A BINAURAL SOUND SOURCES LOCALISATION APPLICATION FOR SMART PHONES

By

PIUS KAVUMA BASAJJABAKA, MUGAGGA

BSc (Elec. & Computer Eng.)

A dissertation submitted to the Department of Electrical Engineering,

University of Cape Town, in fulfilment of the requirements

for the degree of **Master of Science in Engineering**

At the

UNIVERSITY OF CAPE TOWN

Supervisor:

Dr. Simon Winberg

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author: .....

Cape Town

25 March 2015

Dedication

I dedicate this project to all people that dare to dream.

Abstract

The ability to estimate positions of sound sources is one that gives animals a 360° awareness of their acoustic environment. This helps compliment the visual scene which is restricted to 180° in humans. Unfortunately, deaf people are left out on this ability.

Smart phones are rapidly becoming a common tool amongst mobile users in developed and emerging markets. Their processing ability has more than doubled since their introduction to mass consumer markets by Apple in 2007. Top-end smart phones such as the Samsung Galaxy Series; 3, 4, and 5 models, have two microphones with which one can acquire stereo recordings.

The purpose of this research project was to establish a feasible Sound source localization algorithm for current top-end smart phones, and to recommend hardware improvements for future smart phones, to pave way for the use of smart phones as advanced auditory sensory devices capable of acting as avatars for intelligent remote systems to learn about different acoustic scenes with help of human users.

The GCC-PHAT algorithm was chosen as the underlying core DOA algorithm due to its suitability for pair-wise localization as highlighted in literature. A stochastic power accumulation algorithm was designed and implemented to improve estimation outcomes by GCC-PHAT. This algorithm was based on inspiration from W-disjoint orthogonality assumption in literature and was extended to perform sound source counting and time domain source separation.

The system yielded satisfactory azimuth estimates of sound source directions in real time with pin-point DOA estimation accuracy rates of 64%, and 90.67% accuracy rate when a tolerance of $\pm$ 1 correlation sample is considered. An effort to resolve front back ambiguity using phone orientation data from the MEMs sensors yielded un-satisfactory results prompting a recommendation that an

extra microphone would be needed to achieve 360° localization in a more user friendly way.

The dissertation concludes with plans for further work on the topic and provision of a further refined API and optimised libraries to facilitate development of customised solutions using this system.

Acknowledgements

I express my gratitude to each and every body that interacted with me in one way or the other during these past couple of years. All interactions impacted positively towards my learning process.

Special thanks to God, my supervisor Dr. Winberg, all his colleagues, and my family and friends for all the help and moral support they provided.

Nomenclature

Term	Description
A.I	Artificial Intelligence
ADC	Analogue to Digital Conversion
AOA	Angle of arrival
API	Application Programming Interface
CAS	Central Auditory System
DOA	Direction of Arrival
DOF	Degrees Of Freedom
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
GCC	Generalized Cross Correlation
HRTF	Head Related Transfer Function
iFFT	inverse Fast Fourier Transform
ILD	Inter-aural Level difference
ITD	Inter-aural Time Difference
MPS	Music, Pulse, Speech
PCM	Pulse Coded Modulation
PHAT	Phase Transform
PSM	Pulse, Speech, Music
SCOT	Smoothed Coherence Transform

SDK	Software Development Kit
SPM	Speech, Pulse, Music
SSL	Sound Source Localization
TDOA	Time Difference Of Arrival

Contents

List of Figures

List of Tables

1 Introduction

This work focuses on developing a suitable sound source localisation algorithm for smartphones and implementing a prototype application using the algorithm. This is meant to demonstrate the ability to use smartphones as auditory sensing devices that can be used to develop applications aimed at helping deaf people visualise acoustic scenes around them, in addition to providing input to sound scene analysis systems.

1.1 Background

Sound source localisation (SSL) is the process of estimating or identifying the position of an acoustic source with reference to a point of interest, such as a listener's head [1]. In humans, this task is carried out by the Central Auditory System(CAS) and appears to be done seamlessly although the underlying process is considerably complex [1] (front-back ambiguity resolution, sound level normalisation[2], ranging).

According to perception studies, the CAS has been found to use various auditory cues for SSL [3][2] of which the most adopted by biomimicry are, inter-aural time difference (ITD) and inter-aural level difference (ILD), collectively known as binaural cues. These cues are used to estimate the position of an acoustic source based on the time delay and signal level difference between a pair of acoustic receivers. Biomimicry according the online Oxford dictionary is "the design and production of materials, structures, and systems that are modelled on biological entities and processes". Another cue used by mammals but rather more technically complex and less mimicked in electronics is the Head Related Transfer Function (HRTF). HRTF takes into account spectral scattering due to the anatomy of the person in order to localise sound sources [4][5].

Sound source localisation is a very valuable capability for animals (including humans) in helping them learn about and interact with their respective environments. For example, humans use auditory sensing to acquire a 360°

awareness of the acoustic scene that compliment the visual scene which is usually restricted to 180° [2]. This ability helps humans to be aware of acoustic sources that are out of range of the visual scene. For animals in general, this ability is essential in providing early warning against predators or any other adverse/hazardous event that produces acoustic signals, such as falling objects or approaching vehicles. Hence, deaf people are lacking in this amazing ability.

1.1.1 SSL for Hearing impaired people

Hearing impaired people include deaf and hard-of-hearing people. Deaf refers to a level of impairment in which there is complete hearing loss whereas hard-of-hearing refers to partial hearing loss[6] that may be compensated by use of hearing aids or cochlea implants[7]. In cases where hearing loss is not compensated for, the people affected may be at elevated safety risks considering that various safety alarms such as fire alarms, burglar alarms, and car horns use sound signals.

It would therefore be ideal to have a system that can convert sound cues into visual cues to visually alert deaf people of the sound activity in their environment. Such a system would be required to provide as a minimum, the direction of sound sources and relative intensity level indication of the detected sounds.

1.1.2 SSL as a key component for Artificial Intelligence (A.I) systems

Artificial intelligence is a field of science that aims at achieving learning, perception, and cognition abilities in computerised systems such as robots[8]. According to the Oxford online dictionary, perception is defined as "the ability to see, hear, or become aware of something through senses." In the same dictionary, cognition is defined as "the mental action or process of acquiring knowledge and understanding through thought, experience, and senses."

A human being uses five senses to acquire data from their environment, of which, auditory, visual, and smell senses are largely used to trigger vital memories [9].

Hearing helps people learn how to communicate with one another through oral language. Our ability to comprehend speech is one that truly sets us apart intellectually from the rest of the animal kingdom[10].

Visual cues can be used to solve ambiguities that arise amongst sounds that are acoustically confusable as for the case of phoneme pairs /m/ and /n/ [11]. SSL would therefore be vital in guiding the visual sensors of an A.I system towards the direction of an active sound source in order to use both audio and visual modalities in the process of perception, learning, and cognition.

1.1.3 Smart phones as an engineering tool

Smartphones are increasingly becoming a common tool amongst people that can be utilised for various purposes other than making and receiving calls. Several smart phone handsets nowadays come with two microphones, adequate processing capabilities[12], and a barrage of sensors. Coupled with regularly updated Software Development Kits (SDK) such as the Android SDK, smart phones have the potential to be used for a variety of applications. The SDK provides a useful abstraction between the developer and the underlying hardware. This abstraction can help improve on a developer/engineer's productivity by eliminating the need for them to handle the hardware level control of the underlying sensors. As a result, a smart phone can be used with relatively less complexity, as a sensing tool with substantial processing power.

1.2 Objectives

The main objective of this project is to design, implement, test, and analyse a prototype of an algorithm to perform sound source localisation on a smart phone. The outputs should be source directions and intensity estimates displayed in a user-friendly manner on the screen.

The main objective of this project is separated into the following sub-objectives:

Table 1.2-1 Objectives

#	Sub-objective
01	Design and implement a sound source localisation algorithm prototype on a smart phone
02	Analyse the performance of the implemented system.

- **01:** Designing and implementing a sound source localisation algorithm prototype on a smartphone will provide a proof of concept showing that smartphones can be used as auditory sensing devices for applications aimed at using smartphones to provide auditory cues to deaf people in the form of visual cues.
- **02:** Analysing the performance of the implemented prototype will help us refine the algorithm through parameter selection, further testing, and analysis to establish working limits.

The outcomes from these objectives can be used to establish recommendations for future smart phone SSL implementations. This would help ensure continuity of this work to develop better systems that may overcome current limitations and add more functionality. Fig 1.2-1 is an illustration of how the intended current work is meant to fuse with future work and 3rd party systems that may run on separate machines other than the smartphones.

