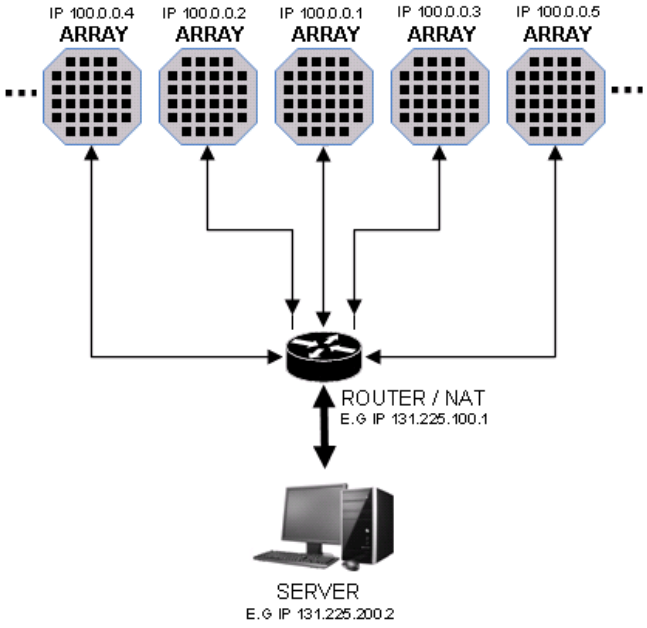

Figure 4.10. Distributed network system. The figure above illustrates the system operating with a router in a private network [Ku05].

4.4 System Operation

The system operation is controlled by software, more specifically the GUI interface custom made for the array based on CAPTAN technology. The GUI has the capability of programming the hardware of the microphone array with the following parameters; sampling rate, ADC resolution, and number of cells to be read.

At the system level the GUI can select the board IP's with which the user wishes to establish a connection. Once the GUI programs are running the single or multiple array system goes into data acquisition mode. In data acquisition mode the boards are continuously sending data to the computer and the computer pipes this data to a file on the hard disk, at the same time a second thread of the software provides visualization of the raw data on the screen if the system is set for raw data mode. If the system is set

instead to processed data mode the computer simply dumps the data in a file for posterior analysis.

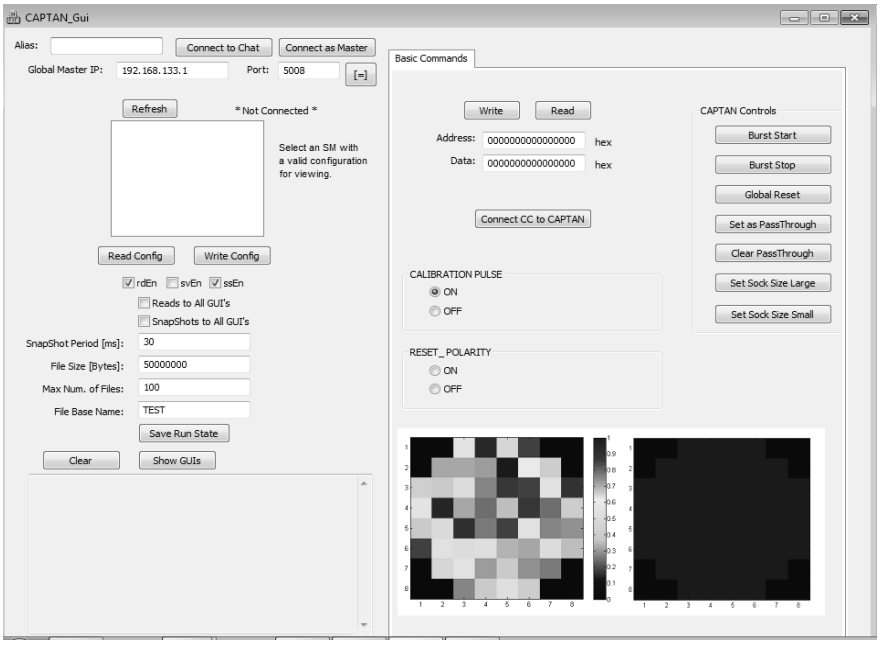

Figure 4.11. Graphical User Interface Window. Above is a picture of the custom made GUI for the array board. On the left portion of the window, network parameters, and connection status are provided. On the right side of the window the array status and configuration are displayed.

There are three modes of operation that the system can run: raw mode, processed mode and mix mode. The raw mode simply sends the following information to the computer in a 64 bits word: board ID, microphone number and 12 bit acquired data. The processed mode, on another hand, is dependent on the configware that the user designed. It can vary from simple sample averaging to FFT's and more complex analysis. The mix mode transmits both raw data and processed data words in real-time. The Ethernet communication bandwidth is 800 Mbps which far exceeds the bandwidth required for raw data mode of 22 Mbps for a 36 Ksps sample rate, the difference between the communication and raw data bandwidth allows the simultaneous transmission of raw and

processed data in real-time. All network communications are transparent to the user and are managed by the CAPTAN system. The user must set the IP switches on the GEL board before any network communication can take place.

A case study to demonstrate the system operating in raw mode is illustrated by Figure 4.12. In this test the sound source was an omnidirectional microphone emitting a single tone tuned to 4 kHz and located at 10 cm from the array, the diameter of the speaker was 1 cm. It is also important to observe that for this test the array was not calibrated and therefore the numeric results are not absolute but relative. It is possible to observe that even with raw mode, meaningful information can be obtained by the system. The source was initially placed on the left of the array and then move to the right.

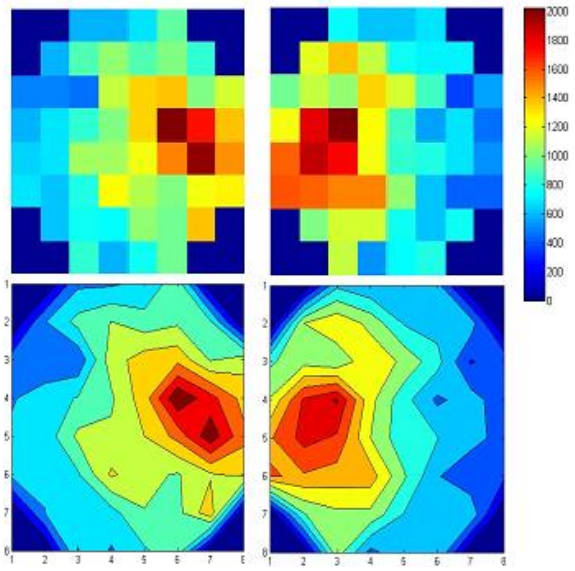

Figure 4.12. Above a representation of the system acquiring data in two different situations is presented, on the right a sound source was placed 20 cm to the left of the center of the array and centered on the vertical. On the left a source was placed 20 cm to the right of the center of the array board centered on the vertical axis, in both cases the source was 10 cm perpendicular to the board plane. The scale is in ADC counts.

The GUI can also be modified in order to provide real time analysis when the system is running on processed/mix mode, especially if all computation is done by the NPCB board, if some of the analysis is left to be done by the computer real time operation may not be possible.

4.5 System Calibration

After the initial functional test of the system it was necessary to calibrate the AMA II array in order to be ready for use.

The first step of the calibration is to calibrate the individual gain of each channel on the array. The test stand used for this end is shown on Figure 4.13 and is composed of the array itself, a CDMG13008L speaker [Cui06] and its support.

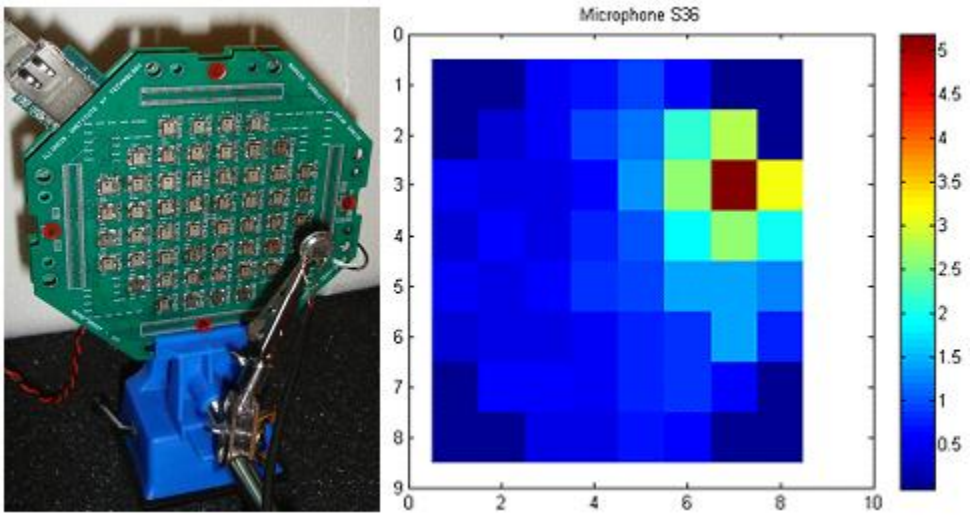

Figure 4.13. On the left the test setup, where in green is the sonic array, the red dots are the labels and in blue the setup mechanical support. On the right the mapping of the array responding to the stimulus of the microphone.

This first step of calibration consists in acquiring data with the array, positioning the speaker at a distance of 1 cm from the target microphone, and acquiring one second of data at time. This process is repeated 52 times until the whole array is scanned. The result of this process is presented in Figure 4.14 providing the mask to which all data will be corrected so that the array responds homogeneously when excited.

This process also provides an absolute calibration of the array since at 1 cm the sound pressure was set to 5 µPa, in this case on Figure 4.14 we have the normalized scale set from 1 to 0, where 1 corresponds to 5 µPa, on Figure 4.13 instead the scale was not yet normalized since the data for this calibration was still being collected and therefore the scale was in µPa.

After the absolute calibration a fast test was performed to check if the array was capable of tracking the position of objects just by measure the sound pressure profile of the array as shown on Figure 4.12.

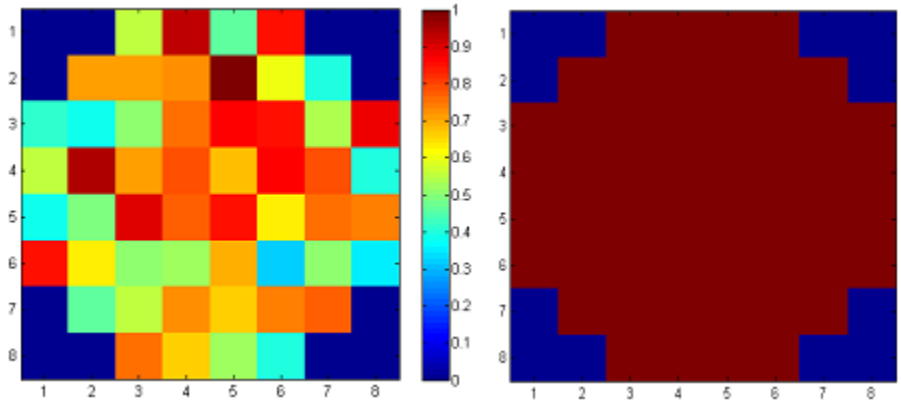

Figure 4.14. On the left the gain map of the array, the colors represent the pressure on the sensor, with red being the maximum pressure and blue the minimum, on the right the array digitally equalized.

The next calibration performed used a chirp waveform ranging from 100 Hz to 8 kHz to extract the frequency response of every individual channel. As expected the frequency response is pretty homogeneous throughout the array. Figure 4.15 shows the frequency mean response of all channels of the sonic array, showing also the channels with maximum and minimum gain. The high pass response of the front end is due to the poles introduced by the amplification stage.

Another important factor that must be analyzed when studying the frequency response of the array is the channel to channel variation of the frequency response. Basically the frequency response of every channel is compared to the mean frequency response of the array presented in Figure 4.15, and then the difference in gain is plotted.

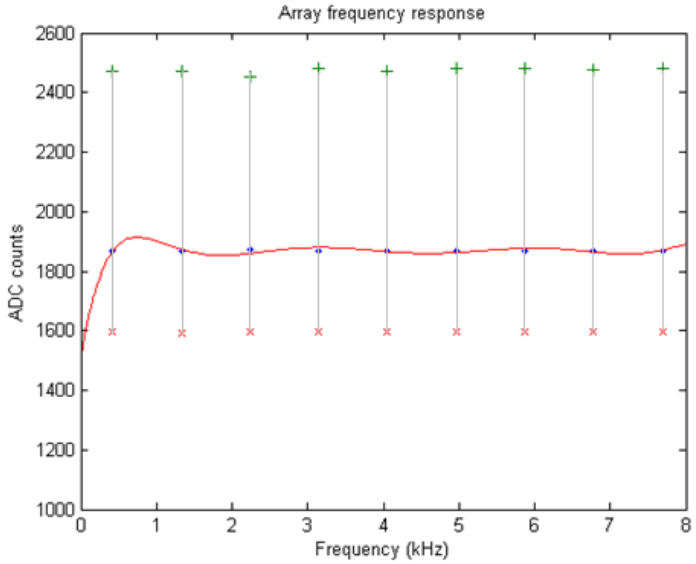

Figure 4.15. Frequency response of the array with the mean represented by the red solid line, the maximum and minimum gain are represented by the bars, with the maximum in green and the minimum in red.

It is possible to observe on Figure 4.16 that the frequency response between channels does not vary much with only channel 24 having a difference of 20 ADC counts, this is

less than 5% of the full scale of the ADC and it is by far the worst case. The mean difference between channels is 2.4 ADC counts and the mean is heavily influenced by channel 24.

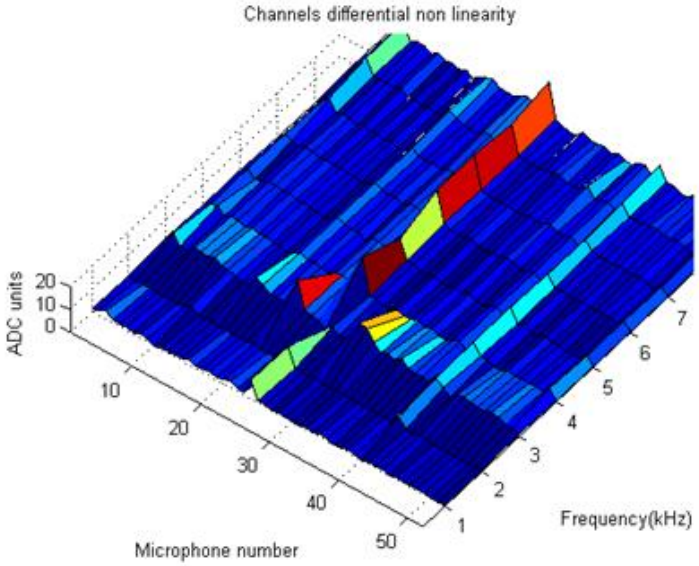

Figure 4.16. Channel to channel differential frequency response, where blue is the minimum distortion and red the maximum, in yellow and green moderated distortion.

4.6 Conclusions

This chapter describes the development and capabilities of a scalable microphone-array-based system that can be used to improve sound processing and acquisition for several types of applications. The system uses MEMS microphones arrays associated with the CAPTAN scalable architecture in order to deliver a powerful real-time signal processing and acquisition platform.

The system’s capabilities, and the many possible configurations that the system can take were explored, and connected to possible applications such as sound source separation, sound tracking, and sound imaging.

The system’s implementation and operation were also discussed in this chapter where its most important features and characteristics were exposed.

CHAPTER 5

SOUND SEPARATION AND TRACKING

5.1 System Applications

The system presented here has the potential to be used in a broad range of applications due to its real-time processing power and scalability. The scalability and processing power are closely related, since the system has a unique capability of stacking NPCB boards on the same node provides unprecedented processing power per node. The possibility of networking the system further improves its processing power.

The main applications foreseen by the author are those that will take advantage of the above mentioned characteristics of this system, such as multiple sound source separation and localization [Nak02].