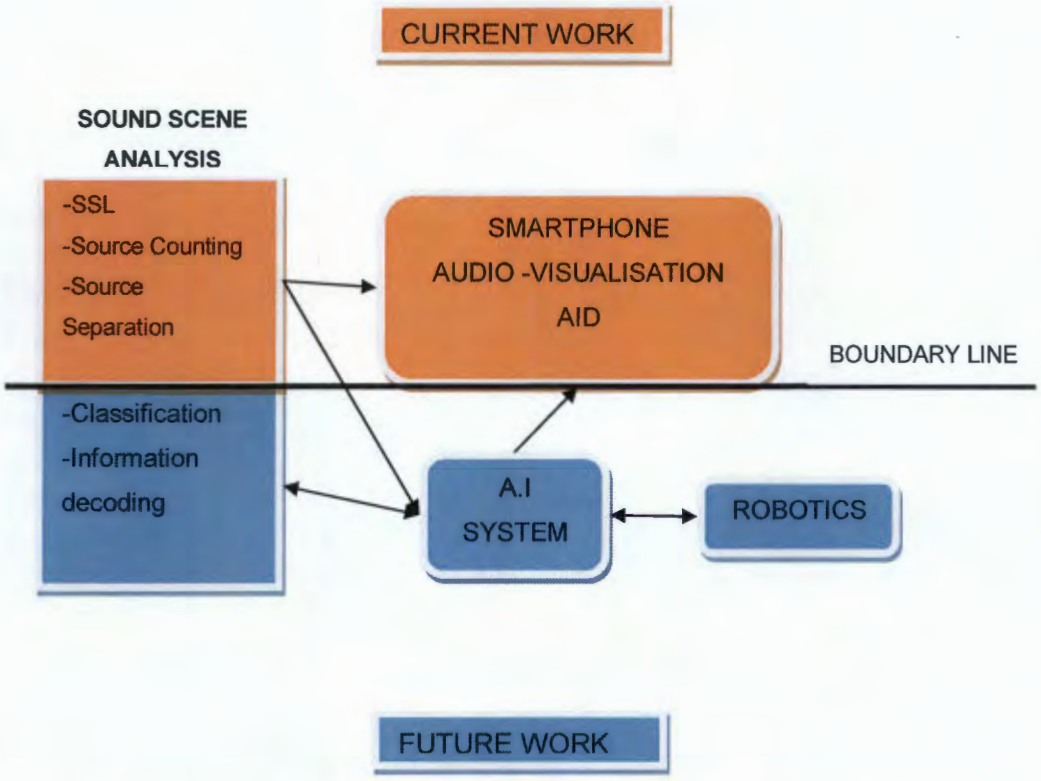


Figure 1.2-1 Illustration of current work being done on smartphone and how it can be combined with future work and 3rd party systems

The boundary line in fig 1.2-1 is a distinction between the intended current scope of work, and future work. The diagrams above the boundary line indicate what is intended to be covered by the scope. The diagrams below the boundary line are an indication of the proposed future work aimed at transforming the current work into a full sound scene analysis system that would be able to provide much more functionality to the smartphone audio-visualisation application for deaf people. The arrows indicate the flow of information between modules. The blocks filled in orange are implemented on the smart phone whilst those in blue may be implemented by 3rd party remote systems that bear more processing and storage power.

1.3 Research questions

In order to achieve the mentioned objectives, the following research questions need to be assessed:

1. What would be the most suitable algorithm for performing SSL on a smart phone, given that the system is constrained by; processing ability, number of microphones available, and the spacing of the microphones?
2. Is it possible to use orientation data to overcome localisation ambiguity caused due to microphone constraints?
3. What clustering value for parameter dT yields the best results?
4. What would be the accuracy of the sound source localisation algorithm?
5. What would be the time response of the proposed system?
6. Can source counting and separation be done within the same framework of the proposed algorithm?

1.4 Scope and Limitations

The scope of the project is to design, implement, test, and analyse a prototype of an algorithm to perform sound source localisation on a smart phone. Recommendations for future work are to be made based on outcomes from analysis. Due to hardware and software implications of smartphones, the scope of this project is limited as follows:

- The project is to be prototyped on an android smart phone. Reasons being Android operating system is an open source platform and currently has largest market share of the global smartphone market[13].
- To the best of my knowledge, at the time of development of this project, there existed no smartphone with more than two programmer-accessible microphones and an API to support simultaneous recordings from more than two microphones.
- As a result, the Algorithm is meant to acquire and process signals from a single pair of microphones rather than arrays of greater than two microphones.

- This project is not intended to cover sound scene analysis but rather provide a basis on which sound scene analysis can be built onto.

1.5 Document Outline

Chapter 2 begins by analysing sound wave generation and propagation. Sound source localisation by humans is then discussed. Key attributes of sound source localisation together with the methods used to achieve them, are pointed out and discussed in detail in individual sub sections of the chapter. The Chapter ends with a conclusion summarising the pros and cons of the various approaches to sound source localisation techniques in context of the objectives of this dissertation.

Chapter 3 describes the methodology to be followed during this research. The chapter starts by defending the choice to use a combined methodology.

- A flow diagram representing the methodology is presented then discussed in detail in subsequent sub-sections.

- A high level design of the proposed system is then presented and discussed based on how literature is used within the design.

- The chapter sums up with a list of experiments to be covered in order to attain answers to the research questions listed in section 1.3

Chapter 4 delves into the design process by first presenting a high level abstraction of the proposed designed. The subsequent sections illustrate detailed designs of the underlying modules while using results from Chapter 5 to refine the designs.

Chapter 5 starts by describing the experiment set up and the test procedure to be used. The results from this chapter are used as feed back in the spiral design model encompassed by Chapter 4.

Chapter 6 Draws conclusion from results attained with reference to the objectives and research questions stated in chapter 1. The chapter ends off with a hardware and software recommendation to enable a more robust algorithm to be implemented in future.

2 Literature Review

This chapter begins by looking at the principles of wave propagation from a point source. After this, the main cues used for binaural sound source localisation are described, namely; Inter-aural Time Difference (ITD), Inter-aural Level Difference (ILD), and Head Related Transfer Functions (HRTFs). Binaural localisation cues are cues inspired by the human auditory sensory system that relies on two ears to localise sound in space.

A brief discussion is given on beamforming, which is an alternative method for sound source localisation and source separation. The downside to beamforming however, is the requirement to use beyond 2 microphones and an underlying heavy demand on computing power.

We pre-conclude the chapter by discussing cones of confusion that arise when ITD cues are used and similarly spheres of confusion caused when ILDs are used.

We end off the chapter by looking at additional methods of sound source counting and separation in context to our scope of limitations.

2.1 The anatomy of sound

Sound travels through air by compression and rarefaction of air molecules, propagating as a spherical wave away from a point source as indicated in fig 2.1-1 like ripples in a pond.

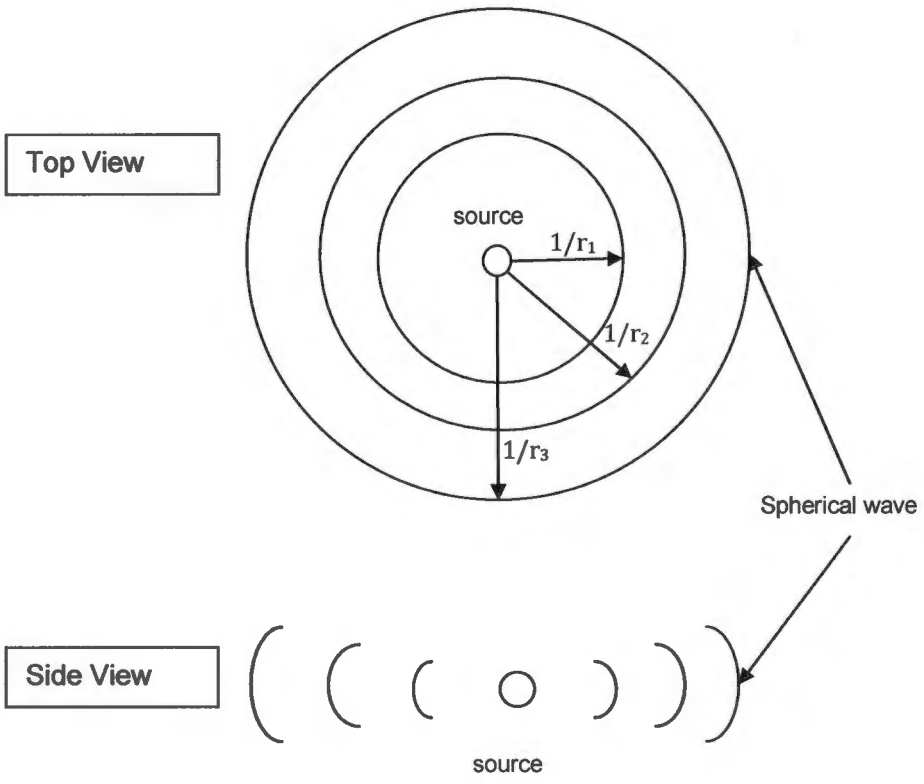


Figure 2.1-1Spherical wave propagation from a point source

r represents the distance from the point source to the n^{th} wave. The amplitude of each spherical wave of radius **r**, approximately decays by a constant factor of 1/r [14] due to spherical spreading loss.

As the radius increases, the curvature of the spherical wave more closely approximates a plane wave in relation to an arbitrary microphone array being

utilised in far field conditions. This allows for the approximation of sound waves as planer waves in order to simplify design.

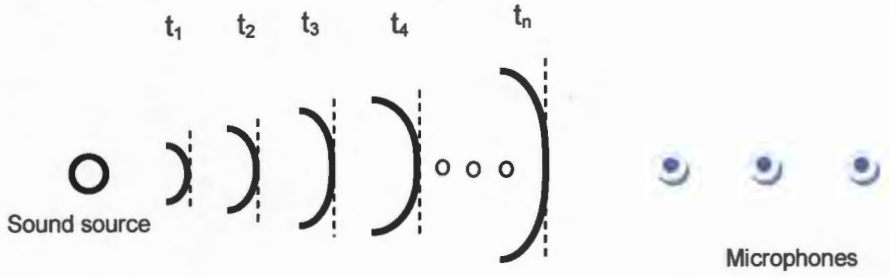


Figure 2.1-2 Illustration of decrease in curvature as a sound wave propagates from a point source

The figure above illustrates how the curvature of spherical wave decreases as the wave propagates outwards. t_1 to t_n represent time stamps theoretically associated to each wave.