Multiple sound source separation and sound localization can be very useful for applications such as teleconferences and cell phones, and can be achieved by this system through techniques such as Independent Component Analysis and beamforming [Hod80]. For sound location signals acquired by each microphone are decomposed and split in information about frequency, phase and magnitude. A beamforming algorithm can then dynamically adjust the phase of the microphones and combine it with the magnitude in order to give directivity to the array and in this way scan the array's acoustic horizon on the horizontal and the vertical axis. This can provide the location of multiple sources in space and time making use of the massive processing power of the system [Tur08]. The main challenge in this approach is to devise an algorithm that is fast enough to be able to combine this massive amount of information and compute the result in real time.

Another possible application to this system is ultrasound ranging, which can be applied in robotic visualization and provide sound orientation to industrial robotic arms [Nak02]. In this application the array's central transducer is used as an ultrasound source emitting frequencies up 60 kHz, the signals emitted by the speaker then bounce back from the obstacles allowing the distance to the object to be precisely calculated by the time difference between the signal emitted and received by the microphones, furthermore the phase difference between the microphones will further improve obstacle localization providing the size and the shape of the object. Processing speed is fundamental in this kind of application since the robot must make decisions based on the information provided by the array. The AMA system provides the unique ability of moving data very fast from the acquisition layer to the processing layer due to its four 64 bits bus delivered by the CAPTAN architecture.

5.2 Sound Separation

The performance of the system integrated with the Independent Component Analysis algorithm previously illustrated, will now be evaluated with the purpose of source separation.

A set of tests especially design to evaluate the system performance towards source separation were created and performed. The efficacy of the system was tested for a diverse set of situations involving multiple sources with background noise. The system was tested to distinguish up to four different sources at the same time. Sources of different levels of complexity where used to provide a very good picture of the system's capabilities in distinguishing different types of signals. The efficiency of the system

versus the number of sensor channels was benchmarked and studied. Another important test carried out was on the system's performance in different scenarios of spatial disposition of the acoustic sources.

One key feature missing from the ICA algorithm utilized [Hyv99] is the capability of automatically interpreting the results coming out from the algorithm. This capability was added in this work specifically to work with this system and it will be discussed with the first test.

All of the following tests were carried out with the array sampling frequency set to 36 Ksps, all channels of the system where continuously digitizing, but the algorithm just used a fixed set of microphones depending on the test, all sets are listed on Figure 5.1 and the numbering convention used to identify specific microphones is shown in Figure 5.2.

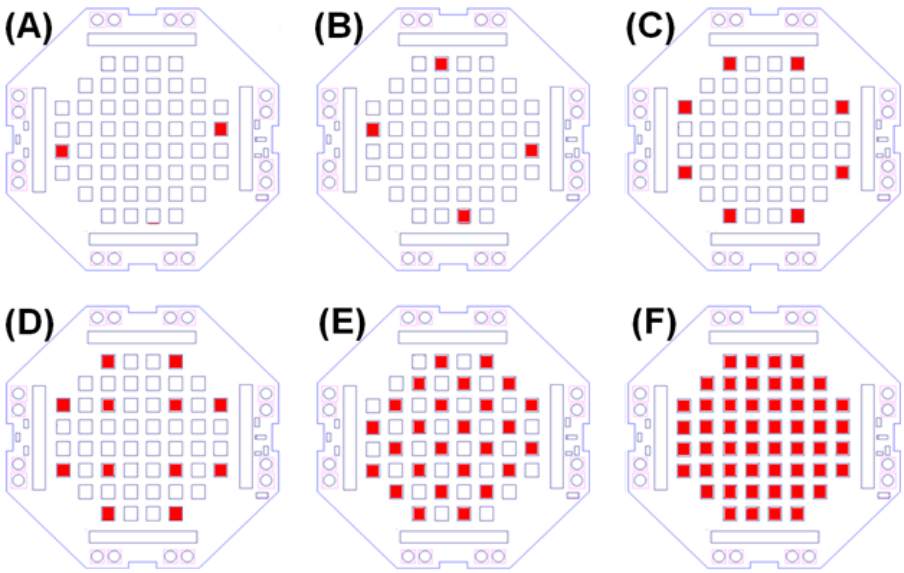

Figure 5.1. Test configurations, in red are the microphones utilized. Configuration (A) used only two microphones, (B) make use of four, (C) eight, (D) twelve, (E) twenty six and in (F) all microphones where used.

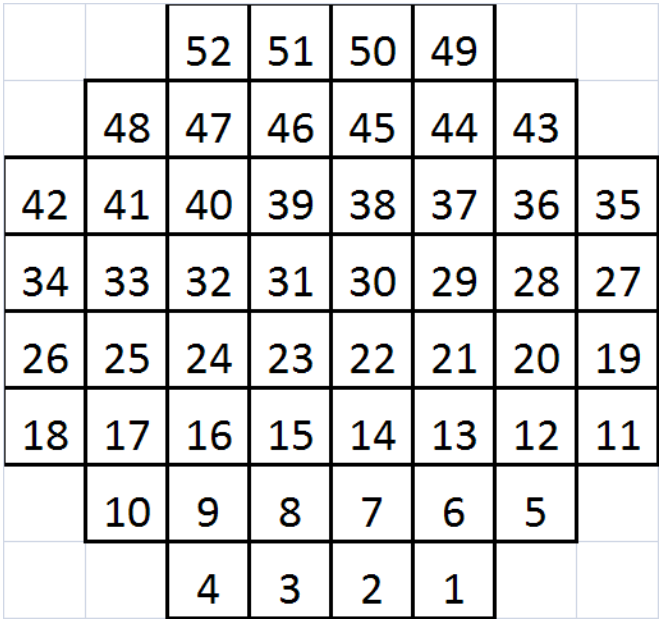

Figure 5.2. Numbering convention for the microphone array.

5.3 Source separation and evaluation algorithms

Independent Component Analysis was the algorithm of choice for this research due to its versatility and generic use in dealing with mixed signals. Independent component analysis or ICA is a mathematical technique that has as its primary objective the recovery of the original set of random variables or signals from a mixed set of random variables or signals.

The main idea behind the algorithm is that a linear mixture of independent random variables is necessarily more Gaussian than the original variables, this is a result of the central limit theorem [Ric95]. The central limit theorem states that the mean of a sufficiently large number of independent random variables with finite mean and variance, will be approximately a normal distribution.

The algorithm requires a database to establish the necessary statistics and assumes that the data variables are linear combination of unknown non-Gaussian and independent variables. The algorithm then looks for a transformation in which the components are as statistical independent as possible, this is usually achieved in conjunction with principal component analysis [Nor06]. ICA is particular in that it recovers the original sources while methods such as principal component analysis only find sets of the signals that may be locally uncorrelated but not necessary globally independent.

This algorithm is well suited for the problem of source separation when there are mixtures of simultaneous acoustic signals that have been picked up by sensors such as a microphone array. Such an algorithm is computing intensive since it must accumulate and go through the signal samples performing complex operations. In theory the ICA algorithm can distinguish as many sources as independent variables are generated by the linear combination of the sources. In practice this means that we need at least the same number of sensors as sources plus one to correctly separate the signals. The extra sensor is needed because no practical system is noise free, this also eliminate the possible existence of a Gaussian source since the noise on the system is Gaussian and the algorithm allows at most one Gaussian source as previously explained.

The algorithm implemented in this work goes beyond the ICA, it also implements pre and post processing for an efficient use of the ICA algorithm.

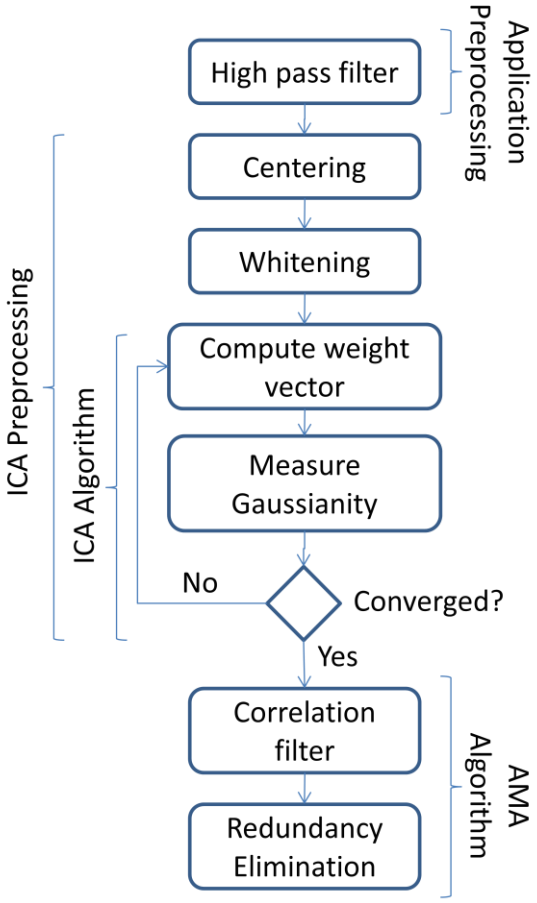

Figure 5.3. Flowchart of the implemented algorithm.

Figure 5.3 illustrates the full algorithm implemented for this system. The full algorithm encompasses a high pass filter, the ICA and the AIIA (Acoustic ICA Interpretation Algorithm) [Tur10c] algorithm.

The first part of the algorithm is a second order high pass filter with the corner frequency set to 100 Hz and is meant to enhance the application of source separation by filtering out low frequency noise. After that, the signal enters the pre-processing stage of the ICA algorithm. In this stage the data is initially centered, which is simply the centering of the incoming variables, and consist of subtracting the mean vector for each

variable making them a zero-mean variable. Still in the pre-processing stage the data is then whitened. The whitening process decorrelate the input signals as much as possible and equals their variance to one. This is required to get one step closer to the original signal, since the linear transformation that mixed the signals added correlation between the original source signals.

The whitening process is a linear transformation where the covariance matrix of the input signals is equalized to the identity matrix. In this case it will be given by;

$$cov(X\,Y) = I \tag{5.1}$$

where X is the new vector with the whitened data, Y is the vector containing the mixed data and I is the identity matrix. The transformation can be accomplished by several different methods; the most used being Principal Component Analysis [Jol02].

After the ICA pre-processing, the data enters the ICA algorithm itself. There are many different ways to implement the ICA algorithm, these techniques involve maximizing the non-Gaussianity of the vectors feed to the algorithm. This is the basis of the ICA algorithms since the algorithm expects non-Gaussian sources for input signals.

The technique applied on this work for maximizing the non-Gaussianity of the input variables was negentropy. The concept of negentropy is based on entropy of the variables being processed. The more random a variable is, larger is its entropy. Gaussian variables have the largest entropy of any random variable with the same variance [Com94]. Therefore, negentropy can be used to measure the distance to normality that is

a measure of non-Gaussianity. If a random variable is Gaussian its negentropy will be zero.

Mathematically negentropy is expressed by the difference between the differential entropy of the Gaussian random variable of the same covariance matrix as the random variable being processed and the differential entropy of the random variable itself,

$$J(Y) = H(Y_{gauss}) - H(Y) \tag{5.2}$$

where Y is the vector containing the data and Y_{gauss} the Gaussian vector with the same covariance matrix as Y and H the differential entropy. The differential entropy is defined as

$$H(Y) = -\int f(Y) \log\big(f(Y)\big)\, dY \tag{5.3}$$

$f(Y)$ is the density function of the random variable Y.

Negentropy gives an excellent measurement of non-Gaussianity, however it is not practical to be implement due to the complexity of the calculations involved, therefore a formula that gives an approximation of negentropy based on [1] was chosen, where:

$$J(y) \approx \left[E\{\log(cosh(y))\} - E\left\{\log\left(cosh(y_{gauss})\right)\right\}\right]^2 . \tag{5.4}$$

The next step now is to maximizing the non-Gaussianity of the *Y* vector containing the mixed signals, to do that vector *X* is created. Vector *X* is initially identical to *Y* and is recalculated by using the follow expression

$$X = W^T Y \tag{5.5}$$

where *W* is a weight vector that was initially chosen and *X* the new vector. The algorithm then proceeds into an iterative process making use of the expression;

$$D_n = J(y_n) - J(y_{n-1}) \tag{5.6}$$

where D_n is the negentropy direction, $J(Y_n)$ the current negentropy and $J(Y_{n-1})$ the last measured negentropy. If D_n is increasing it means that the non-Gaussianity is increasing on the other hand if D_n is decreasing the Gaussianity is increasing. Therefore the algorithm will iteratively look for a new weigh vector *W* that will always increase the non-Gaussianity of *X*. The algorithm proceeds until it passes a threshold for D_n defined by the user that will mean that the algorithm converged. This new vector *X* then contains the estimated original signal.

The ICA algorithm can provide redundant information and non optimal solutions. Most of the time ICA will provide redundant information when the sensor array feeds the algorithm more signals than real acoustic sources exist in the environment on wish the system is immersed in. In principle the ICA algorithm will always generate the same number of outputs as the number of inputs. This is not always the case in this particular

system because in order for the algorithm to quickly converge, a limit to the number of iterations was set, so if the limit is reached the ICA output is not generated, making it possible that the number of signals generated by the ICA algorithm will be either equal to or less then the number of real signals. The limit on the number of iterations also increases a problem that is already present in the algorithm, non optimal conversion.

In order to address the two problems above mentioned the Acoustic ICA Interpretation Algorithm (AIIA) [Tur10C] was developed specifically for this system and is not guaranteed to work on other applications.

The AIIA algorithm is feed with the output vectors X_n from the ICA , it first try all combinations for correlation. This is achieved by applying the Pearson product-moment correlation coefficient also known as population correlation coefficient and given by

$$\rho_{X_n,X_m} = \frac{cov(X_n X_m)}{\sigma_{X_n}\sigma_{X_m}} \tag{5.7}$$

where ρ_{X_n,X_m} is the population correlation coefficient, X_n is the ICA variable that we want evaluate against a second ICA variable X_m, and σ_{X_n} and σ_{X_m} the respective standard deviation. The number of combinations that this algorithm will calculate is given by $\binom{t}{2}$ where t is the number of ICA variables and m and n the index of the variable. The population coefficients are then compared with the population correlation coefficient threshold variable, set by the user, if the value is above the threshold, the signals are considered different, if bellow, the signals are considered the same. If the signals are

considered the same the second part of the algorithm is triggered choosing which of the same versions of the signal will be used.

The algorithm then gets all the signals that were considered to be equal and chooses those that have the smallest sample to sample mean derivative. This is done because, in this system, it was observed that signals with multiple copies of themselves originated from the ICA algorithm will be very similar, but will contain different levels of high frequency noise. This technique chooses the least noisy of the signals.

Another integral part of this algorithm is the power to identify variables that are mostly composed of noise or highly uncorrelated samples and can fail to be filtered out by the first part of the algorithm and therefore yield fake results. This is done by using the sample correlation coefficient derived from the Person's correlated coefficient and given by

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i-\bar{X}}{\sigma_X}\right)\left(\frac{Y_i-\bar{Y}}{\sigma_Y}\right) \tag{5.8}$$

where n is the sample number, X_i and Y_i the sample pair, σ_x and σ_y the standard deviation of the variables X and Y respectively. $\bar{X}$ and $\bar{Y}$ are the respective means of X and Y. The value of r is then compared with the sample correlation coefficient threshold set by the user, the signal is then deemed relevant or not, the user can always set the threshold to zero and bypass this step.

5.3.1 Multiple Signal Separation Evaluation. The objective of the present test is to probe the capability of the system in separating multiple sources and then evaluate which

of the signals coming rom the ICA algorithm are relevant. The ICA algorithm can provide redundant information and non optimal solutions. Most of the times when the system feeds more signals to the algorithm than real acoustic signals coming from the environment in wish the system is immerse the system will provide redundant information.

In order to validate the AIIA algorithm and at the same time test the system's capabilities for multiple sources detection, a test was set where four sinusoidal acoustic waves are emitted by four different loudspeakers as presented on Figure 5.4.

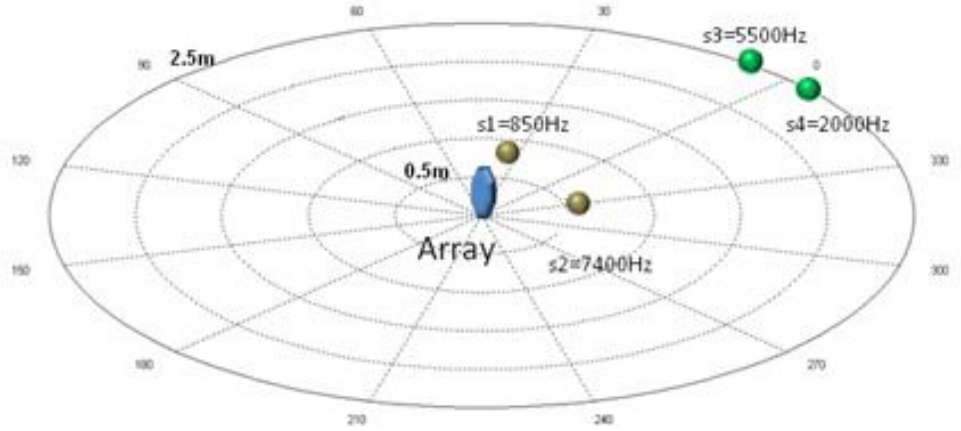

Figure 5.4. Localization in space of the four sinusoidal sources. In blue the sonic array and in green and brown the acoustic sources.

Source s1 is set at a distance of 0.7 m from the array in a 45 degrees angle to the left of the array, s2 is set at a distance of 0.7 m and 45 degrees to the right of the array, s3 is set a distance of 2.5m and 5 degrees left of the array and s4 at a distance of 2.5 m and 5 degrees to the right of the array. The frequency of the sources s1,s2,s3 and s4 are respectively 850 Hz,7400 Hz,5500 Hz and 2000 Hz.

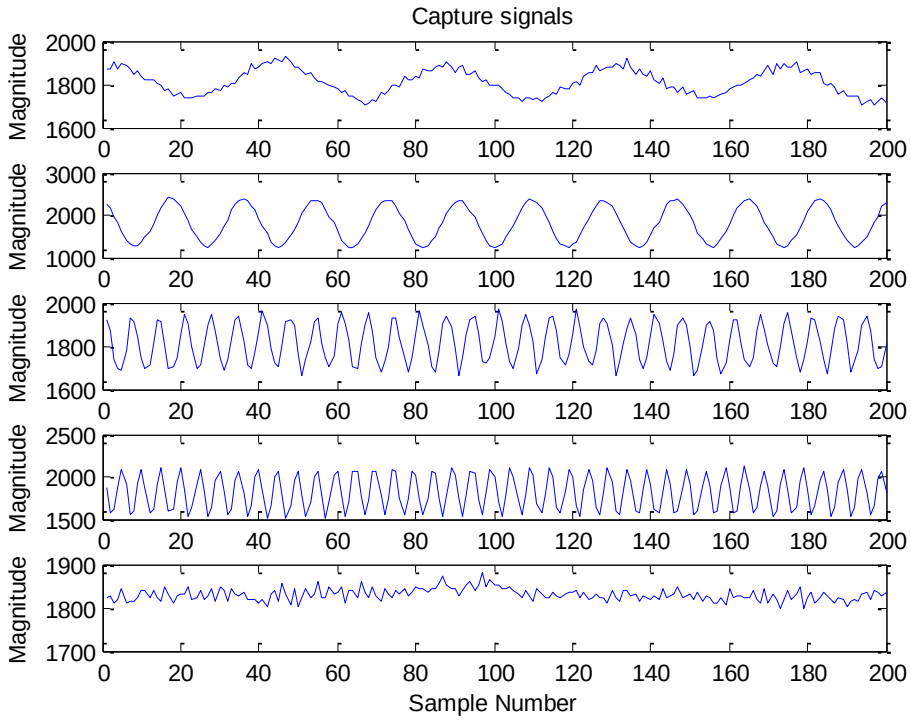

Figure 5.5. The five individually captured signals are from top to bottom s1, s4, s3, s2 and the background noise.

The first step of this experiment was to acquire the data independently for each source, one source at the time was stimulated and its output acquired. After that, the array acquired data with no source to obtain measurements of the background noise, these measurements can be observed on Figure 5.5

After the individual signals where acquired for posterior comparison, the system was activate to collect data of all sources at the same time and the result of this data acquisition is provided by Figure 5.6.

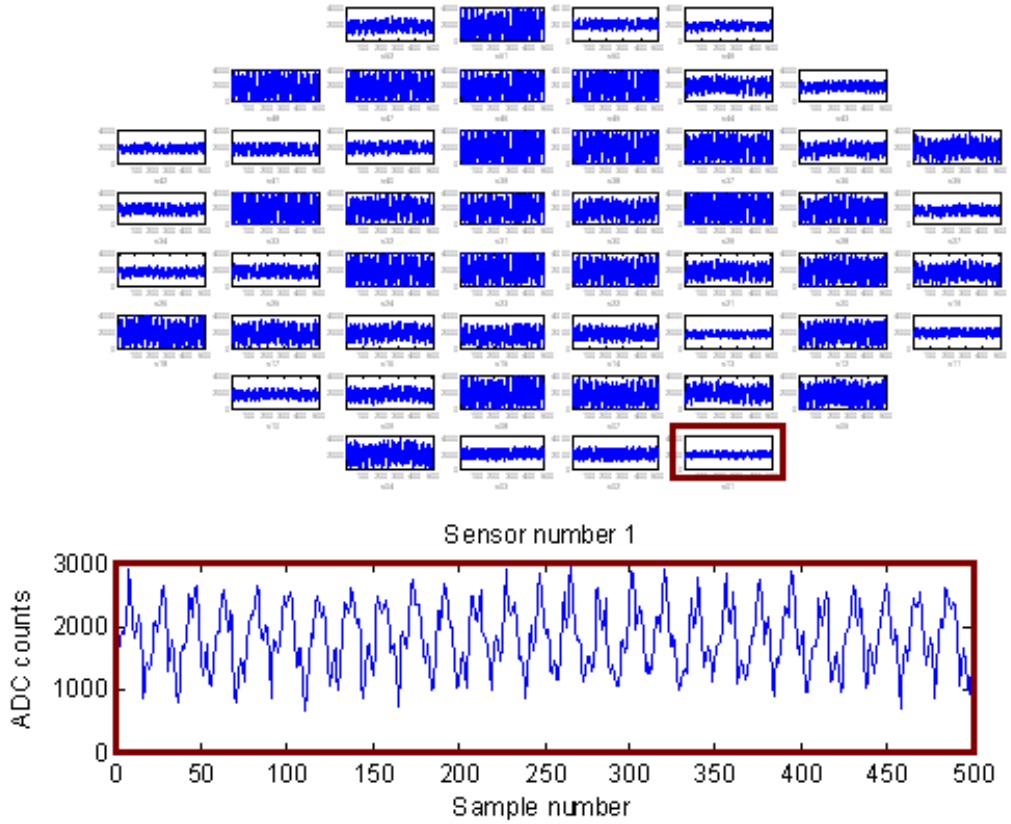

Figure 5.6. On the top, a snapshoot of the array acquiring data, on the bottom microphone channel one in detail.

After the data was acquired it was immediately sent to the ICA algorithm, that was set to configuration (A) as shown in Figure 5.1 with only two channels being processed.

The ICA algorithm returned the two vectors shown in Figure 5.7 which were then sent to the AIIA algorithm for post processing that accepted both results.

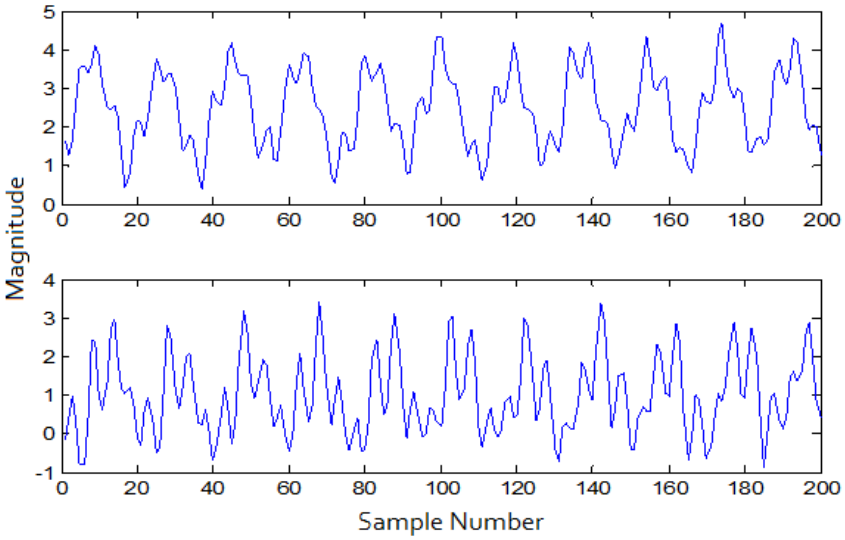

Figure 5.7. Two channels ICA algorithm response.

The AIIA algorithm could not filter out the result presented on Figure 5.7 since it passed all its criteria's and therefore a false result was accepted. This will always be the case when the number of sensors is less than the number of real acoustic sources and the only way to prevent this is to guarantee a larger number of sensors than sources.

On the next test the ICA algorithm was run in configuration (B) with four microphones, this time the AIIA algorithm selected two vectors and eliminated two, the results are shown on Figure 5.8.

From these results it can be observed that the ICA algorithm was still not able to distinguish the four original signals even with the same number of sources available. The AIIA algorithm in this case performed as expected and eliminated two similar vectors. Although, arguable the system still did not have the same number of acoustic sources as sensors due to the presence of background noise, the system found only two signals. This shows that noise and system non-linearity strongly affect the algorithm. The same

experiment was tried with perfect sine waves feed to the algorithm, and the algorithm found them all when a linear combination of the signals providing four variables was provided to the algorithm.

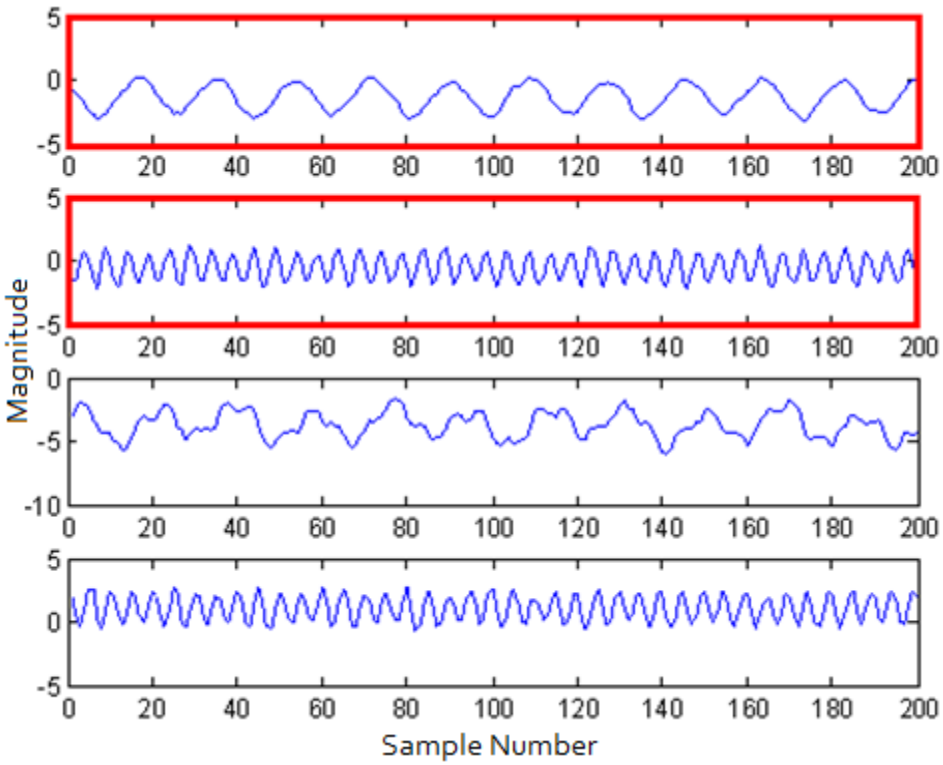

Figure 5.8. In blue, the four microphone results. The two top results in red were selected by the AIIA algorithm.

In the next test the algorithm was run with eight sensors acquiring data as shown in configuration (C), the results are presented on Figure 5.9.This was the first run of the algorithm where it was able to recover all four signals and the background noise. The AIIA algorithm performed extremely well and was able to eliminate all doubles or fake signals as the processed data shows. Signals S2, S3 and S4 where recovered with a high level of accuracy, while signals S1 had a lot of high frequency noise mixed with it.

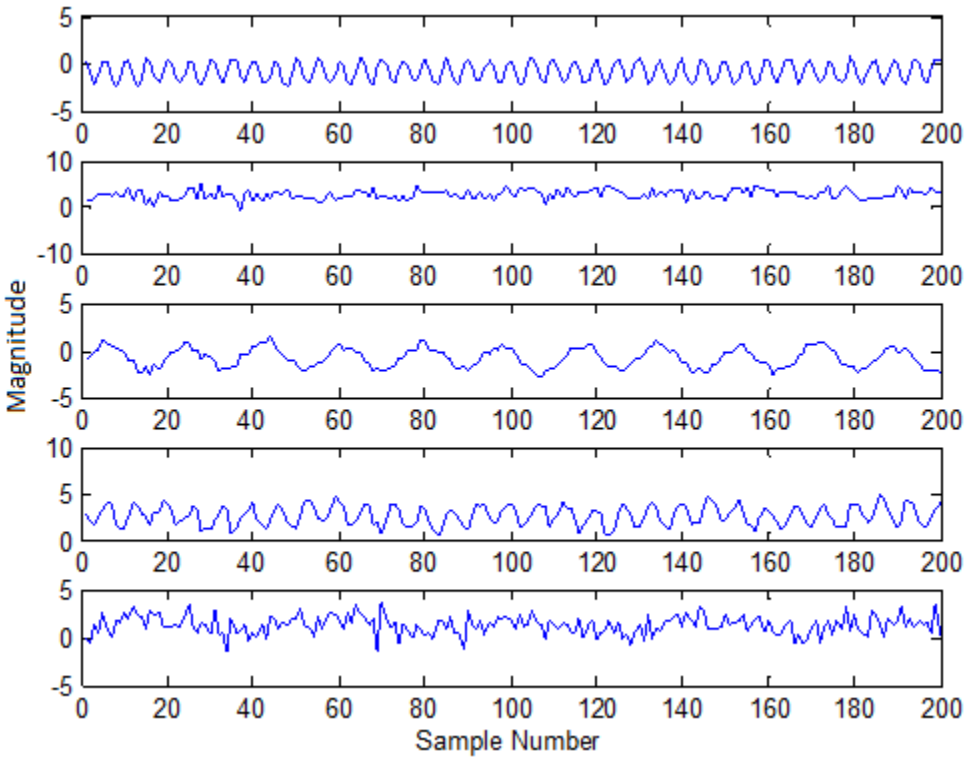

Figure 5.9. Eight microphones result from the ICA algorithm after being processed by the AIIA.

The next test was performed with twelve sources and was a challenging test for both the ICA algorithm as well as the AIIA. Both algorithms performed very well in this configuration, the results obtained by the ICA are displayed in Figure 5.10 and the AIIA selections are displayed by Figure 5.11.

This test provided the highest level of correlation with the original signals. After this test, the algorithm was run with configurations (E) and (F) and the results were very poor due to non convergence of the ICA algorithm. In order for (E) and (F) to converge the limit on the number of iterations for the ICA algorithm needed to be increased

exponentially causing a significant increase in processing time. Even after that, the results were not better than the ones reached with configuration (D).