Section 2.2 builds upon the assumption of planer waves into the inter-aural time difference theory.

2.2 Inter-aural Time Difference

Inter-aural time difference of arrival methods utilise differences in arrival time of a plane wave at two spatially separated sensors as illustrated in fig 2.2-1. Each plane wave generates instantaneous sound samples at each microphone as it propagates past them. A time-series collection of these samples collected per microphone is referred to as a sound frame. The length of the sound frame is the number of samples constituting the frame.

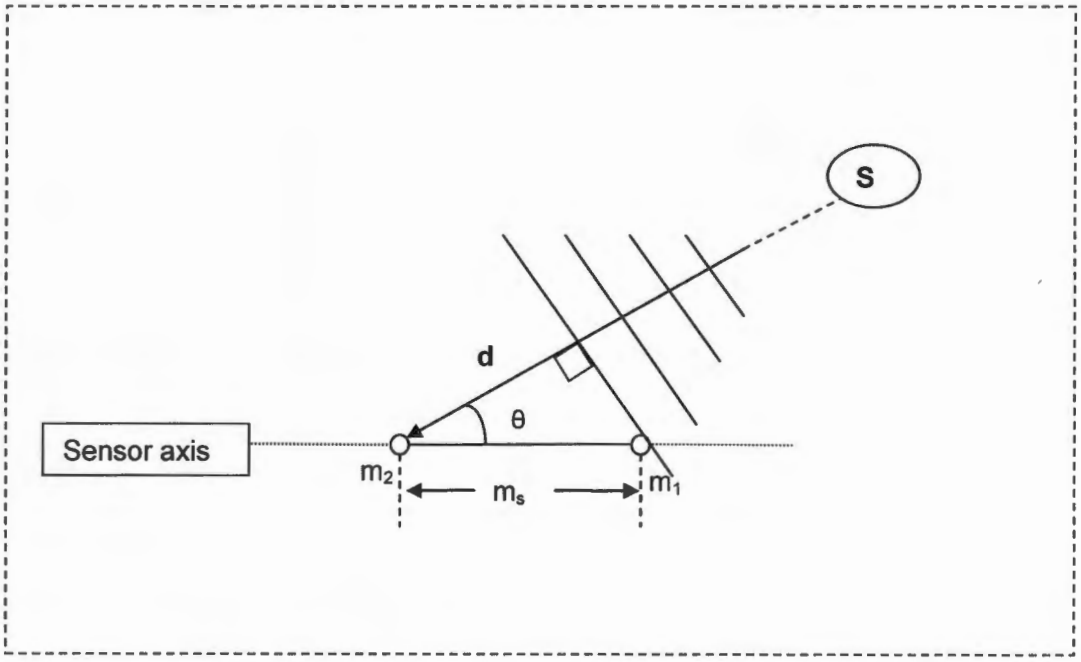


Figure 2.2-1 Plane wave propagation from source towards sensor axis

The signals received by microphone m_1 and m_2 are referred to as $x_1(t)$ and $x_2(t)$ respectively in this section, and mathematically modelled as:

$$x_1(t) = s_1(t) + n_1(t) \quad (1)$$

$$x_2(t) = \alpha s_1(t + D) + n_2(t) \quad (2)$$

Where $s_1(t)$, n1(t) and n2(t) are real, jointly stationary random processes[15]. $n_1(t)$ and $n_2(t)$ are noise signals and are assumed to be uncorrelated to $s_1(t)$, the signal emanating from the sound source **S**.

Since the sampling of sound in digital systems is a discrete process governed by sampling frequency, there are only a discrete number of samples by which one sensor lags or leads another. The length **d** in fig 2.2-1 represents the distance a plane wave has to travel to reach the second sensor after propagating past the first one. In the subsequent equations the length **d** is linked to the delay in terms of samples and an elaborate explanation linking the variables in fig 2.2-1 is made.

From basic right angled trigonometry, θ the angle incident to the direction of propagation of the plane wave is related to **d** and m_s (microphone spacing) by the cosine function as follows:

$$\cos\theta = d/m_s \tag{3}$$

$$d = c \times D$$

d is the distance the plane wave has to travel to sensor m_2 after reaching sensor m_1. Where c is the speed of sound approximated to be 345m/s [16] . D is the time (delay) it takes for the plane wave to travel from sensor m_1 to m_2. It is related to the delay in samples D_s and sampling frequency F_s by equation (4)

$$D = D_s * T_s = D_s * 1/F_s \tag{4}$$

Where T_s is the sampling Period and is equivalent to the propagation time represented by each sample.

$$\therefore d = c * D_s / F_s \tag{5}$$

Substituting equation (5) into equation (3) yields equation (6)

$$\cos\theta = c * D_s/(F_s * m_s) \quad (6)$$

$$\theta = \arccos(c * D_s/(F_s * m_s)) \quad (7)$$

Taking the arccos in equation (7) yields the angle of arrival; θ. A look-up table can be generated to establish the corresponding θ for a given set of sample-delays D_s, given that c, F_s, and m_s are held constant. Chapter 4 gives a detailed analysis of the look up table.

F_s and m_s are found to have a bounding effect on the maximum number of samples by which one sensor can lag the other. This limit is due to the arccos only being valid for the range where its argument $\varphi \in [-1,1]$, given that equation (7) is re-written as:

$$\theta = \arccos(\varphi)$$

$$\text{where } \varphi = c * D_s/(F_s * m_s) \quad (8)$$

Taking an ascending order of Ds from 0, the last value to causes $\varphi \leq 1$, is considered as the maximum number of samples one sensor lags the other.

2.2.1 Cross correlation

Cross correlation is a means of measuring how closely one waveform matches a subsection or the entirety of another as a function of a time-lag[15] which can be expressed as equation (9)

$$R_{x_1x_2}(\tau) = E[x_1(t)x_2(t-\tau)] \quad (9)$$

Where E represents Expectation and the argument τ that maximises the function, relates to the estimated time delay between the two microphones. Finite observation times of the signals are used to maintain the correlation model's

assumption of stationarity of the observed signal [15]. An estimate of the correlation is hence given as

$$R_{x_1x_2}(\tau) = \frac{1}{T_w - \tau} \int_{\tau}^{T_w} x_1(t)x_2(t-\tau)dt \qquad (10)$$

Where T_w symbolises the observation interval and is equal to the windowing length. For a range of time shifts τ, the signals are multiplied, integrated and squared until a peak is obtained[15].

In context of sound source localisation by 'time difference of arrival' methods, cross correlation can be used to determine the number of samples by which a relatively egordic-stationary sound source signal arriving at one microphone lags or leads the same signal arriving at another microphone. The delay can then be appropriately substituted into equation (7) to estimate the associated direction of arrival.

It can be noted, that the complexity of processing equation (10), is directly proportional to the observation interval T_w since a set of multiplications and integrations has to be done for every shift in the observation interval. Consideration has to be given to the choice of T_w length to ensure that stationarity is catered for. For example, stationarity of speech signals may be assumed within a couple of 10milliseconds [17]. A maximum assumption length of 30milliseconds for T_w [18] is commonly used. Furthermore, the length of T_w determines the FFT resolution; the longer the length, the higher the resolution. Hence design considerations need to optimise for a good trade off between stationarity assumptions and FFT resolution.

2.2.2 *Generalised Cross Correlation*

The generalised cross correlation is an efficient means of performing cross correlation in the frequency domain. It utilises the principle that a convolution in one domain translates to a multiplication in the other[19]. A cross correlation between $x_1(t)$ and $x_2(t)$ is a convolution in the time domain and hence transforms to a multiplication in the frequency domain. Each signal is independently converted to the frequency domain using an FFT algorithm. For discrete analysis, the Generalised Cross Correlation can be expressed as

$$R_{x_1x_2}[n] = F^{-1}\{X_1[k].X_2^*[k]\} \quad (11)$$

Where $X_i[k]$ is the FFT of the signal received at the i^{th} microphone and k are the frequencies from k = 0, 1....N-1. N is the FFT length. In practice, N is twice the observation interval T_w, given it's a power of 2, in order to avoid cyclic convolution.

In addition to its efficiency of performing cross correlation, it bears the advantage of being easily weighted by various filters for various performance enhancements[19]. Equation (11) can therefore be expressed as

$$R^{(g)}_{x_1x_2}[n] = F^{-1}\{\Psi_g[k](X_1[k].X_2^*[k])\} \quad (12)$$

Where $\Psi_g[k]$ represents the general frequency weighting operator. Knapp and Carter give a detailed discussion in [15] on the effect of various weighting operators on the performance of the estimator. In brief, Knapp and Carter highlight that broad peaks generally result when the weighting operator $\Psi_g[k] = 1$. They point out that in case of a single delay (one sound source), the peak, though broadened would still be a good estimate of the actual delay. However, in the presence of various delays, delta functions corresponding to the various delays are smeared by the signal spectrum and lead to indistinguishable peaks relating to the desired delay. Therefore, in order to improve on resolution, a

suitable weighting operator must be selected to sharpen the peak relating to the estimated delay. The downside to sharp peaks is that they are more prone to errors due to finite observation time especially in cases of low Signal to Noise ratio. As a result, the choice of $\Psi_g[k]$ leads to a trade off between good resolution and stability[15].