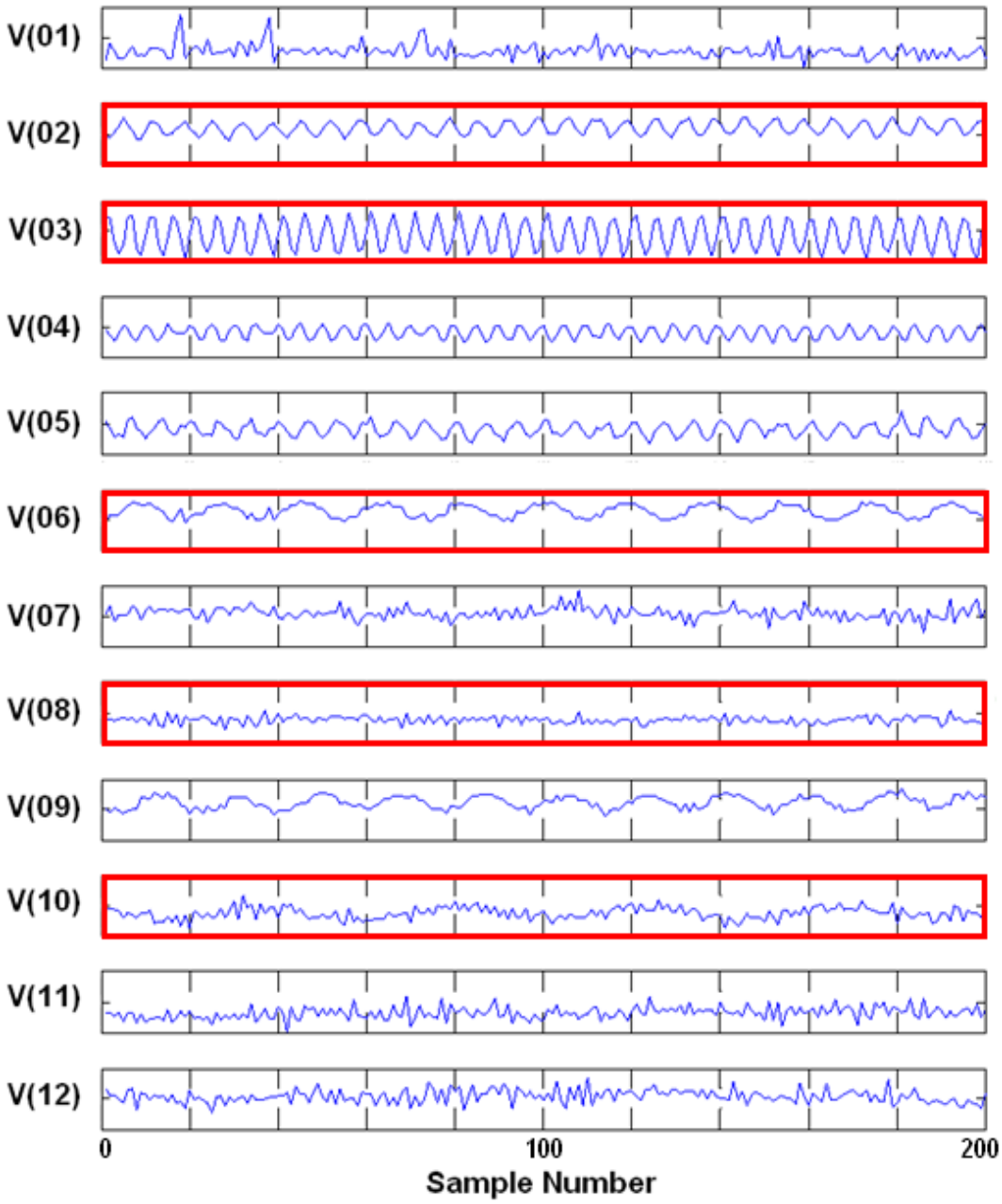

Figure 5.10. In blue, the twelve vectors generated by the ICA algorithm. Signals highlighted in red are the vectors chosen by the AIIA.

The number of sources was also decreased to three and two, the relation observed was that the number of microphones used to produce the best results in this system is equal to approximately three times the number of real sources, adding mores microphones does not depreciate the results as long as the algorithm converges.

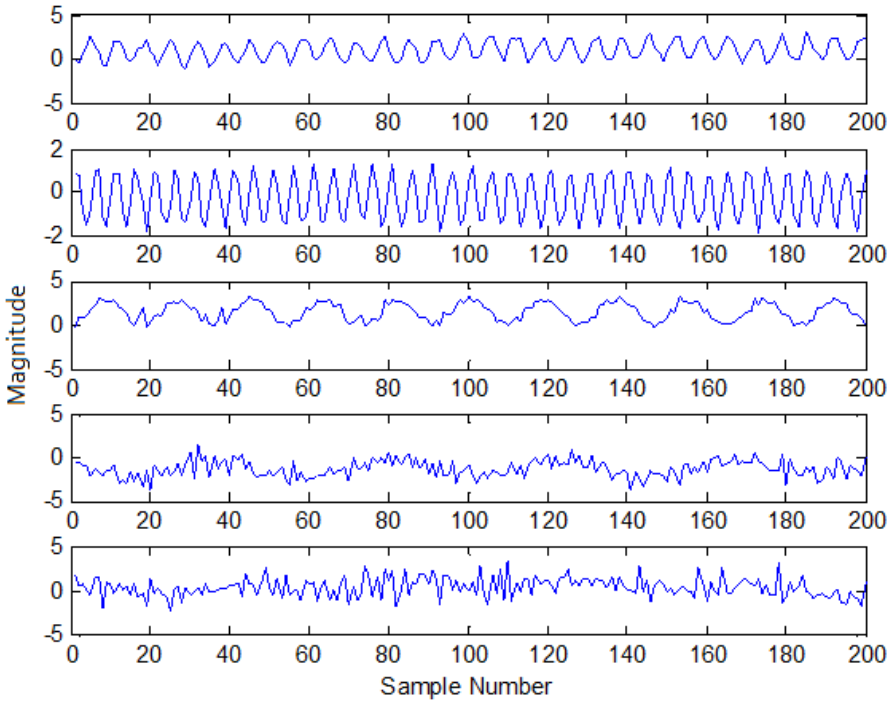

Figure 5.11. The five vectors chosen by the AIIA algorithm when twelve microphones were utilized.

The AIIA results prove that the algorithm is pretty reliable for this system as long as the population correlation coefficient threshold and the sample correlation coefficient threshold are calibrated beforehand. The calibration of these coefficients is straight forward using just a sine wave and the local background noise, usually the default values

will work without the need of change as long as the gain calibration of the array does not change. The results provided by the ICA/AIIA algorithm are illustrated by Figure 5.12.

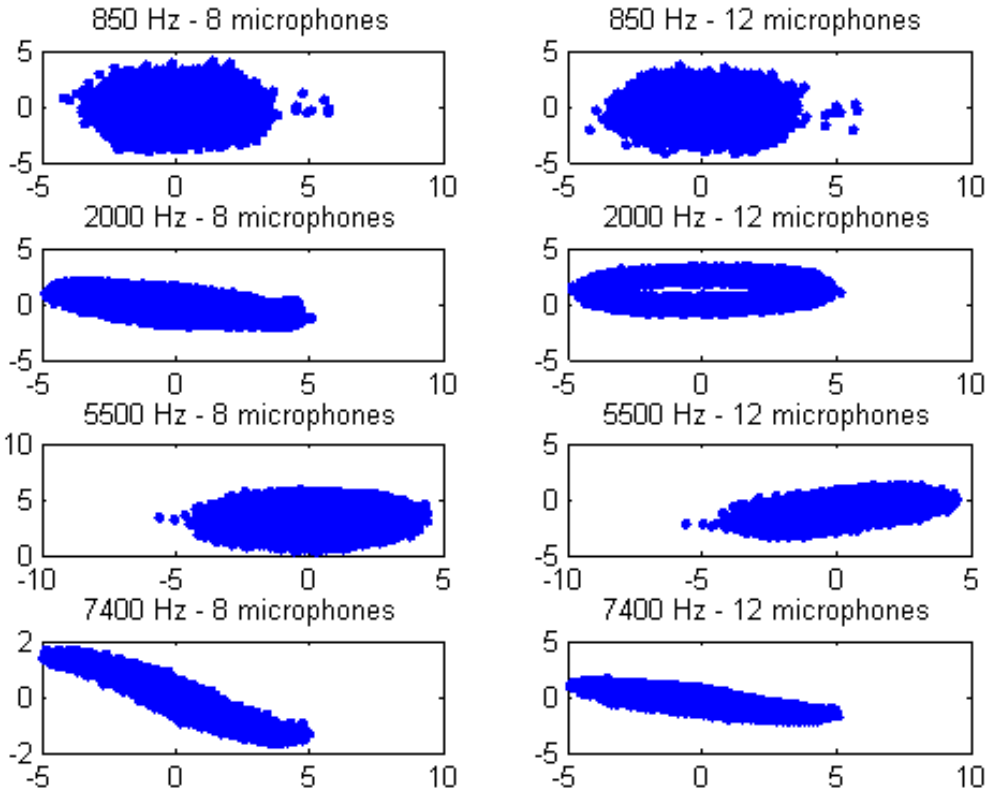

Figure 5.12. This plot graphically represents the correlation of the original signals with the recovered signals for eight and twelve microphones.

5.3.2 Complex signals separation. The purpose of the next set of tests is to demonstrate the system's capabilities and efficacy in dealing with complex signals.

The setup for this test was composed of two loudspeakers, one 30 degrees to the left at a distance of 0.7 m from the array, the other 30 degrees to the right also at a distance of 0.7 m from the array. The speaker to the right of the array was playing a recording of a Pelican and the one on the left a recording of a chicken. Figure 5.13 shows

the spatial disposition of the setup and Figure 5.14 the signals captured by the array when played individually.

The system digitized 3 seconds of data in all tests performed in this experiment and the number of samples accumulated for each signal was on the order of a hundred thousand.

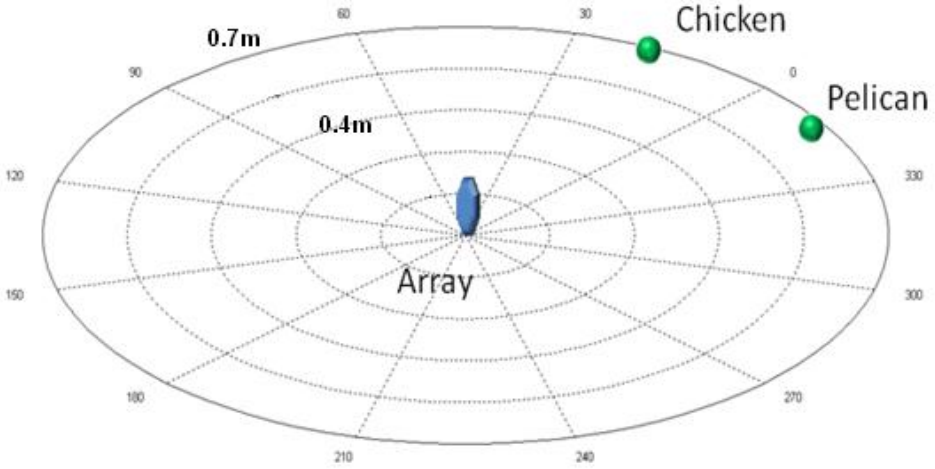

Figure 5.13. Spatial view of the setup. In blue the sonic array and in green the sources.

After the two signals where collected the setup was set to perform with both speakers on, the resulting captured signal by microphone one is illustrated by Figure 5.15. It is possible to observe that these signals are very rich in frequency components and a complete overlap of the signals was performed during the 3 seconds capture.

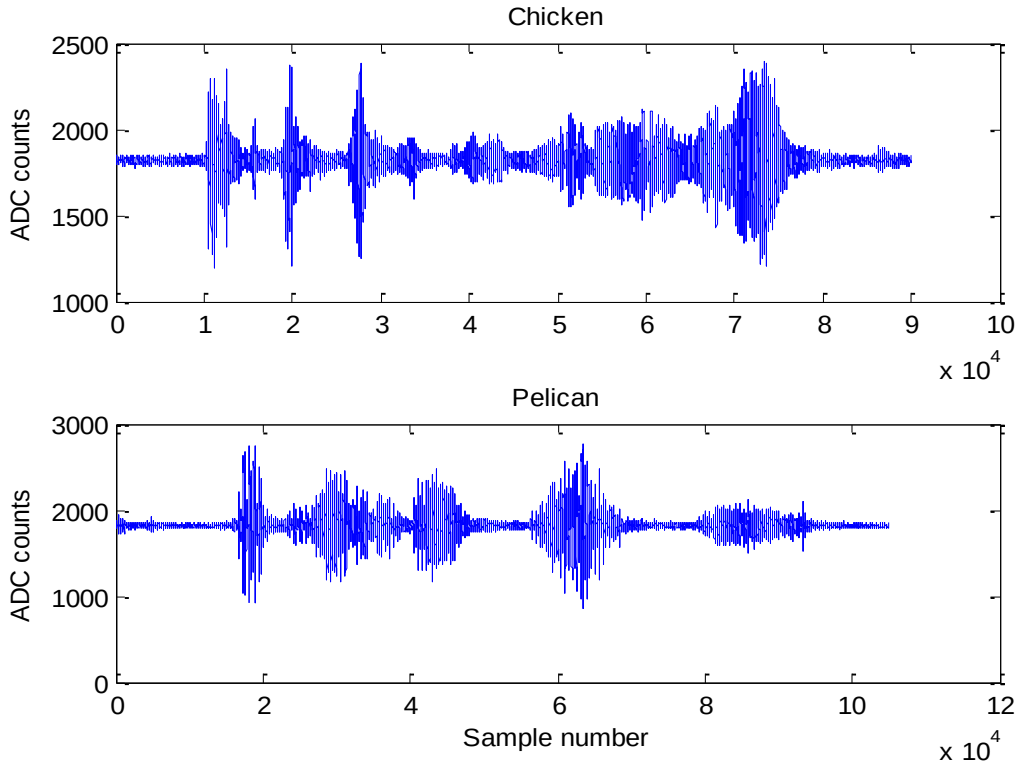

Figure 5.14. Chicken and Pelican sound digitize separately.

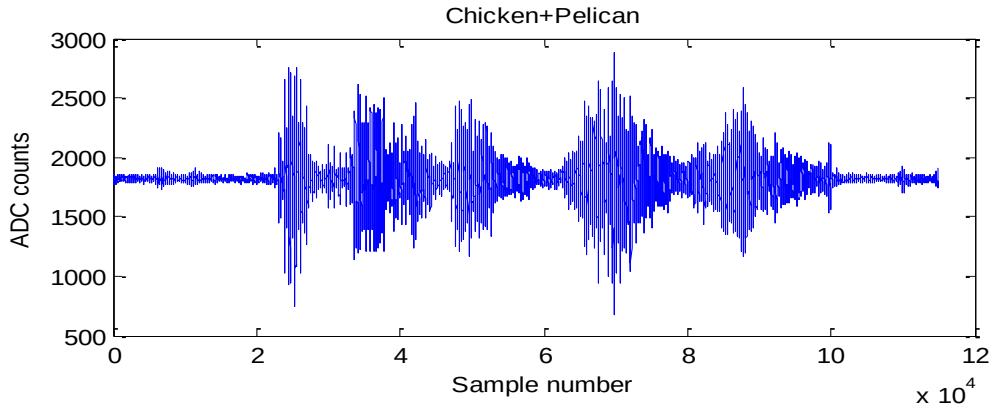

Figure 5.15. Pelican plus Chicken captured signal as received by microphone number one.

After captured, the data was uploaded to the ICA/AIIA algorithm and subjected to an eight sources signal extraction, configuration (C). The results of this test are shown in Figure 5.16.

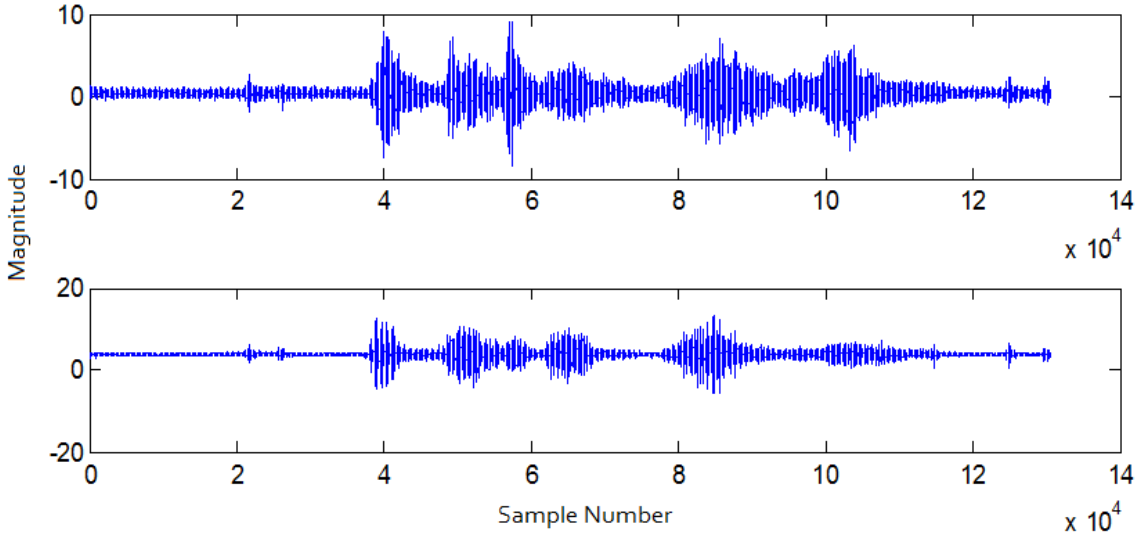

Figure 5.16. Eight microphones signal extraction using the ICA/AIIA algorithm. On the top the chicken recovered signal and on the bottom the Pelican recovered signal.

In order to verify if any further improvement was possible the signal was then processed with twelve microphones in configuration (D) and the result is shown in Figure 5.16. In order to evaluate the recovered signals, Figure 5.17 shows a comparison between the original signals and the recovered signals.

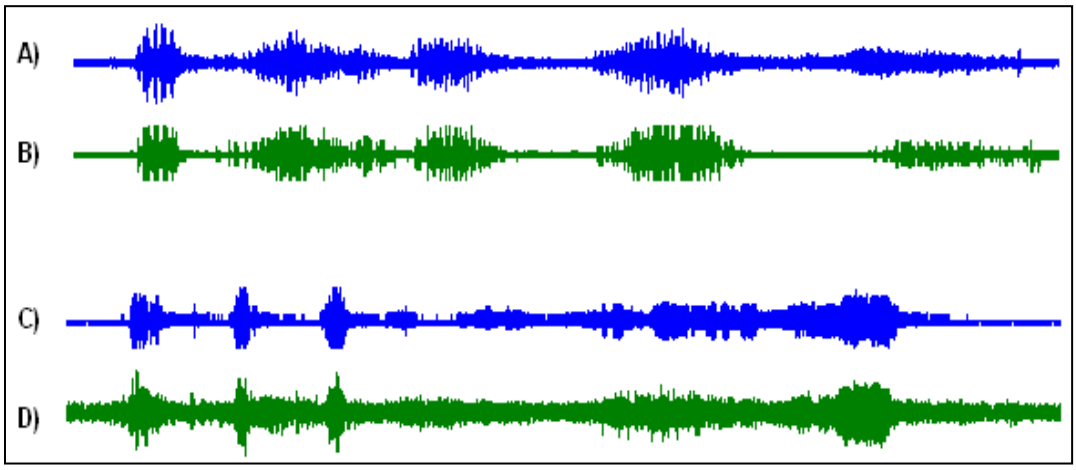

Figure 5.17. Twelve microphones signal extraction using the ICA/AIIA algorithm. (A) in blue original chicken signal , (B) in green the recovered signal,(C) the original pelican signal and (D) the recovered signal of the chicken.

As expected no further improvement was achieved since with eight microphones the number of sensors was already more than three times greater than the number of signals.

Figure 5.18 shows the plot of the joint probability distribution of the signals captured by microphone number one and microphone number fifty two, during the test performed with twelve microphones. Figure 5.19 shows the joint probability distribution of the signals recovered by the ICA algorithm.

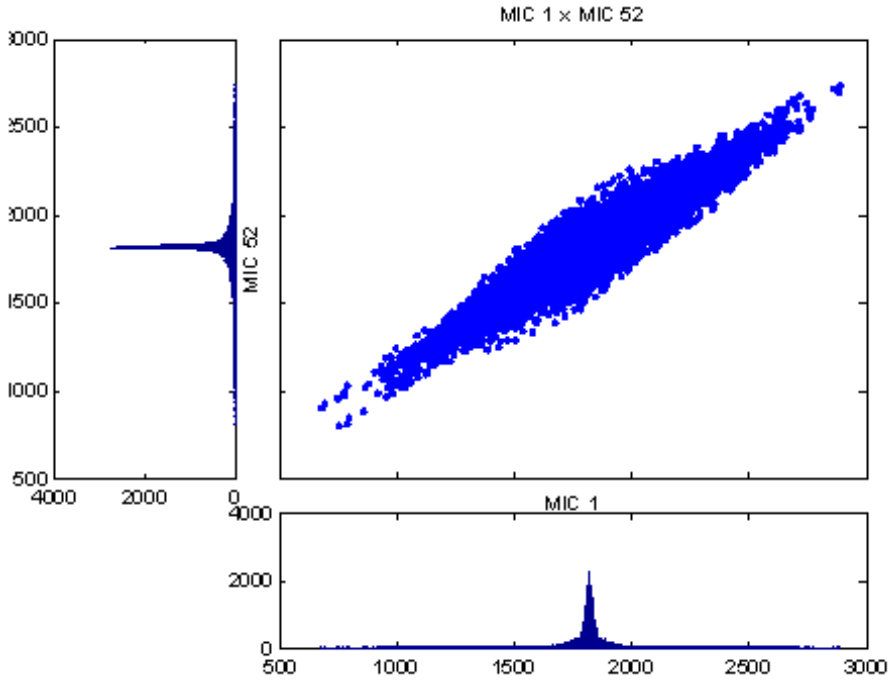

Figure 5.18. Joint probability distribution of microphone one versus microphone fifty two before processing, both microphones where used during the data processing.

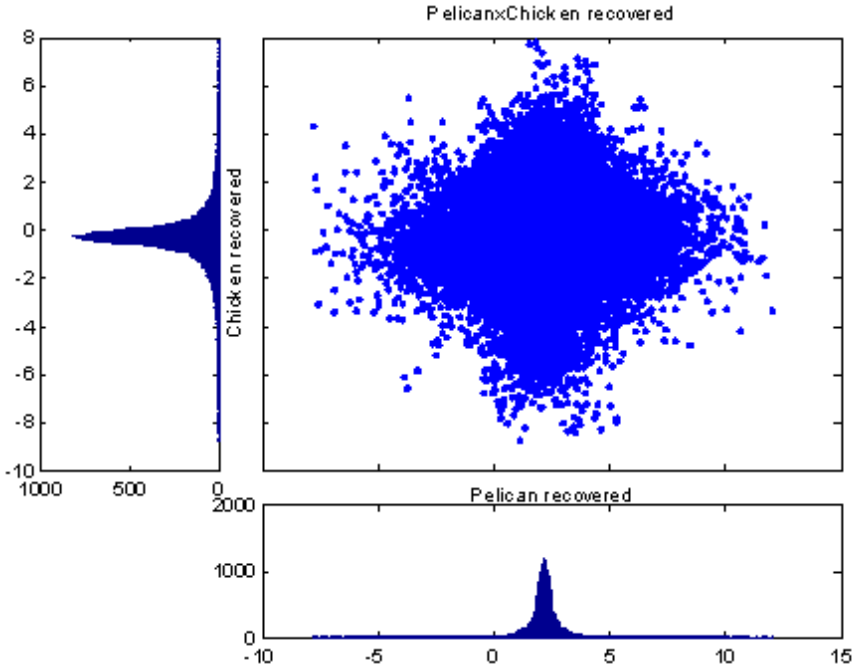

Figure 5.19. Joint probability distribution of microphone one versus microphone fifty two after processing, signal recovered with the twelve sensors test.

5.3.3 Separation for signals with broad spectrum frequency. In this test the objective is to determine the effectiveness of the system when presented with a high power broad spectrum signal. Two test setups with the same sound sources were placed in different geometric arrangements and with different time delay between the sources.

The sound sources chosen for this test were a recording of an Ak47 riffle and a recording of a low altitude pass by from an A10 jet aircraft, the low pass aircraft is the broad spectrum source and both were emulated by loudspeakers. The sources disposition were set in two different scenarios, in the first one the Ak47 source was set 60 degrees to the right of the array and at a distance of 70 cm, while the A10 source was set 60 degrees to the left of the array and at a distance of 70 cm.

In the second scenario the A10 was set straight ahead of the array and at 2.5 m distant, while the Ak47 kept the same position as in the first scenario. Both scenario setups are illustrated by Figure 5.20. In both scenarios the acquisition time was close to seven seconds, in the second scenario a half second delay was added to the A10 source in order to observer effects of interference of the signals in different times.

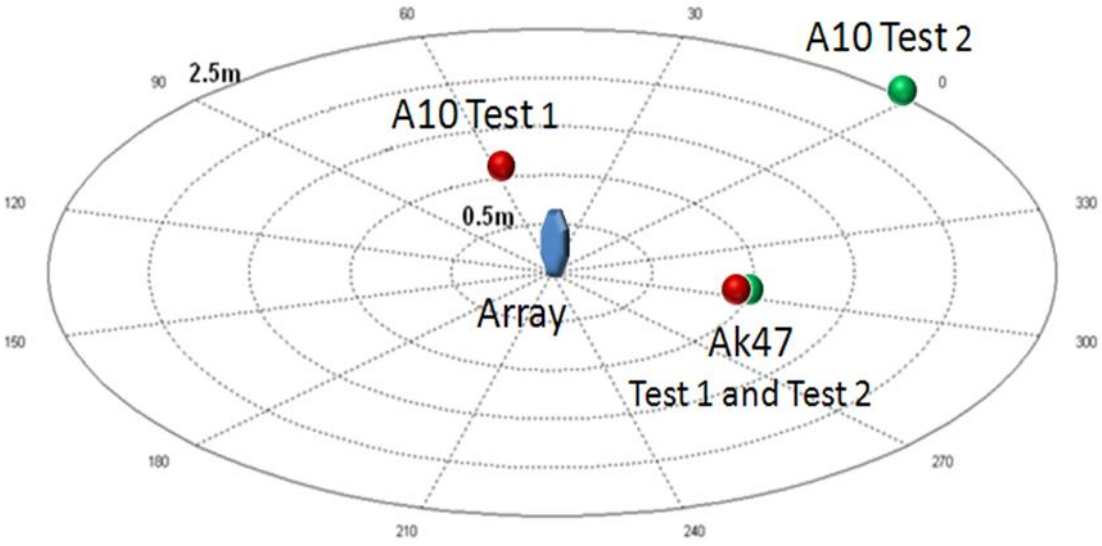

Figure 5.20. Test setup spatial arrangement. In blue the sonic array, in red sources of the first test and in green sources of the second test.

Figure 5.21 illustrates the two signals used for this test, from Figure 5.22 it is possible to observe that the A10 signal has a very broad spectrum while the AK47 a very narrow one.

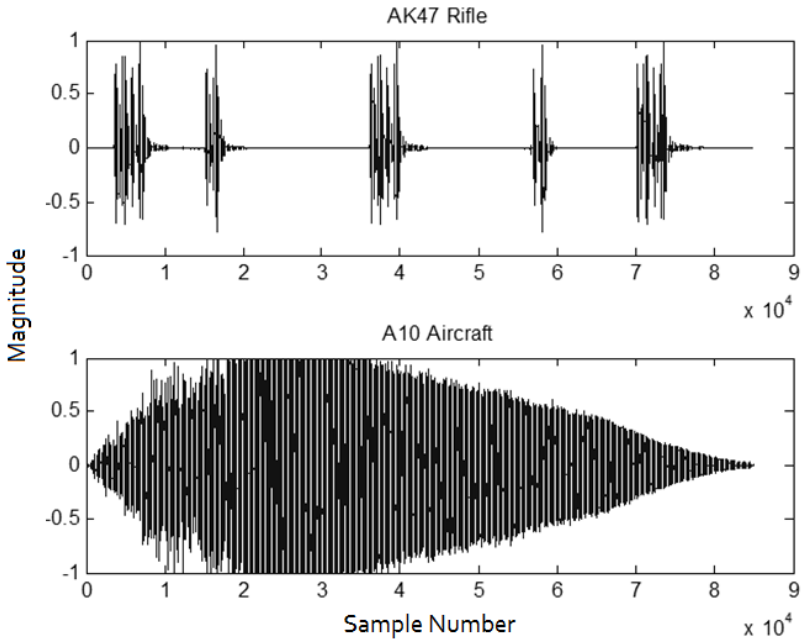

Figure 5.21. The two signals used for both tests are shown above.

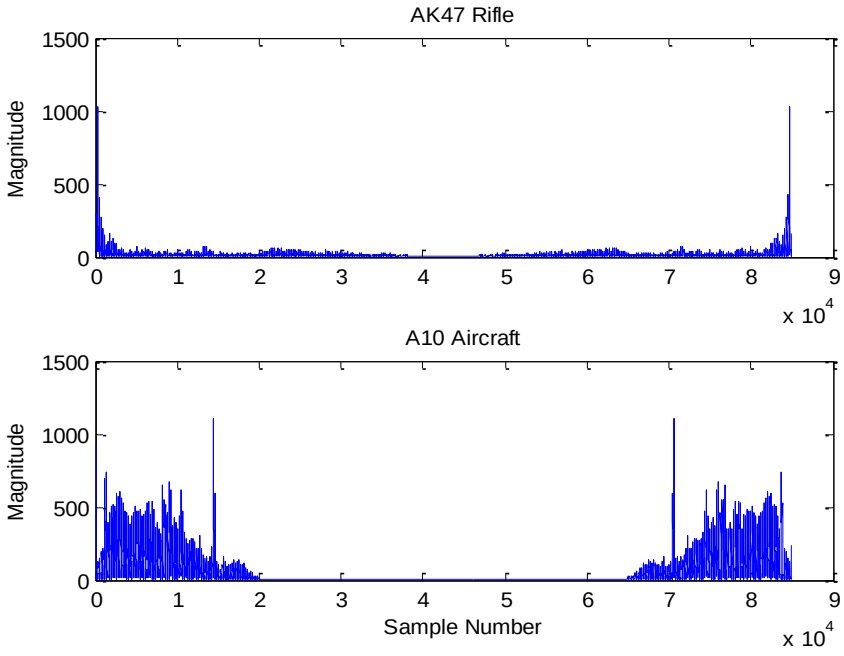

Figure 5.22. FFT of both acoustic sources.

The first test acquired data in configuration (C) with eight microphones, the signal acquired by microphone number one is show on Figure 5.23.

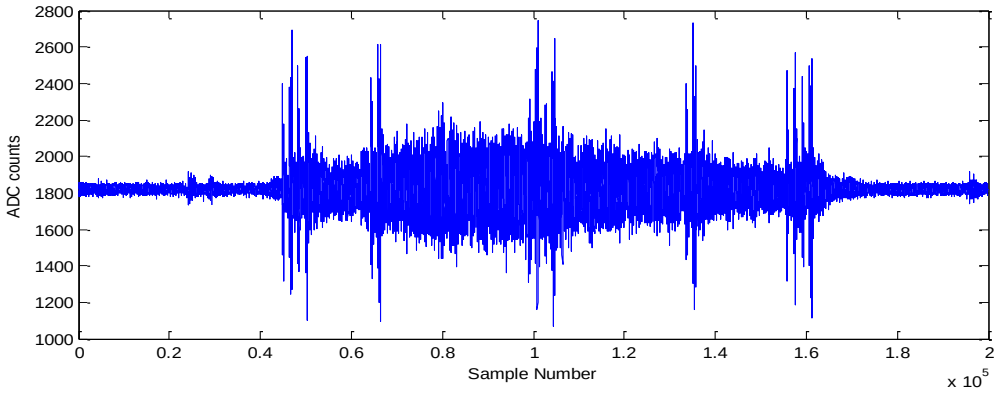

Figure 5.23. Signal acquired by microphone number one on the first test.