The most common weighting operators that have been proposed over time are:

ROTH

The ROTH processor was proposed by Roth while working at Hewlett-Packard Company [20].

$$\Psi_R[k] = \frac{1}{G_{x_1x_1}[k]} \tag{13}$$

It has the desirable ability to suppress frequency regions where $G_{n_1n_1}[k]$ is large [15]. However if

$$G_{n_1n_1}[k] \neq c\big(G_{s_1s_1}[k]\big),$$

Where c is a constant, the delta function will again be spread out [15].

SCOT

The Smoothed Coherence Transform was proposed by Carter as part of a solution to the similar problem Roth was trying to solve; "an attempt to determine time delays between weak broad-band correlated noises received at two sensors" [21].

While using the ROTH processor, one may be uncertain on whether to use

$$\Psi_R[k] = \frac{1}{G_{x_1x_1}[k]} \text{ or } \frac{1}{G_{x_2x_2}[k]} \tag{14}$$

Since errors in $G_{x_1x_2}[k]$ may be caused by either bands where $G_{n_1n_1}[k]$ or $G_{n_2n_2}[k]$ is large.

The SCOT processor hence selects

$$\Psi_S[k] = \frac{1}{\sqrt{G_{x_1x_1}[k]G_{x_2x_2}[k]}} \tag{15}$$

When $G_{x_1x_1}[k] = G_{x_2x_2}[k]$ the SCOT becomes similar to the ROTH processor, and if $n_1(t) \neq 0$, $and\ n_2(t) \neq 0$ spreading persists [15].

PHAT

The Phase Transform was purely developed as an ad-hoc technique which in cases of un-correlated noises does not suffer spreading like the ROTH and SCOT [15]. The PHAT processor is defined as

$$\Psi_P[k] = \frac{1}{|G_{x_1x_2}[k]|} \tag{16}$$

This weighting sets unity gain to all frequency bins while preserving phase information [19].

The Generalised Cross Correlation has commonly come to be used for TDOA [22] with the Phase Transform as the choice of weighting processor [23][24], due to its favourable performance in reverberation environments compared to its counterparts. GCC-PHAT is an ideal method for fast pair-wise TDOA estimations with low computational complexity.

2.3 Inter-aural Level Difference

The signal Si in fig 2.2-1 not only varies in time difference but also in level difference at both sensors. The sensor closest to the sound source will receive higher signal intensity than those further away. This binaural cue forms the basis for Inter-aural level difference sound source localisation mechanisms. In accordance with the square inverse law, equations (1) and (2) can be reformulated to model the signal received at the i^{th} microphone as

$$x_i(t) = \frac{s(t)}{d_i} + n_i(t) \tag{17}$$

Where d_i is the distance between the sound source and the ith microphone and $n_i(t)$ is additive white Gaussian noise [25]. As one can notice from equation (17), the time shift between signals is ignored in the formulation in order for emphasis to be on the ILD cues.

The energy received at the ith microphone can be calculated by integrating the square of the signal over the observation interval [0, Tw]. Hence it follows

$$E_i = \int_0^{T_w} (s^2(t)/d_i^2 + n_i^2(t))dt$$

$$E_i = \frac{1}{d_i^2}\int_0^{T_w} s^2(t)dt + \int_0^{T_w} n_i^2(t)dt \tag{18}$$

As stated earlier, if $n_i^2(t)$ is assumed to be uncorrelated noise with zero mean, its integration becomes zero and what is left of the equation is evidence of the inverse square law whereby the energy received by the microphones decreases as the inverse of the square of the distance to the sound source[25].

In the case of a two-microphone set up with assumed zero mean noise ($\int_0^{T_w} n_i^2(t)dt = 0$), a relationship between the energies received at both microphones can be derived from equation (18) as follows

$$E_1 d_1^2 = E_2 d_2^2 \tag{19}$$

Assuming a planer world by setting (x_i, y_i) as the coordinates of the ith microphone and (x, y) as the coordinates of the sound source; $d_i^2 = (x - x_i)^2 + (y - y_i)^2$ [25]. Substituting this into equation (19) yields the following system of equations

$$[x \; y \; 1] \begin{bmatrix} c_e & 0 & -c_x \\ 0 & c_e & -c_y \\ -c_x & -c_y & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0, \tag{20}$$

Where

$$c_e = E_1 - E_2$$

$$c_x = E_1 x_1 - E_2 x_2$$

$$c_y = E_1 y_1 - E_2 y_2$$

$$c = E_1(x_1^2 + y_1^2) - E_2(x_2^2 + y_2^2)$$

Setting $E_1 \neq E_2$ leads equation (20) to be simplified into

$$(x - \frac{c_x}{c_e})^2 + (y - \frac{c_y}{c_e})^2 = \frac{E_1 E_2 d_{12}^2}{c_e^2} \tag{21}$$

Which is a circle centred at $(\frac{c_x}{c_e}, \frac{c_y}{c_e})$ with radius $\frac{d_{12}}{c_e}\sqrt{E_1 E_2}$ where

$$d_{12} = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

This circle acts to constrain the locus of a sound source to lie on it in order for the microphones to receive E1 and E2 simultaneously.

In the case of $E_1 = E_2$, equation (20) simplifies to

$$2c_x x + 2c_y y = c \tag{22}$$

This is an equation of a perpendicular bisector bisecting the microphone axis mid way both microphones [25]. The figure below and is extracted from [25] indicating the resulting isocontours and line for both $E_1 \neq E_2$ and $E_1 = E_2$ scenarios.

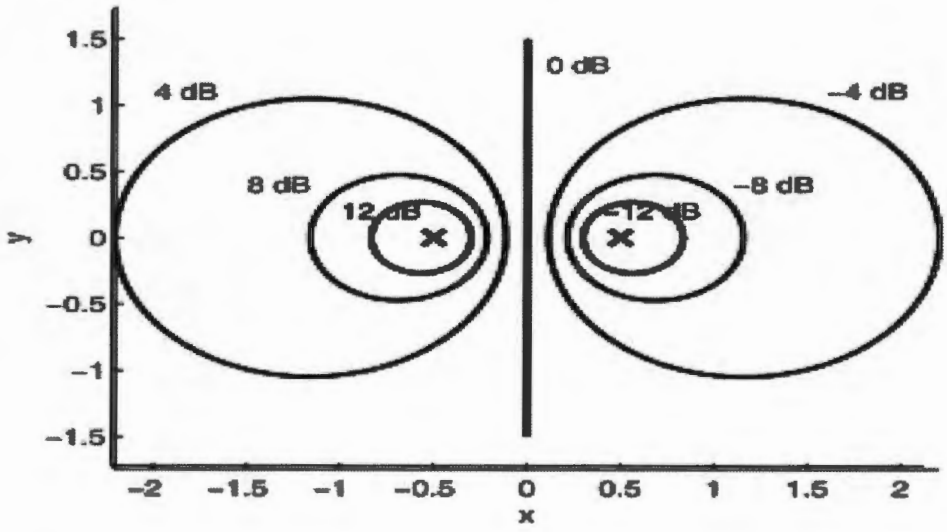


Figure 2.3-1 Isocontours and a line indicating possible source locations, based on range of dBs received from sound source. caption is taken from [25].

2.4 Head related transfer functions

Head Related Transfer Functions are auditory Transfer functions that characterise how the auditory sensors receive sound properties of a sound in space. These sound properties include; ILDs, TDOAs, and spectrum [5]

HRTFs vary from case to case based on geometrical differences of the objects for which they are being measured [5][4]. For humans, they are determined by shape of the pinna, head, and torso [26]. In [27], the effect of presence and absence of pinna was studied in a group of bats to analyse the impact it had on HRTF localisation. The absence of pinna was found to have detrimental impact on the ability to localise using HRTFs.

2.5 Beam forming

Beam forming, also known as spatial filtering is a means of source localisation that uses an array of sensors to search for spatial locations[28] with the highest signal energy. The signal components from the desired locations are enhanced through summation whilst others are attenuated as a result of destructive interference or weighting[29][30]. Beamforming is also used to perform sound source separation.

The simplest beam former is a sum and delay beamformer commonly referred to as the conventional beamformer [31][32] which aligns signals arriving at various microphones by delaying them in relation to the direction of signal's arrival. The signals are then summed in order to enhance the signal components from the desired direction through constructive interference.