The results of this first test are shown in Figure 5.24. From this figure it can be seen that the recovered AK47 signal had a very high degree of correlation with the original signal, while the recovered A10 signal had a low degree of correlations with the original signal.

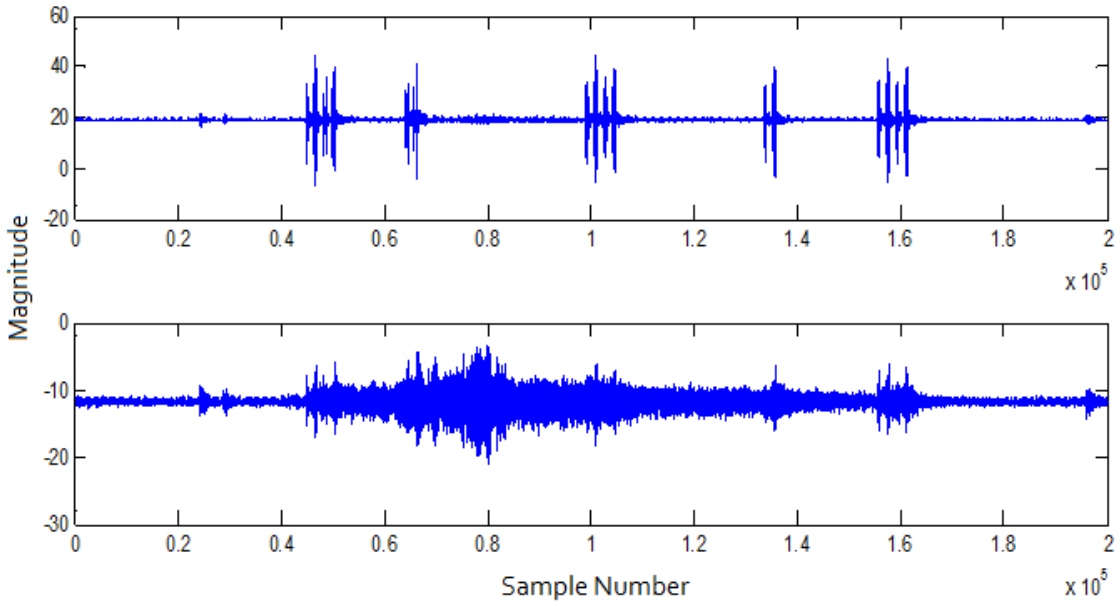

Figure 5.24. Vectors provided by the ICA/AIIA algorithm for test one.

The differences can be seen clearly when Figure 5.24 and Figure 5.21 are compared.

The second test was performed and 0.5 seconds of delay was added to the second signal resulting in the waveform shown in Figure 5.25.

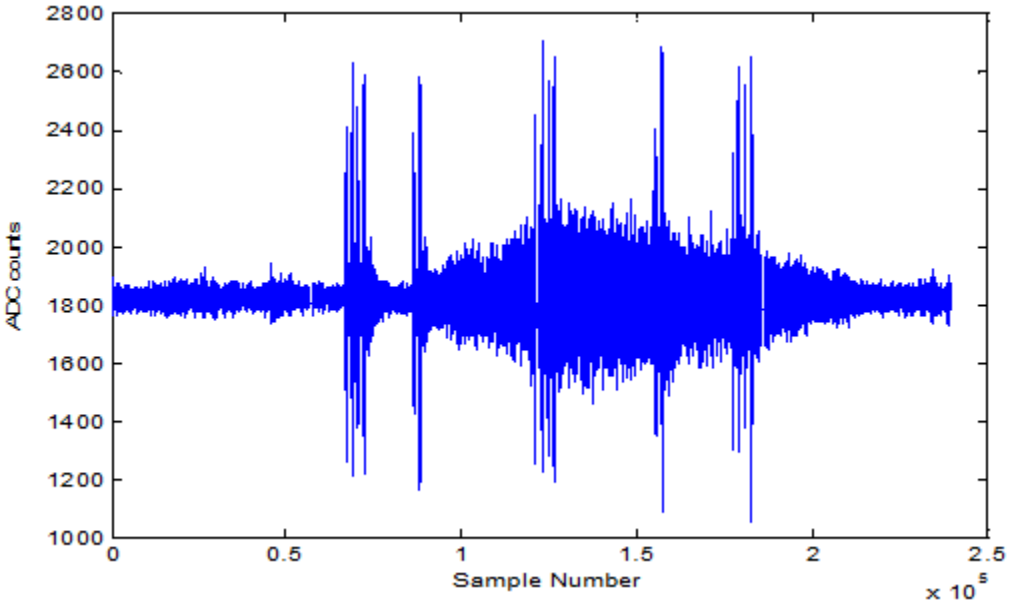

Figure 5.25. Signal acquired by microphone number one on the second test.

Again in this test the algorithm had difficulty in the recovering of the A10 signal, but was able to recover the AK47 signal to a high degree of correlation.

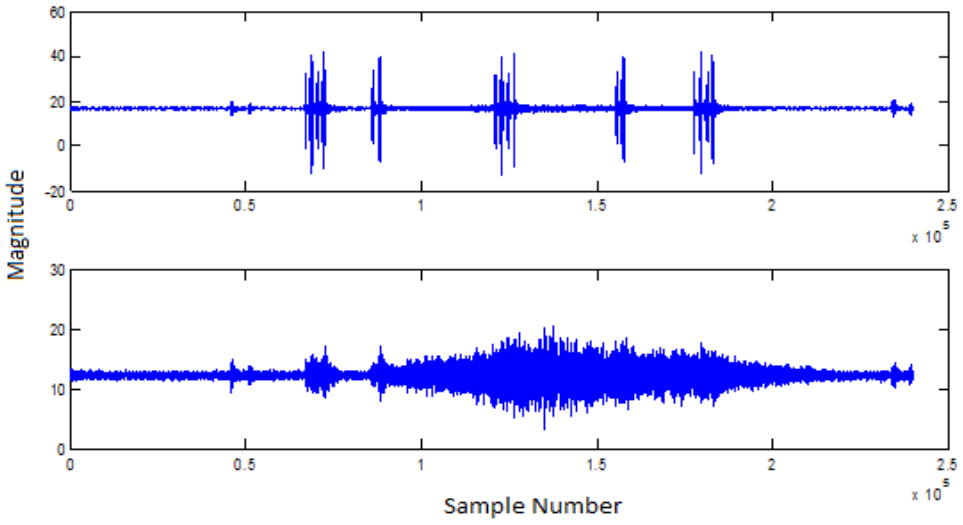

Figure 5.26. Vectors provided by the ICA/AIIA algorithm for test two.

Figure 5.26 shows the signals recovered by the algorithm from the second test, in this case the shape of the recovered signal was closer to the original, but the A10 signal still has a strong correlation with the AK47 signal. This correlation is especially prominent in the time elapsed between sample 50000 and sample 100000. The performance of the algorithm can also be observed from the FFTs of both the original and recovered A10 signals, shown on Figure 5.27.

The reason why the A10 signal is so poorly recovered is because it is highly Gaussian. This means that there are two Gaussian signals in this scenario; the background noise and the A10. As previously stated the algorithm can deal with at most one Gaussian source this is the reason why the A10 signal was not successfully recovered.

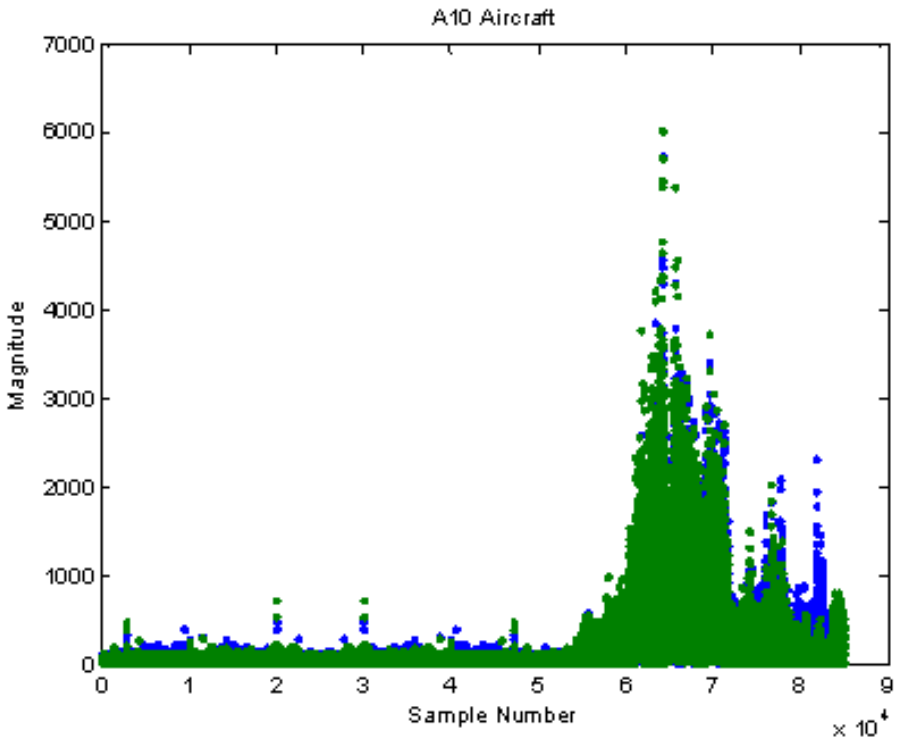

Figure 5.27. In green FFT of the A10 recovered signal on test one and in blue the A10 recovered signal on test two.

5.3.4 System used for cardiac auscultation. The final test of the system for source separation was carried out in a different environment; the system was used to perform cardiac auscultation. Figure 5.28 shows the system collecting data on the chest of the subject.

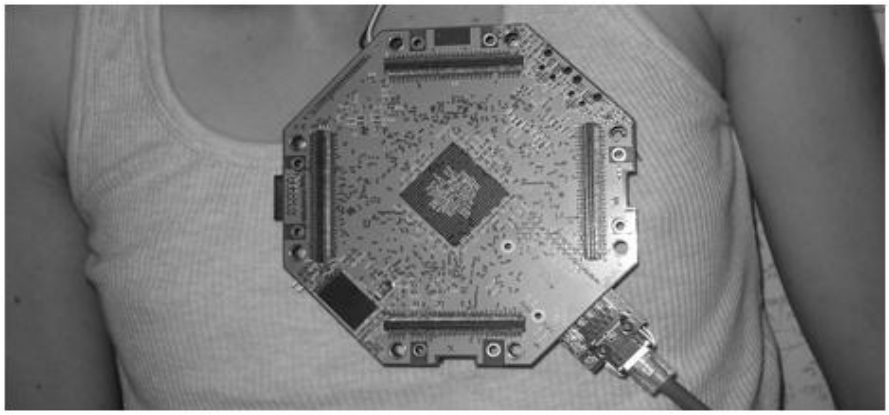

Figure 5.28. System collecting data on the subject's heart.

For this test the system was set to configuration (C) with eight microphones, a sampling rate of 36 Ksps and a total digitizing time of 11 seconds.

Each heart beat in healthy adult human beings is composed of basically two distinct sounds, that occur in sequence, the first heart sound (S1) and the second heart sound (S2). The first heart sound is caused by the Mitral (M) and Tricuspid (T) atrioventricular valves and the second heart sound is caused by the Aortic (A) and Pulmonary (P) semilunar valves. The localization of these heart valves on the human chest is illustrated in Figure 5.29.

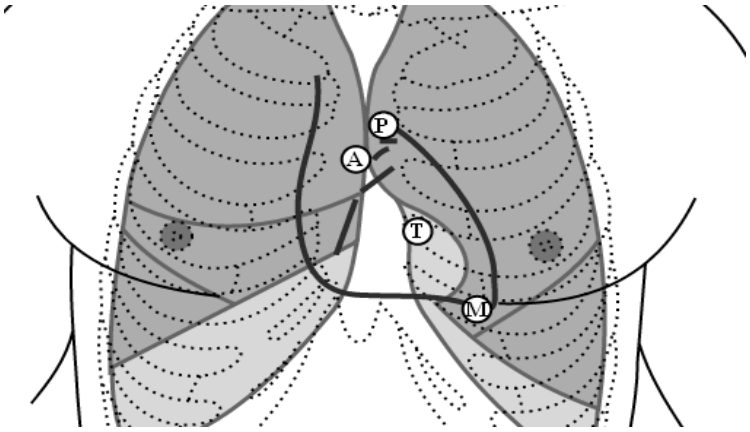

Figure 5.29. Localization of heart valves [Bic05].

The results from the digitization are shown in Figure 5.30, although 400.000 samples were captured only 200.000 are displayed on this picture for clarity. It is important to observe that this time the signals on the microphones are very different from each other, this occurs because the array is so close to target and because the target is of a similar size to the array.

The S1 and S2 sounds can be observed in detail on Figure 5.31. The S1 first heart tone precedes the S2 second heart tone.

As mentioned previously each sound is generated by the works of two valves, bringing the total number of sources to four. It is also important to mention that there are many other sounds occurring in the body at the same time, these sounds will act as background noise. For example, a deep breath during the test can add a fifth important sound source, but this was not the case during this test.

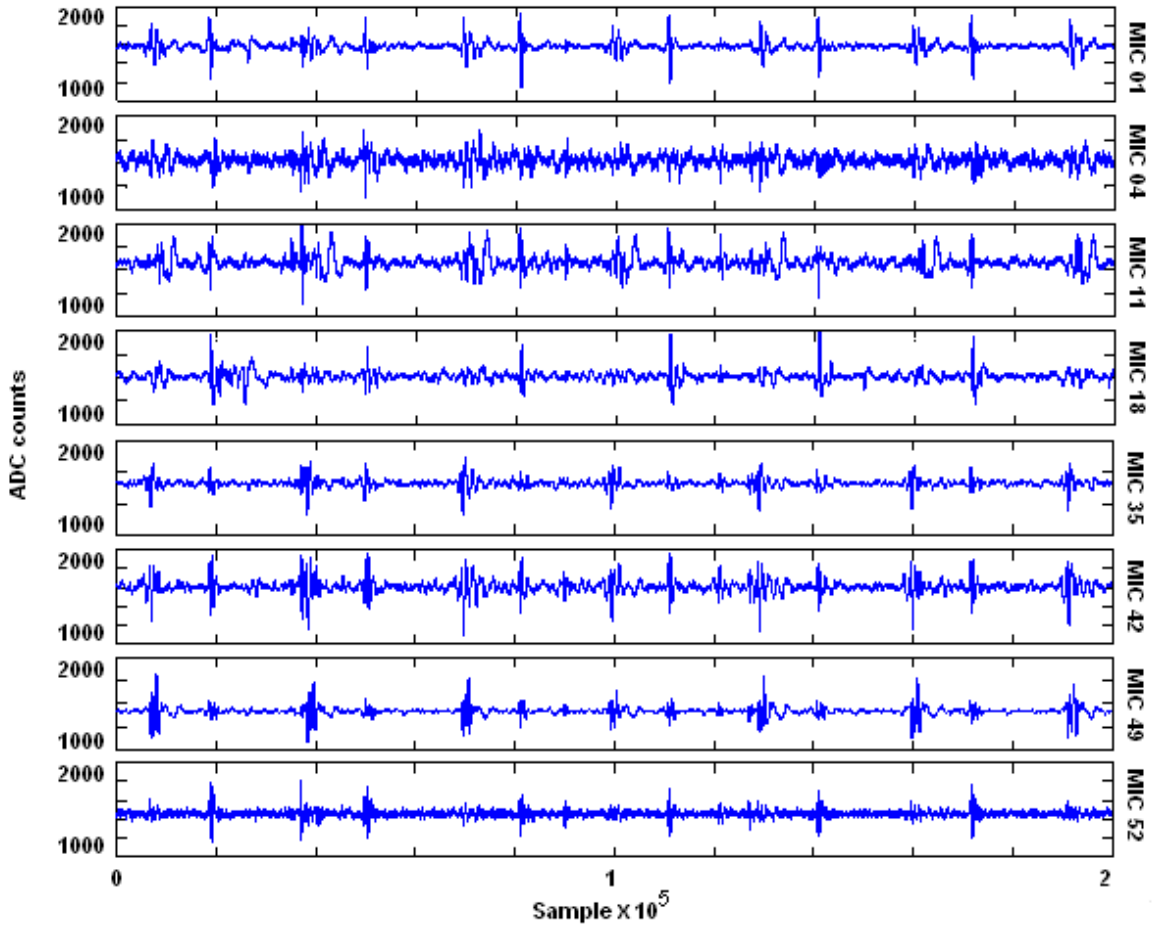

Figure 5.30. In blue, from the top to the bottom data acquired by microphones 1, 4, 11, 18, 35, 42, 49 and 52.

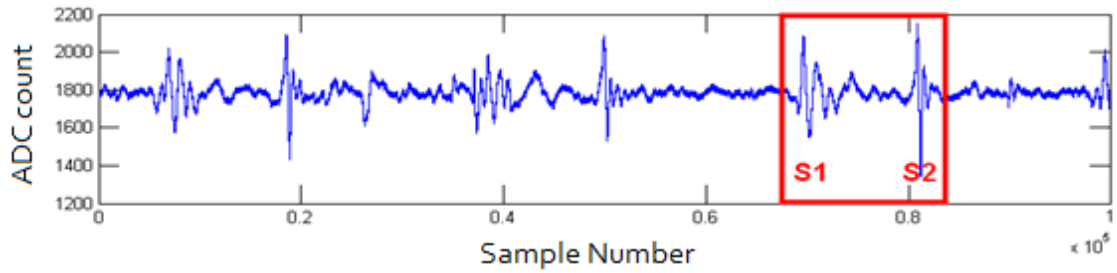

Figure 5.31. In blue, microphone number one in detail, 100.000 samples are displayed here and it is possible to observe three heart beats. In the red rectangle is possible to observe a single heart beat.

The ICA/AIIA algorithm processed a total of 6.4 million samples and the results are shown in Figure 5.32.

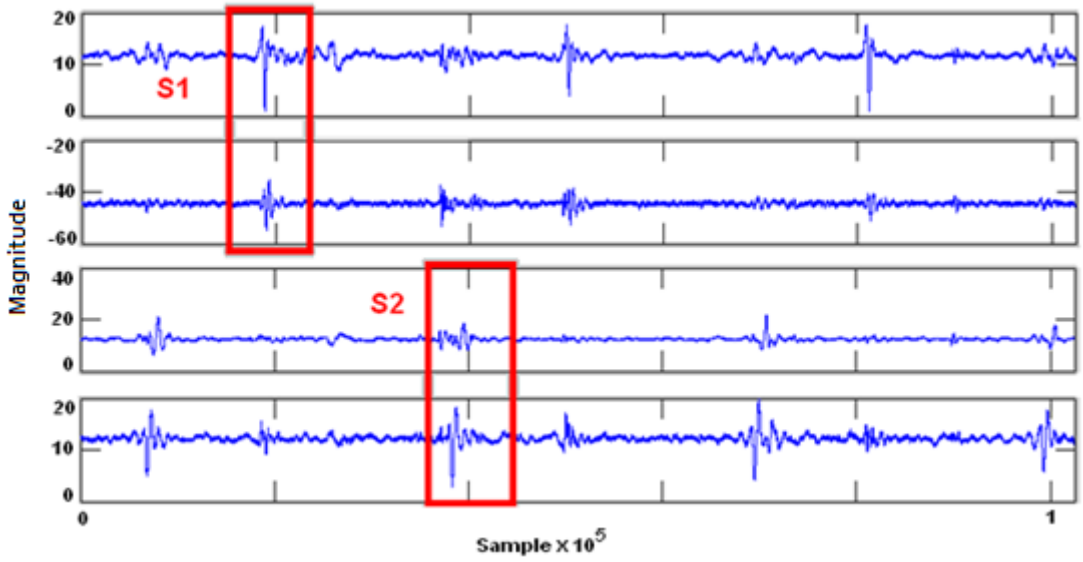

Figure 5.32. In blue, the four signals identified by the ICA /AIIA algorithm. It is possible identify the two fundamental heart tones in the red rectangles marked as S1 and S2 present on the extracted signals.

From the figure it can be clearly seen that the algorithm was able to find the sounds S1 and S2 and split them into four waveforms. It seems that the algorithm was actually able to split S1 into its Mitral (M) and Tricuspid (T) components and S2 into its Aortic (A) and Pulmonary (P) components. That is possible since these valves are in different locations, however is not completely clear that this is the case and further studies would be necessary for confirmation.

It is important to observe that the array was not modified in any way to perform this test and therefore was not optimized for this application. An increase in the gain of the second stage of the amplifying circuit would be beneficial since these signals are on

mean using only one eight of the ADC dynamic range. Also, a change in the corner frequency of the high pass filter to a frequency around 50 Hz would be desirable.

5.3.5 Source localization. Source localization in this work is used to provide the direction that sound sources are located in the acoustic horizon of the array when the sources are farther than ten centimeters, and to pin point sources when they are closer than ten centimeters.

Two different techniques can be used to perform source localization with this system; when the source is farther than ten centimeters, beamforming techniques using phase shifts is the preferred. When the source is located closer than ten centimeters, the sound pressure magnitude information was the choice.

The beamforming performed to give the direction of the incoming acoustic pressure wave is based on time delay and sum operations. In order to test the phase discerning and beamforming capabilities of the system an experimental setup was assembled with a source emitting a 0.5 seconds chirp waveform ranging from 2 kHz to 5 kHz. The source was placed at five different localizations as illustrated on Figure 5.33.

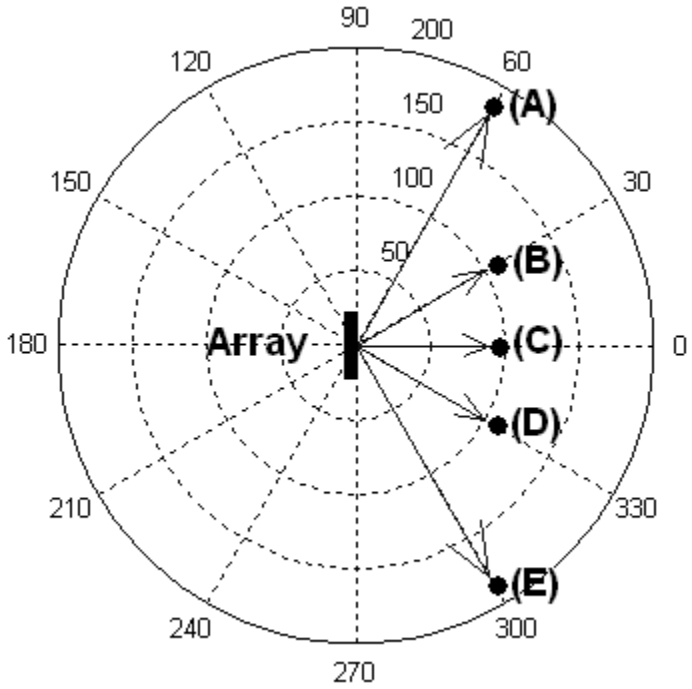

Figure 5.33. Test setup. Position relative to the array where the chirp source was introduced (A) located at 60 degrees left and 180 cm distant, (B) located at 30 degrees left and 100 cm distant, (C) located straight ahead and 90 cm distant,(D) located at 30 degrees right and 100 cm distant and (E) located at 60 degrees right and 180 cm distant.

The array was initially set to acquire 200 samples of data and it was triggered by the acoustic source initially set at position (A). The array captured data with all its sensors activated. Figure 5.34 illustrates the signals captured by two microphones on opposite sides of the array, the phase difference between the signals can be clear seen, it was measured in six samples with the ADC digitizing with a sample frequency of 36 Ksps. This is the basic principle on which phase based source localization is realized on this work.

The algorithm implements initially peak detection, where it detects where the first peak of the signals occurs and then fits a fifth order polynomial to it. After that, the fitting equation is derived and a more precise peak is found, the sample where this peak

occurred is then stored and this process is repeated for the whole acoustic array and stored in a matrix called the phase matrix.

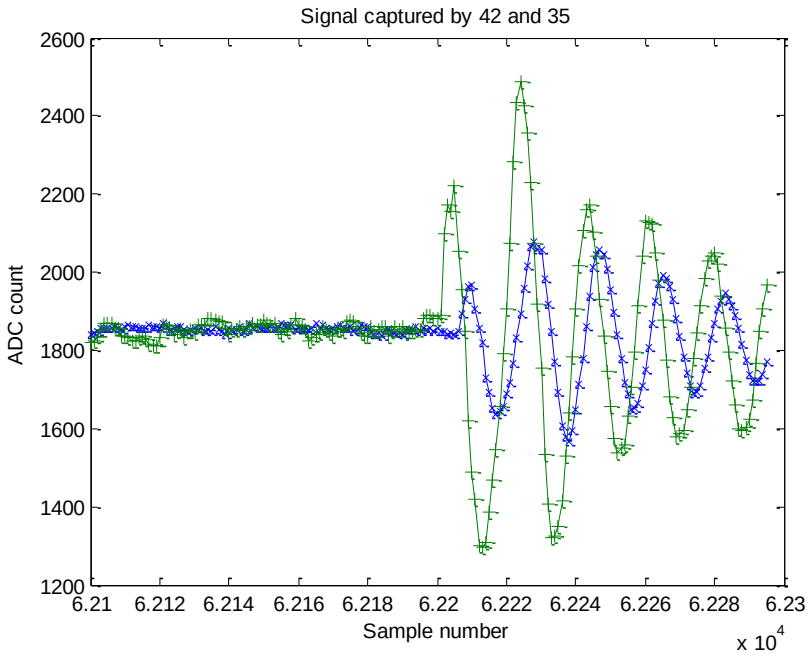

Figure 5.34. Phase difference from a signal captured by two different microphones on the sonic array originated from a source at 60 degrees from the array normal and 180 centimeters far. In green, microphone 42 and in blue microphone 35.

The phase matrix basically provides all the information that the system needs to performed beamforming, the matrix itself can provide source localization, Figure 5.35 displays the phase matrix. It's possible to observe on this picture that raw direction from where the sound originated can be obtained by histograming the rows and colunms of the matrix. A further improvement on source localization can be achivied by performing interpolation of the phase matrix and then calculating the center of mass of the system, this can be observed on Figure 5.36.

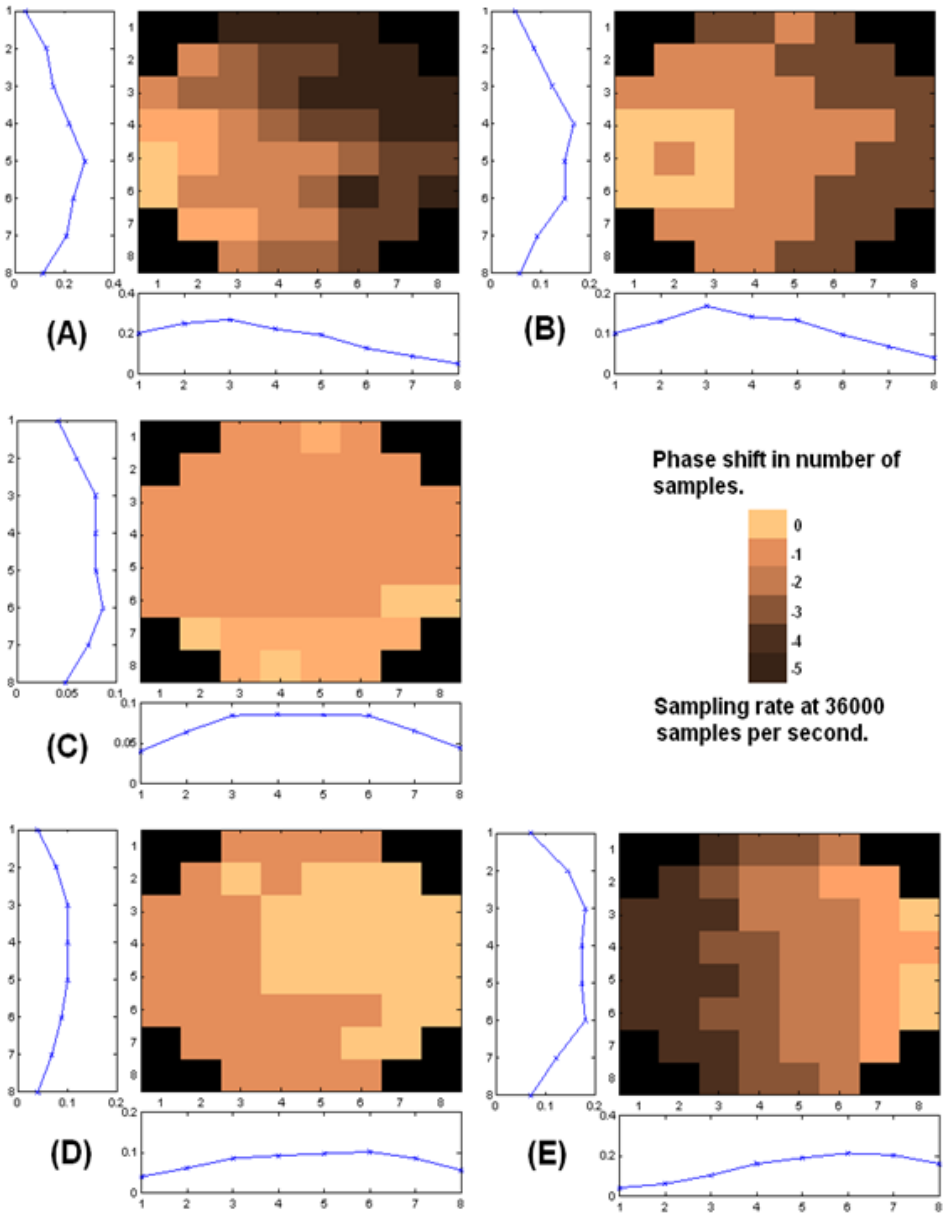

Figure 5.35. The phase delay array for the five positions that the test source was placed in, with its correspondent horizontal and vertical probability density function. (A) 60 degrees left,(B) 30 degrees left, (C) perpendicular, (D) 30 degrees right and (E) 60 degrees right.

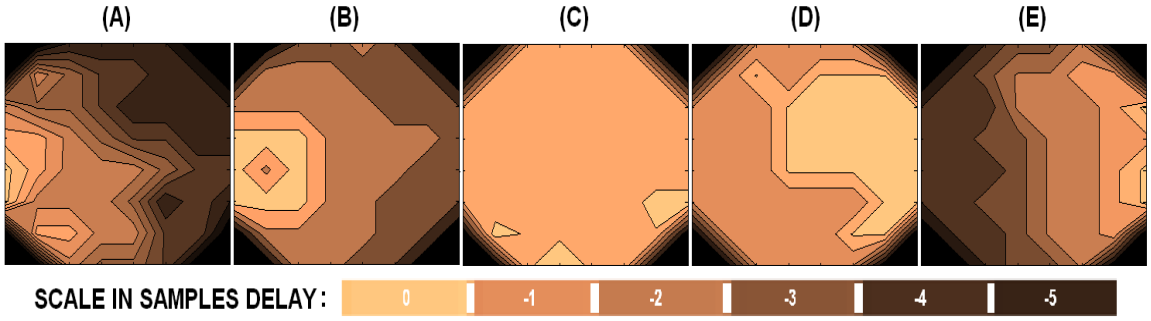

Figure 5.36. The interpolated phase matrix for the five positions wish the test source was placed in, (A) 60 degrees left,(B) 30 degrees left, (C) perpendicular, (D) 30 degrees right and (E) 60 degrees right.

With the phase matrix, the system adjusts all the channels to have the same phase and then sums and accumulate all the signals, this will result in a single signal that will be called the template vector. Figure 5.37 illustrates this concept with two microphones.

All the signals coming from the sonic array can now can be compared with the template vector making it possible to immediately realize if the source is moving and in what direction by comparing any of the sources with the template vector. This is the simplest form of beamforming that can be performed with the sonic array. The array can also have multiple templates, for example the columns can have one template and the rows another, in this way more precise vertical and horizontal tracking and localization is possible.