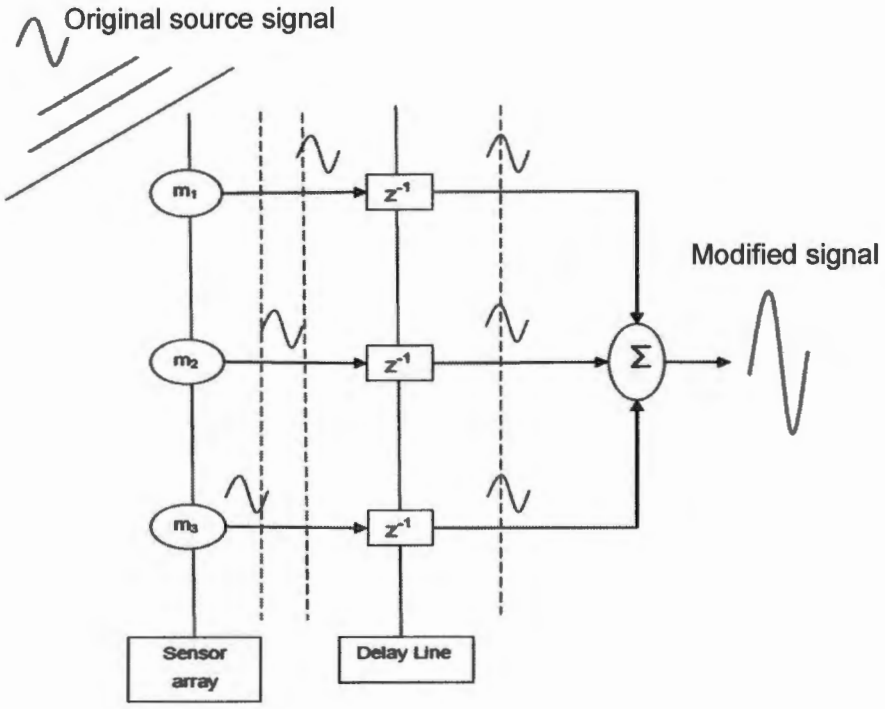


Figure 2.5-1 Delay and sum beam former illustration showing constructive interference.

The weights in the delay line are adjusted accordingly in order to steer the beam through the directions of interest. If the weights are left constant during operation, the beam former remains directed in one direction[33].

Mathematically, the sum-and-delay beamformer can be modelled as

$$y(n) = \sum_{m=0}^{M-1} x_m(n - \tau_m) \tag{23}$$

Where $x_m(n)$ is the mth microphone's signal and τ_m is the corresponding delay[1]. The energy (E) contained in a frame of length L can be computed as:

$$E = \sum_{n=0}^{L-1} y(n)^2 \tag{24}$$

E is hence maximised when the delays cause the microphone signals to be in phase. However, for broadband signals, the energy peak that ensues from this

method is broad and results in poor resolution[34]. This is partly due to a conventional beam-former's inability to uniformly attenuate noise and interference signals coming from directions other than the look direction over its entire spectrum when used for broad band acoustic signals [35]. Additionally, in the presence of multiple sources, the energy peaks relating to each source may overlap and become indistinguishable[1]. As a solution to the broad peaks problem, the microphone signals are whitened before processing the energy. An efficient way to processing the energy for a steered beamformer and its whitening is by utilising the frequency domain[1][33].

Minimum Variance Distortionless Response (MVDR), the Generalised Side lobe canceller (GSC), and the Steered Power Response (SRP-PHAT) are amongst the most widely used beamforming methods [33][32][24]. MVDR and GSC weights are defined in form of spatial correlation matrixes which normally require long segments of stationary data to estimate in adverse acoustic conditions. This makes these methods difficult to implement for speech localization which is largely characterized as short temporal and spatial stationary[32].

The biggest downside to beam forming in the past has been the computational demand for processing data from the large number of microphones used in the arrays. The resolution and accuracy of the system in most beamforming methods, improves as the number of microphones is increased [28][32]. This makes Beamformers unsuitable for this project due to the constraints listed in Chapter 1.

2.6 Cone and sphere of confusion

Binaural sound source localisation systems relying entirely on level difference or time delay alone, are generally only capable of localising sources in the azimuth[36]. Kneip and Baumann[36] give a detailed study of localisation geometry involved for binaural sensors using TDOA methods.

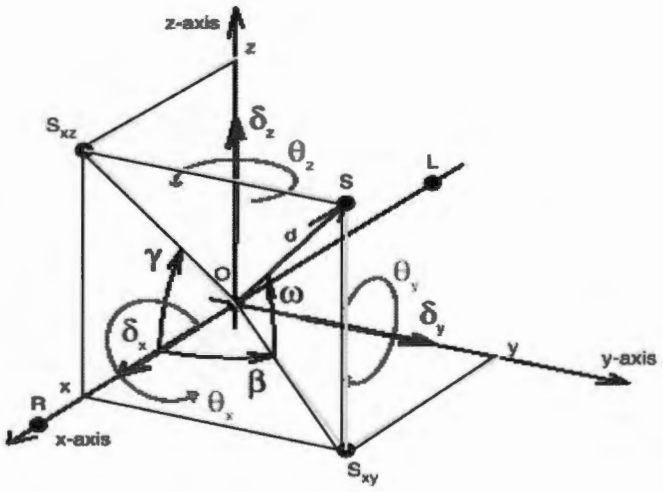


Figure 2.6-1 illustration by Kneip & Baumann of localization coordinate system centered between the left (L) and right (R) microphones [36], **where β is azimuth γ is elevation.**

Due to symmetry, of a linear binaural sensing axis, a sound source located in front of the axis using TDOA methods will yield the same direction estimate as a source located at the back [37]. This translates to a hyperbolic surface of rotation that is symmetrical around the sensor axis[38]. This hyperboloid is approximated asymptotically by a cone of confusion[36].

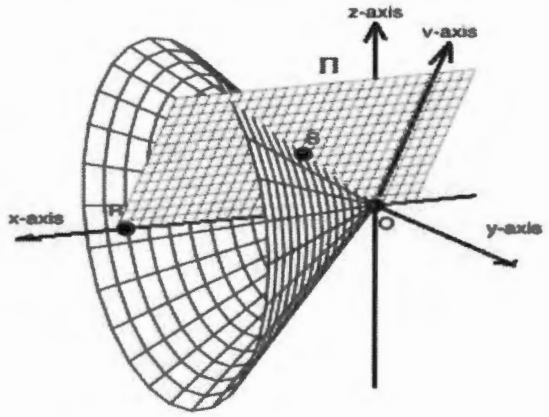


Figure 2.6-2 illustration by Kneip and Baumann of the cone of confusion [36]

In the case of ILD methods, isocontours together with a line of confusion are yielded by sound sources generating given ranges of dBs as was shown in fig 2.3-1. These isocontours extend into Spheres and the line into a plane of confusion when the system is extended from azimuth localisation to 3D localisation [25].

In order to solve these cones and spheres of confusion, the sensor axis must at least have 2 degrees of freedom (DOF) [36].

2.7 Binaural Source Separation

Sound source separation is an active field of research that has been widely studied for more than a decade; mainly, as a means to solve the cocktail party phenomenon [26]. The cocktail party phenomenon(CPP), which was first coined up by Cherry in 1953 [39], refers to a situation in which a person can focus on one acoustic source in a noisy environment [40].

Various binaural methods have been proposed to solve this problem, such as [26][41][18][42][43]. A common assumption amongst most of these methods is W-disjoint orthogonality, a condition which assumes that only one sound source amongst several sources in an environment, is acoustically active during each time-frequency representation [44]. A time-frequency representations is a Short Time Fourier Transform (STFT); a technique ideal for analysis of quasi-stationary signals such as speech[45].

In [26], the authors use clustering, and spectral masking, also known as binary masking, in order to facilitate separation of an arbitrary number of sources. They do mention that only a few methods exist, formally combining 2D localisation and separation. Based on the W-disjoint orthogonality assumption, they 'filter' the signal by weighting the target source time-frequency segments with 1, and the rest with 0. A ratio of spectrograms of the left and right sensors is used to establish the Inter-aural spectrogram, from which, training data can be used to estimate the location of a source in space. This method utilises a human-like head with HRTFs in order to resolve the cone of confusion.

3 Methodology

This chapter begins by explaining the choice of research method used for the project. An illustration of the methodology is given in form of a flow chart and explained in detail thereafter. A system design is then presented based on findings from chapter2. The chapter then concludes by outlining the experiment and analysis procedure.

3.1 Research Method

An agile model of development was used for this project. It embedded both experimental research, and build research methodologies.

An experimental research methodology is made up of two phases: an exploratory phase and an evaluation phases. A build research methodology entails constructing either a physical artefact or software system to demonstrate its feasibility [46]

The exploratory phase is required to determine hidden parameters for the proposed design that may not be directly mentioned in the initial general research questions. The evaluation phase will then be used to analyse the effects of varying the parameters.

3.2 Proposed approach

Fig 3.2-1 provides a flow representation of the methodology used for this research dissertation. The subsequent sections within this chapter are used to explain what each stage of the flow diagram entails.

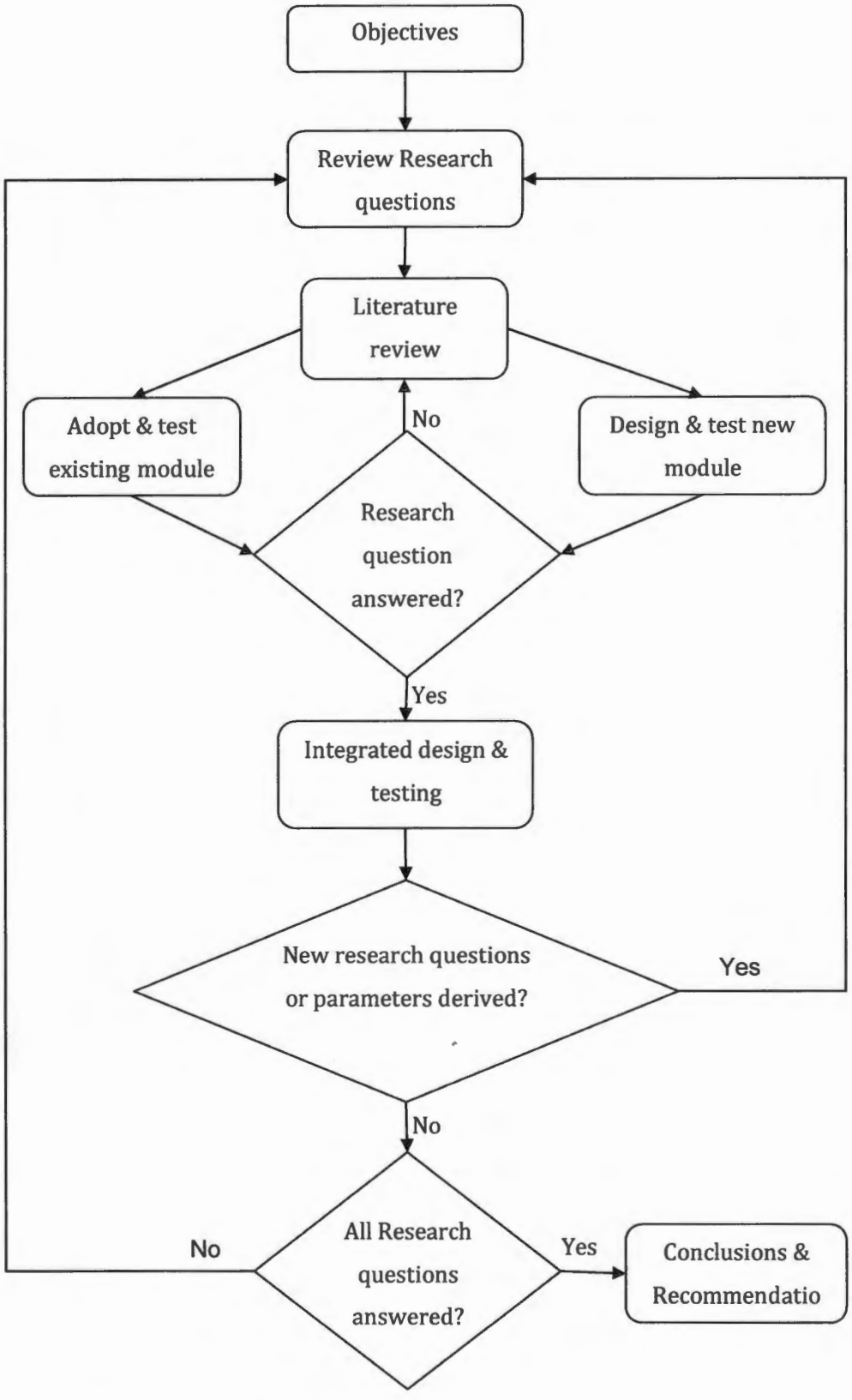


Figure 3.2-1: Methodology flow diagram

3.2.1 *Objectives*

The project starts out by listing objectives that need to be met in order to decide if the project is successful or not. From these objectives, a list of research questions is established to guide the research.

3.2.2 *List Research questions*

In Chapter 1, a broad set of research questions was established. Each of these research questions may however have underlying questions of their own. This stage of the research is used to list out the broad questions and pass them onto the next phase one at a time.

3.2.3 *Literature review*

Existing literature is used to evaluate and understand what each research question involves by first of all understanding what has been done so far, the problems that were faced, and what is yet to be done within similar fields of study. This helps one determine whether there exist modules that can be adopted to solve the given problem at hand, or if a new module needs be designed to solve the problem.

3.2.4 *Adopt & test existing modules*

At this stage, existing modules from the literature are adopted to suit the purpose of this project. The adoption could be made in hardware, software or both. An example would be adopting a portion of an existing algorithm to work on android based smart phones if it only existed for custom controllers. Once the module has been modified, it is tested and documented prior to moving onto the next phase of the methodology.

3.2.5 *Design and test new module*

This stage of the methodology is run compliment to the previously mentioned phase. In this phase, if literature indicates that there does not exist a readily available solution to the research question at hand, a new solution is proposed and developed. The solution is then tested and documented prior to commencing the next phase.

3.2.6 *Research question answered?*

The individual module tests results help determine whether or not the corresponding research question has been answered. If the research question has been satisfactorily answered, the flow progresses to the next stage. Else, the flow loops back to analysing literature to determine if another solution is probable through the previous branches mentioned – Adopt existing modules, or designing new ones.

3.2.7 *Integrated design and testing*

Due to the agile approach of this project, integrated design and testing are carried out at each iteration when either separate modules are being integrated or when new functionality is being added. This is to ensure that additional functionality at later iterations is added to a correctly functioning framework.

3.2.8 *New sub questions or parameters derived?*

The individual modules may present new sub questions or parameters either as a result of integration or during the process of their design. If no new questions or parameters are present at this point, flow continues downwards onto the next phase. Otherwise, if new sub questions or parameters are present, flow is looped back to research questions to integrate the new questions and parameters.

3.2.9 *All research questions answered?*

If only a subset of the total questions have been answered up to this stage, flow is directed back to the first stage to iterate through the next research question. Once all questions have been answered, flow is directed towards the final stage of the methodology.

3.2.10 *Conclusions & recommendations*

Conclusions are drawn based on results acquired from the testing stages. The overall system is analysed to assess how well it meets the project's objects through answering the research questions.

Recommendations are then made for future work to overcome limitations faced by the designed system. The future work recommendations are also used to highlight how the project is to be extended into other work.

3.3 System Design

From literature, the W-disjoint assumption is to be used as a guideline for the system design. In other words, an assumption is made that amongst active sound sources in an environment; only one source is active during each time-frequency unit. The proposed high level design of the system is shown below.

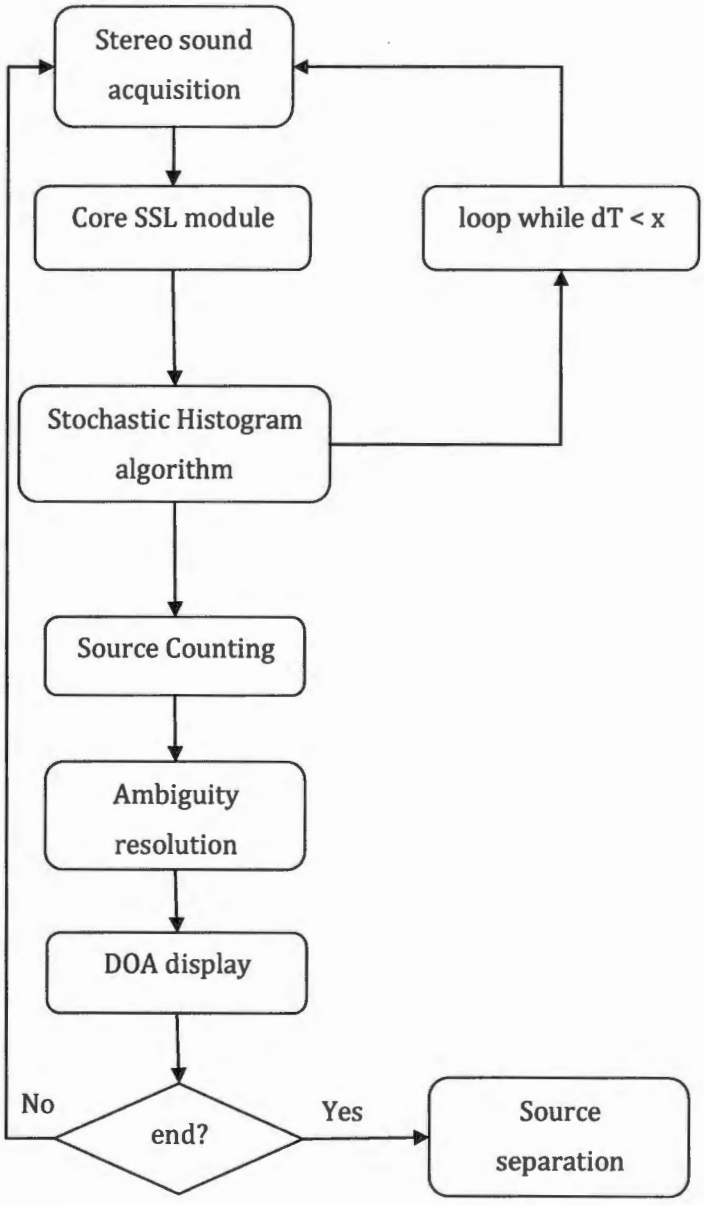


Figure 3.3-1 High level system design

The sound acquisition is to be handled using classes provided by the Android API. The GCC-PHAT algorithm is chosen from literature as the core SSL module. The Stochastic Histogram is inspired and developed from the assumption of W-disjoint sound sources. A crucial parameter dT, emerges as a result of the histogram algorithm. It is used to set the number of time frames used to accumulate DOA estimations across DOA bins in a histogram.

Source counting is added as a research question and to the design on a later iteration of the agile process after results from experiment 5 lead to a hypothesis of its possibility.

Ambiguity resolution is developed upon phone orientation data acquired from a module developed in[47].

The user interface is designed under the DOA module to provide an easy to understand visual representation of active sound sources in terms of their DOA and perceived power levels.

Results from experiment 5 and 6 prompt the question of whether the W-disjoint assumption can be used to perform time domain source separation as a continuation to the proposed framework of this project. A source separation module is hence developed to separate sound sources after the DOAs and source counting has been performed.

3.4 Experiments & Analysis

Unit tests are performed at preliminary iterations (experiment 1 to 4) of the agile process to ensure that fundamental modules work properly prior to addition of more functionality. Subsequent iterations with added functionality lead to integration testing as is the case from experiment 5 to 9.