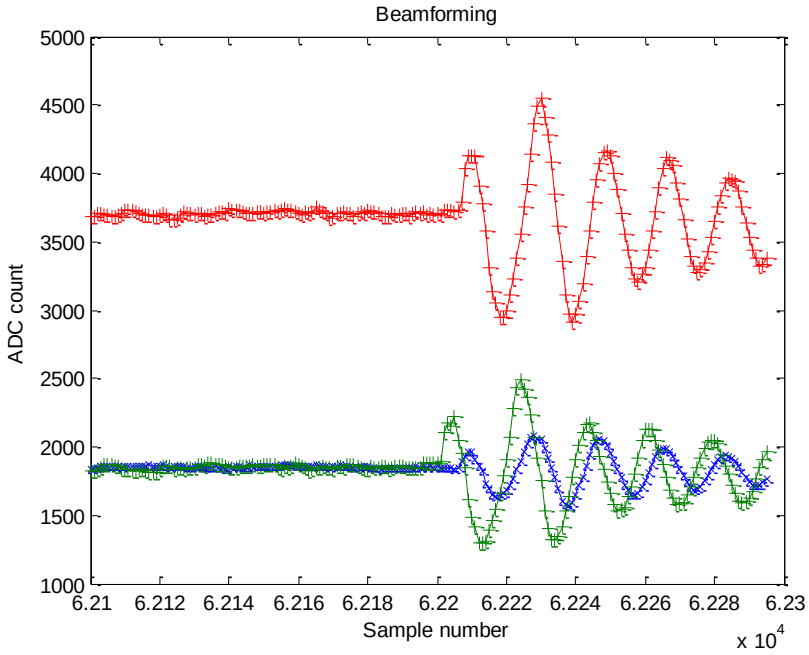

Figure 5.37. Beam forming with two microphones. In green microphone 42, in blue microphone 35 and on the top in red the template vector.

Source localization using the sound pressure magnitude was already illustrated on Figure 4.12. This kind of source localization requires the gain of all channels in the system to be equalized as explained on section 4.5, after gain equalization the source localization becomes trivial. Basically the signals are interpolated and the center of mass of the matrix is calculated thus indicating the position of the source or sources. Figure 5.38 shows the results obtained by using of this method, two sound sources were placed a distance of 2 centimeters from the acoustic array, the first one right in front of microphone number 16 while the other one in between microphones 44 and 37. It is pretty clear from the picture that the array is capable of locating both sources just by measuring the sound intensity difference.

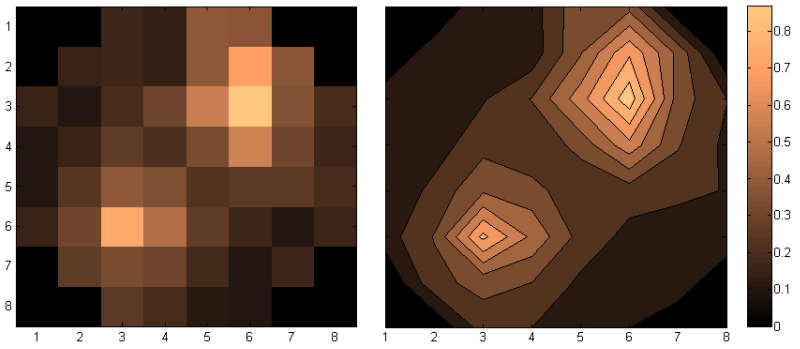

Figure 5.38. Two sources located, on the left the raw data and in the right the interpolated data pinpointing the sources.

Another example of sound intensity based source localization is the collecting of information for the heart experiment depicted in Figure 5.39. Here the heart is imaged using the sound intensity measured at each microphone, a posterior interpolation of the data was done and is illustrated on the same picture. It is possible to observe two very distinct patterns for s1 and s2 on the image.

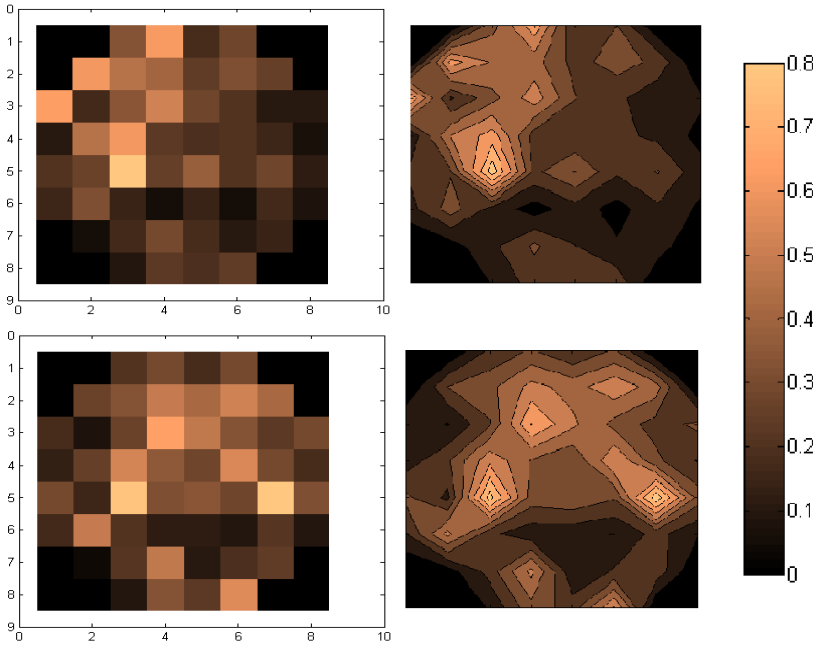

Figure 5.39. Imaging of the heart, on the top the sound image captured when the heart beat was at the s1 stage and on the bottom when it was at s2 stage.

CHAPTER 6

CONCLUSION

This work describes the design, development, and capabilities of a scalable MEMS microphone-array-system that can be used to enhance sound processing and acquisition for several types of applications. The system uses MEMS microphones arrays associated with the CAPTAN scalable architecture to deliver a powerful real-time signal processing, and acquisition platform. The system's capabilities enable versatile configurations for important applications such as sound source localization, sound separation, and sound imaging.

This research demonstrates that it is possible to integrate key technologies such as MEMS, high performance FPGA, and Gigabit Ethernet to produce a very compact network based acoustic array with high performance.

The data acquired by the array was provided and utilized for the application which this acoustic array was intented for: source separation and localization. The results show that the array integrated with the ICA/AIIA algorithm is capable to performing the task for which it was designed.

It is also part of this work the description and development of the CAPTAN architecture, its characteristics, capabilities, and limitations. Furthermore its application using the specially designed AMA sonic array is evaluated. The AMA array was integrated with the ICA, AIIA and sound localization algorithms and its capabilities were demonstrated in detail in conjunction with the algorithms mentioned.

Key performance parameters and system concepts of the CAPTAN hardware were measured and explained, the most important being the internal bus speed, system power consumption, thermo limitations, network operation, and speed.

Several algorithms and tests were presented as a proof of concept of this system. The system's performance using the specially designed AMA array was presented, where its performance was evaluated both electrically and acoustically. These tests provided a measurement of how successfully the CAPTAN system integrates with the transducer array subsystem. To complete the full system an algorithm capable of separating multiple sound sources was implemented, and its performance evaluated. A second algorithm design to provide sound localization was also developed, and demonstrated.

The algorithms's performance was judged based on its accuracy to locate and separate sources, number of sources and direction of sources.

Different versions of the CAPTAN system proposed in this work are already being deployed by universities and research laboratories for various applications.

The specific case of the system working with the MEMS array can bring benefits for many areas of acoustic signal processing, more specifically to multiple source separation and localization. The AMA array can be used in many different applications such as digital cardiac auscultation as described and demonstrated in this work.

BIBLIOGRAPHY

[Adc96] Adcock, J. E., Gotoh, Y., Mashao, D. J., & Silverman, H. F. (1996). Microphone-array speech recognition via incremental map training. In *The 1996 IEEE international conference on acoustics, speech, and signal processing: conference proceedings : May 7-10, 1996, Marriott Marquis Hotel, Atlanta, Georgia, USA* (Vol. 2, pp. 897-900). Piscataway, NJ: Institute of Electrical and Electronics Engineers.

[All00] Allen, G. E., & Evans, B. L. (2000). Real-time sonar beamforming on workstations using process networks and Poxix threads. *IEEE Transactions on Signal Processing, 48*(3), 921-926.

[Ben08] Benesty, J., Chen, J., & Huang, Y. (2008). *Microphone array signal processing*. Berlin: Springer.

[Ber41] Berry, A. C. (1941). The accuracy of Gaussian approximation to the sum of independent variables. *Transactions of the American Mathematical Society, 49*(1), 122-136.

[Bic05] Bickley, L. S. (2005). *Bates' guide to physical examination and history taking.* (9th ed.). [S.l.]: Lippincott Williams & Wil.

[Buc05] Buck, M., Haulick, T., & Pfleiderer, H. (2006). Self-calibrating microphone arrays for speech signal acquisition: A systematic approach. *Applied Speech and Audio Processing, 86*(6), 1230-1238.

[Cam99] Campbell, D. K. (1999). *Adaptive beamforming using a microphone array for hands-free telephony* (Unpublished master's thesis). Virginia Polytechnic Institute and State University.

[Com94] Comon P. (1994). Independent component analysis, a new concept? Elsevier, *Signal Processing*; *36: 287–314.*

[Cui06] *CUI inc. CDMG13008L-02 description: micro dynamic speaker*. (2006, September). Retrieved from www.cui.com/getPDF.aspx?filename=CDMG13008L-02.pdf

[Gir00] Girerd, C., Gardien, S., Burch, J., Katsanevas, S., & Marteau, J. (2001). Ethernet network-based DAQ and smart sensors for the OPERA long-baseline neutrino experiment. In *2000 IEEE nuclear science symposium conference record: October 15-20, 2000, Lyon, France.* (Vol. 2, pp. 12/111-12/115). Piscataway, NJ: IEEE.

[Hel01] Hélal, D., Crochet,, M., Luce, H., & Spano, E. (2001). Radar imaging and high-resolution array processing applied to a classical VHF-ST profiler. *Journal of Atmospherice and Solar-Terrestrial Physics, 63*(2-3), 263-274.

[Hod80] Hodgkiss, W. (1980). Dynamic beamforming of arandom acoustic array. In *1980 IEEE international conference on acoustics, speech and signal processing. proceedings, April 9-11, 1980, Denver* (Vol. 5, pp. 311-314). New York: IEEE.

[Hyv99] Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks, 10*(2), 626-634.

[Hyv00] Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks, 13*(4-5), 411-430.

[Iee99] IEEE P802.3ab 1000BASE-T task force. (1999, June). *Redirection*. Retrieved from http://grouper.ieee.org/groups/802/3/ab/index.html

[Iee93] IEEE P996.1 draft D1.00. (1993, October). *PC/104 Consortium*. Retrieved from http://www.pc104.org/pc104_specs.php

[Jer79] Jerri, A. J. (1979). The Shannon sampling Theorem—Its various extensions and applications. *Proceedings of the IEEE, 67*(4), 695.

[Jol02] Jolliffe, I. T. (2002). *Principal component analysis*. New York: Springer-Verlag.

[Kn06] *Knowles Acoustics surface mount MEMS microphones*. (2006). Retrieved from http://www.knowles.com/search/product.htm?x_sub_cat_id=3

[Ku05] Kurose, J. F., & Ross, K. W. (2004). *Computer networking: a top-down approach featuring the internet* (3rd ed., pp. 331-341). Boston: Addison-Wesley.

[Men07] Menasce, D., Turqueti, M., & Uplegger, L. (2007). The Renaissance: A test-stand for the forward CMS pixel tracker assembly. *Nuclear Instruments and Methods in Physics Research Section A, 579*(3), 1141-1149.

[Nag07] Nagasaka, Y., & Motoyama, H. (2007). A shared dataspace communication framework for data acquisition system. *15th IEEE-NPSS Real-Time Conference: [Batavia, Il., April 29 - May 4, 2007]* (pp. 1-4).

[Nak02] Nakadai, K., Okuno, H. G., & Kitano, H. (2002). Real-time sound source localization and separation for robot audition,. *IEEE international conference on spoken language processing* (pp. 193-196).

[Nat10] *National semiconductor ADC121S101 datasheet*. (2010, January). Retrieved from http://www.national.com/ds/DC/ADC121S101.pdf

[Nor06] Nordholm, S., & Siow, Y. L. (2006). Speech signal extraction utilizing PCA-ICA algorithm with a Non-Uniform Spacing Microphone Array. In *2006 IEEE International Conference on Acoustics, Speech, and Signal Processing: proceedings : May 14-19, 2006, Centre de Congrès Pierre Baudis, Toulouse, France* (Vol. 5, p. V). [Piscataway, NJ]: IEEE Signal Processing Society.

[Oru04] Oruklu, E., & Saniie, J. (2005). Ultrasonic flaw detection using discrete wavelet transform for NDE Aapplications. In *Proceedings of the 2004 IEEE International Frequency Control Symposium Exhibition 23-27 August, 2004, Palais des Congrès, Montréal, Canada* (Vol. 2, pp. 1054-1057). Piscataway, N.J.: IEEE.

[Par03] Park, J., & Mackay, S. (2003). *Practical data acquisition for instrumentation and control systems*. Oxford: Elsevier.

[Ric95] Rice, J. A. (1995). *Mathematical statistics and data analysis* (2nd ed.). Belmont, CA: Duxbury Press.

[Riv08] Rivera, R., Turqueti, M., & Prosser, A. (2008). A software solution for the control, acquisition, and storage of CAPTAN network topologies. In *2008 IEEE Nuclear science symposium conference record: October 2008, Dresden, Germany* (pp. 805-808). Piscataway, N.J.: IEEE.

[San81] Saniie, J. (1981). *Ultrasonic signal processing: system identification and parameter estimation of reverberant and inhomogeneous targets* (Unpublished Ph.D. thesis). Purdue University.

[Sto04] Stone, J. V. (2004). *Independent component analysis: a tutorial introduction* (Vol. 1). Cambridge, Mass.: MIT Press.

[Tur08] Turqueti, M., Rivera, R., Prosser, A., Andresen, J., & Chramowicz, J. (2008). CAPTAN: A hardware architecture for integrated data acquisition, control and analysis for detector development. *2008 IEEE Nuclear science symposium conference record: October 2008, Dresden, Germany* (pp. 3546-3552).

[Tur10A] Turqueti, M., Saniie, J., & Oruklu, E. (n.d.). Acoustic imaging using the CAPTAN architecture. In *I²MTC 2010 IEEE international instrumentation and measurement technology conference : May 3-6, 2010, Austin, Texas, USA*.

[Tur10B] Turqueti, M., Saniie, J., & Oruklu, E. (n.d.). Real-time independent component analysis implementation and applications. *IEEE real time conference: May 24-28, 2010, Lisbon, Portugal*.

[Tur10C] Turqueti, M., Saniie, J., & Oruklu, E. (n.d.). Scalable acoustic imaging platform using MEMS array. *2010 IEEE Iinternational conference on electro/information techonology: May 20-22, 2010, Normal, Illinois, USA*.

[Upl06] Uplegger, L., Appel, J. A., Artuso, M., Cardoso, G., Cease, H. P., Chiodini, G., ... Wang, J. C. (2006). First look at the beam test results of the FPIX2 readout chip for BTeV silicon pixel detector. *IEEE Transactions on Nuclear Science*, *53*(1), 409-413.

[Wei04] Weinstein, E., Steele, K., Agarwal, A., & Glass, J. (2004). *A 1020-node modular microphone array and beamformer for intelligent computing spaces* (Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory). Cambridge, MA: MIT.

[Xil09] Xilinx Inc. Virtex-4 FPGA configuration user guide, UG071. (2009, June 9). *FPGA and CPLD Solutions from Xilinx, Inc.* Retrieved from http://www.xilinx.com/support/documentation/virtex-4_user_guides.htm

[Xil08] Xilinx Inc. Virtex-4 FPGA user guides, UG070. (2008, December 1). *FPGA and CPLD Solutions from Xilinx, Inc.* Retrieved from http://www.xilinx.com/support/documentation/virtex-4_user_guides.htm