Due to the real time multimedia nature of this project, for test analysis, the system is embedded with code to generate log files and media files that can be used to analyse the results from the tests.

3.4.1 Experiment 1: GCC-PHAT Auto-correlation tests

This test is designed to check if GCC-PHAT indeed works as portrayed in literature. Correlation sample lags and leads are simulated using phased auto-correlations of both channels. Sharp peaks are expected to be seen at the positions on the display axis relating to the simulated lags and leads. Results from corresponding un-weighted GCC are used to create further emphasis on benefit of using GCC-PHAT

3.4.2 Experiment 2: GCC-PHAT processing time

These tests are part of the preliminary iterations meant to establish feasibility of implementing such a system. This particular set of tests is expected to show processing times of GCC-PHAT on the test android device being a fraction of the time represented by a sound frame acquired by the acquisition system. The results are presented in milliseconds.

3.4.3 Experiment 3: DOA display tests

DOA display tests are meant to test the user interface for correctness in mapping the correlation information into DOAs. The system was expected to display lines representing the estimated DOA. The length of the lines is proportional to the perceived signal power in decibels. Depending on the perceived sound power, the line is expected to change colour to green, magenta or red. Green symbolising relatively quiet environments with less than 65dB sound levels; magenta indicates environments with sound levels similar to normal

conversations at 3ft; and lastly, red to act as a warning for sound levels close to those that may cause long term hearing damage.

3.4.4 *Experiment 4: Phone orientation tests*

The phone orientation tests are meant to test orientation data generated by the module designed by Lawitzki[47] . These tests help provide an idea of what shortcomings should be expected of the ambiguity resolution module.

3.4.5 *Experiment 5: Determining optimum dT value for stochastic histogram algorithm*

This is probably the most crucial experiment, since it is used to establish a value of dT that would yield the most accurate results. This set of tests inherently tests the system's accuracy under different test conditions. The system is investigated for accuracy values associated with dT = 1, 5, 11, 23, and 43 at test distances of 0.5m, 1m, and 2m for a set of expected DOAs{10^o, 50^o, 90^o, 130^o, 170^o}.

3.4.6 *Experiment 6: Source counting tests*

These tests are used to verify the hypothesis that cumulative power histograms resulting from experiment 5 can be used to perform source counting in addition to source localisation. For these tests, three active sound sources are placed 1m away, at 45^o apart with reference to the centre of test phone. The sound sources are centred at 90^o. The three sources used are: a pulse source represented as 'P'; a music source (the one used in experiment 5) represented as 'M'; and finally a speech source represented as 'S'. The sources positions are permutated as

- S M P
- S P M
- P S M

Gaussian-like peaks are expected to be observed centred at DOA positions corresponding to the actual sound source positions.

3.4.7 *Experiment 7: Ambiguity resolution tests*

These tests were designed to analyse whether it was possible to resolve front-back ambiguity using phone orientation data. Source at the front side of the

microphone are expected to be displayed as positive numbers in the log files, where as sources at the back are expected to be indicated as negative values in the corresponding DOA bins in the log file. This would translate to having positive columns on a column graph for sources at the front side and negative columns for sources at the back side of the sensor axis.

3.4.8 Experiment 8: System processing time

The system processing time tests are for the purpose of analysing the average time taken to yield a DOA estimate after the sound signals have been acquired by both microphones. A comparison amongst results from the best three dT settings established in experiment 5 are used to verify outcomes of the best dT setting.

3.4.9 Experiment 9: Source separation

Source separation tests were meant to test whether it would be possible to perform source separation in the time domain as a continuation to the results attained from experiment 5 and 6. The idea to perform source separation in the time domain was inspired by the fact that most methods described in section 2.7, used a combination of frequency and time domain to perform source separation.

In these tests, the system is expected to create wav output files labelled according to the DOAs of the available active sound sources. The wav files are expected to contain the corresponding sound generated by a source in that direction.

A master wav file containing the mixed received stereo sounds is also generated for purposes of analysis in this project, but would be used for further processing in future work.

4 System Design

Looking back at the objectives and research questions at hand, the project requires building a system that meets the following criteria:

- Performs sound source localisation on a microphone utilising 2 in-built microphones
- Performs source counting and separation, and provides results in a usable format by 3rd party systems
- Attempts to resolve front-back ambiguities using orientation data from on-board sensors
- Has a relatively fast processing time for performing the source localisation

4.1 Available equipment

The device available for this project is a Samsung Galaxy S3 smart phone which has:

- Quad-core cortex-A9 CPU operating at 1.4GHz.
- 2 microphones spaced 13.5cm apart
- Android 4.3 operating system
- Accelerometer, Gyroscope, Compass, Proximity, Barometer

Due to proprietary technology, the phone schematics are un-available to establish the exact type of ADC being used for the Audio Sampling. However, from the Android SDK developer's guide, 44100 Hz is the only sampling rate guaranteed to work on all android devices as of the time this project was carried out. Other sampling rates that may work on other Android devices are 22050, 16000, and 11025 [48]. The guide also provides provision for acquiring audio recordings either Mono or Stereo, which implies that some devices may indeed have 2 channel ADCs. However, no provision is indicated for 3 channel Audio sampling from the ADC.

4.2 Binaural localisation method of choice

As of the time this project was carried out, to the best of available knowledge, there existed no android smart phone with more than 2 microphones accessible through the SDK. This led to the need to design for a 2-microphone system device with comparably lower processing power compared to personal computers.

Beam forming methods attain better accuracies with a higher number of microphones. Their computational cost is however higher than cross correlation method. Subspace methods such as MUSIC are too computationally demanding to opt to implement on mobile phone given the other user and system applications that may be competing for the same processing resources. HRTFs require transfer functions to be established based on the shape of the device, however, mobile phones vary in shapes and sizes and the user would have an impact on the estimated transfer function. In addition, smart phone users have a variety of phone cases to choose from which could affect the estimated transfer function as well.

The logical choices for this project are hence the GCC-PHAT algorithm and Intensity difference approach due to their computational simplicity and suitability for a 2 microphone system.

4.3 Inter-Aural Time Difference considerations

Referring back to equation (7),

$$\theta = \arccos(\varphi)$$

where $\varphi = c * D_s/(F_s * m_s)$

The system design becomes constrained by

$c = 345$ m/s

$F_s = 44100$ samples per second

$m_s = 0.135$ m

As a result, θ varies as a function of D_s and can be attained from a look-up table as generated below

Table 4.3-1 Look up table relating D_s to θ, with sampling rate at 44100Hz

Delay in samples	Distance (meters)	Angle θ (radians)	Angle θ (Degrees)	Resolution (Degrees)	φ
0	0	1.570796327	90		0
1	0.007823	1.512814739	87	3.32	0.057949
2	0.015646	1.45463707	83	3.33	0.115898
3	0.023469	1.396061187	80	3.36	0.173847
4	0.031293	1.33687233	77	3.39	0.231796
5	0.039116	1.276835377	73	3.44	0.289746
6	0.046939	1.215685122	70	3.50	0.347695
7	0.054762	1.153113321	66	3.59	0.405644
8	0.062585	1.088750502	62	3.69	0.463593
9	0.070408	1.022139171	59	3.82	0.521542
10	0.078231	0.952692263	55	3.98	0.579491
11	0.086054	0.879624955	50	4.19	0.63744
12	0.093878	0.801834891	46	4.46	0.695389
13	0.101701	0.717672556	41	4.82	0.753338
14	0.109524	0.62444542	36	5.34	0.811287
15	0.117347	0.517140247	30	6.15	0.869237
16	0.12517	0.383967538	22	7.63	0.927186
17	0.132993	0.172639399	10	12.11	0.985135
18	0.140816	#NUM!	#NUM!	#NUM!	1.043084

As mentioned in chapter 2, the arccos is only valid for the range $\varphi \in [-1,1]$ and since D_s is an element of all positive integers relating to possible lags, the maximum lag is determined by the last value at which $\varphi \leq 1$. In the rows where

$\varphi > 1$, a string '#NUM!' is generated in the angles and resolution columns to indicate that the computation is out of bounds. This is a critical criterion as it bounds the search space for methods that utilize time difference of arrival.

It is worth noting the effect that reducing sampling frequency, has on the resolution and maximum lag of the system. For example, taking the second highest available sampling rate (22050) as stated in the SDK developer's manual[48], the maximum number of lags reduces to 8 samples as compared to 17 in the case of 44100 Hz sample rate.

Table 4.3-2 Look up table relating D_s to θ with sampling rate at 22050Hz

Delay in samples	Distance (meters)	Angle θ (radians)	Angle θ (Degrees)	Resolution (Degrees)	φ
0	0	1.570796327	90		0
1	0.015646	1.45463707	83	6.66	0.115898
2	0.031293	1.33687233	77	6.75	0.231796
3	0.046939	1.215685122	70	6.94	0.347695
4	0.062585	1.088750502	62	7.27	0.463593
5	0.078231	0.952692263	55	7.80	0.579491
6	0.093878	0.801834891	46	8.64	0.695389
7	0.109524	0.62444542	36	10.16	0.811287
8	0.12517	0.383967538	22	13.78	0.927186
9	0.140816	#NUM!	#NUM!	#NUM!	1.043084

To get a better understanding of the effect of sampling frequency on the system, a graph is made of the outputs of the look up tables for cases when Fs is set to 22050Hz, 44100Hz, and 99800Hz. The results are as displayed in Figures 4.3-1, 4.3-2, and 4.3-3 respectively.

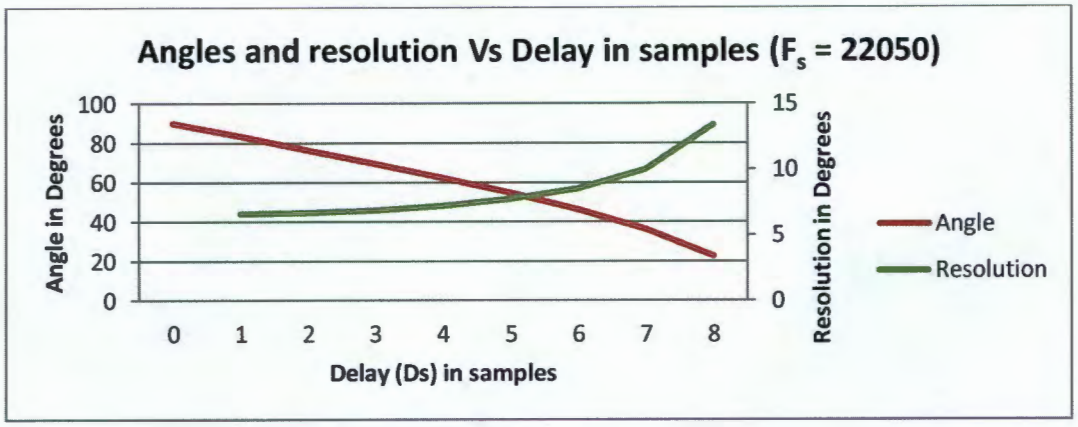


Figure 4.3-1 Angles and resolution Vs Delay in samples (F_s = 22050)

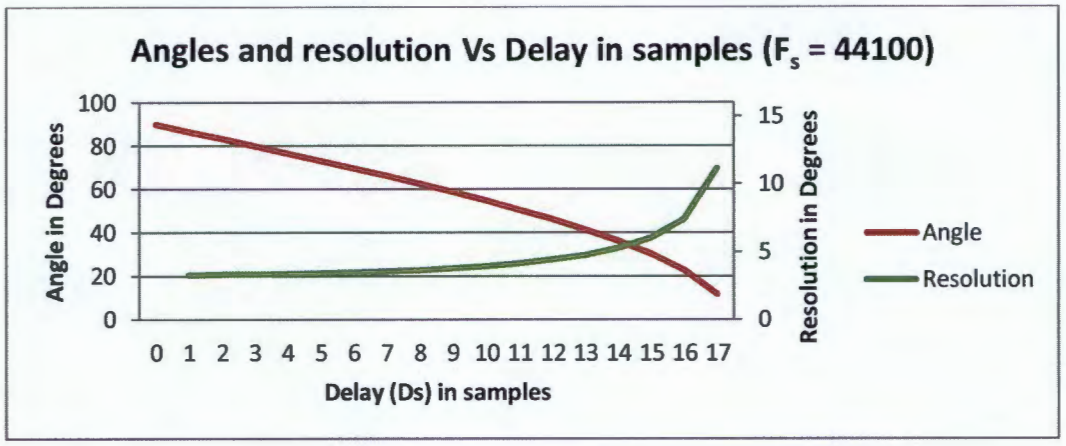


Figure 4.3-2 Angles and resolution Vs Delay in samples (F_s = 44100)

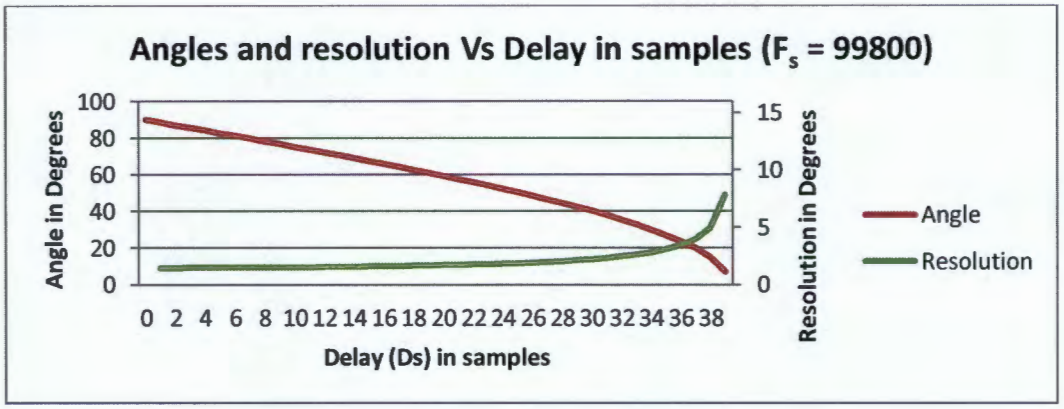


Figure 4.3-3 Angles and resolution Vs Delay in samples (F_s = 99800)

The first observation one can easily make, is the reduction in number of sample delays that the system can estimate as sampling frequency is reduced. At 22050Hz, the system can only approximate 9 sample delays (0 to 8), 18 sample delays (0 to 17) at 44100Hz, and 40 (0 to 39) at 99800Hz. This in turn affects the number of incident angles that the system can estimate.

The next thing to take note of is the lowest (or maximum) angle towards the sensor axis that can be estimated.

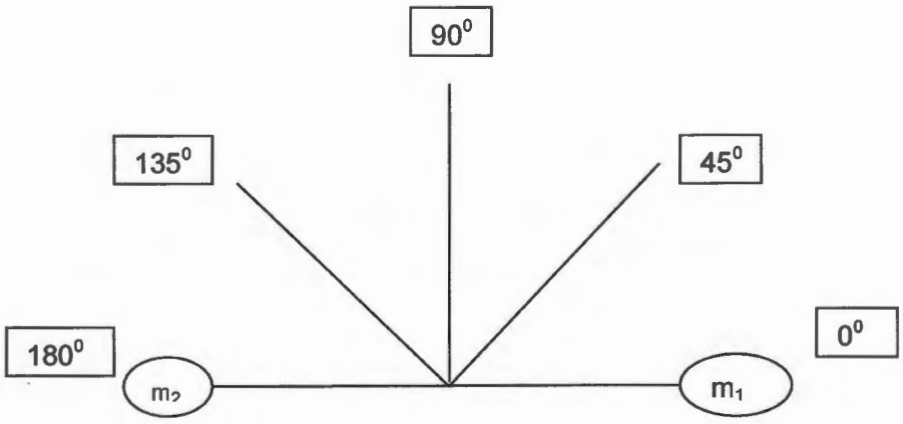


Figure 4.3-4 illustration of microphone orientation with reference to sources

As depicted in figure 4.3-4, the angles range from 0 to 180 degrees. The minimum angle is the angle relating to the maximum sample delay in the look up tables. The maximum angle is derived as(**180^0 – minimum angle**). The look angle is the difference between the maximum and minimum angle. Table 4.3-3 tabulates these values for the three sampling rates at hand

Table 4.3-3 Relationship between sampling frequency and detectable angles

Fs	# of sample delays (lags)	Minimum Angle (Degrees)	Maximum Angle (Degrees)	Look-Angle (Degrees)	Look-Angle Difference between adjacent sampling Frequencies (Degrees)	
22050	9	22	158	136	36	
44100	18	10	170	160		14
99800	40	3	177	174		

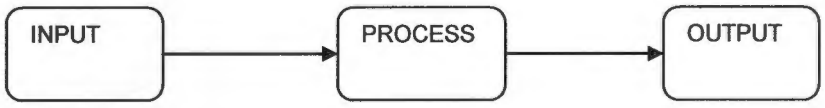
The Look angle difference is the difference between look angles of two consecutive sampling frequencies. It is aimed at comparing improvement in look-angle as the sampling frequency is increased. From the results in Table 4.3-3, the look-angle improves as the sampling frequency increases. However, the improvement as sampling frequency is increased from 44100Hz to 99800Hz, is less than half the improvement when sampling frequency is increased from 22050Hz to 44100Hz. One can hence confidently choose to use 44100Hz sampling frequency as it meets the Nyquist minimum sampling requirement for audible sound and has a good compromise between resolution, look-angle, number of sample delays, and implications on processing speed.

4.4 Digital Signal processing considerations

The default Audio buffer size from ADC provided by the android SDK on the Samsung S3 being used for this project is 8192 bytes. This translates to 4096 16-bit samples and hence 2048 samples per microphone. 2048 samples per second translate to roughly 46ms. To ensure a balance between resolution, signal stationary period, processing speed and algorithm stability; It is opted to utilise half of the buffer data in order to keep data buffered. This leads to 1024 samples per second for each microphone to process as a frame, approximately equivalent

to 23ms. To improve FFT resolution and avoid cyclic convolution during the FFT conversions, the sound frames are zero padded to 2048 samples per channel prior to taking the FFTs.

4.5 Arbitrary High level design



4.5.1 INPUT

- SOUND
  - microphone 1 signal
  - microphone 2 signal
- ORIENTATION DATA
  - Accelerometer data
  - Gyroscope data
  - Compass data

4.5.2 PROCESSING

- Sound conversion from PCM to floats
- Signal power calculations
- GCC-PHAT processing
- Stochastic direction estimation
- Orientation Data processing

4.5.3 OUTPUT

- source direction display in real time
- orientation data display
- summary log data file
- Audio master file
- separated audio files

4.6 Detailed High level design

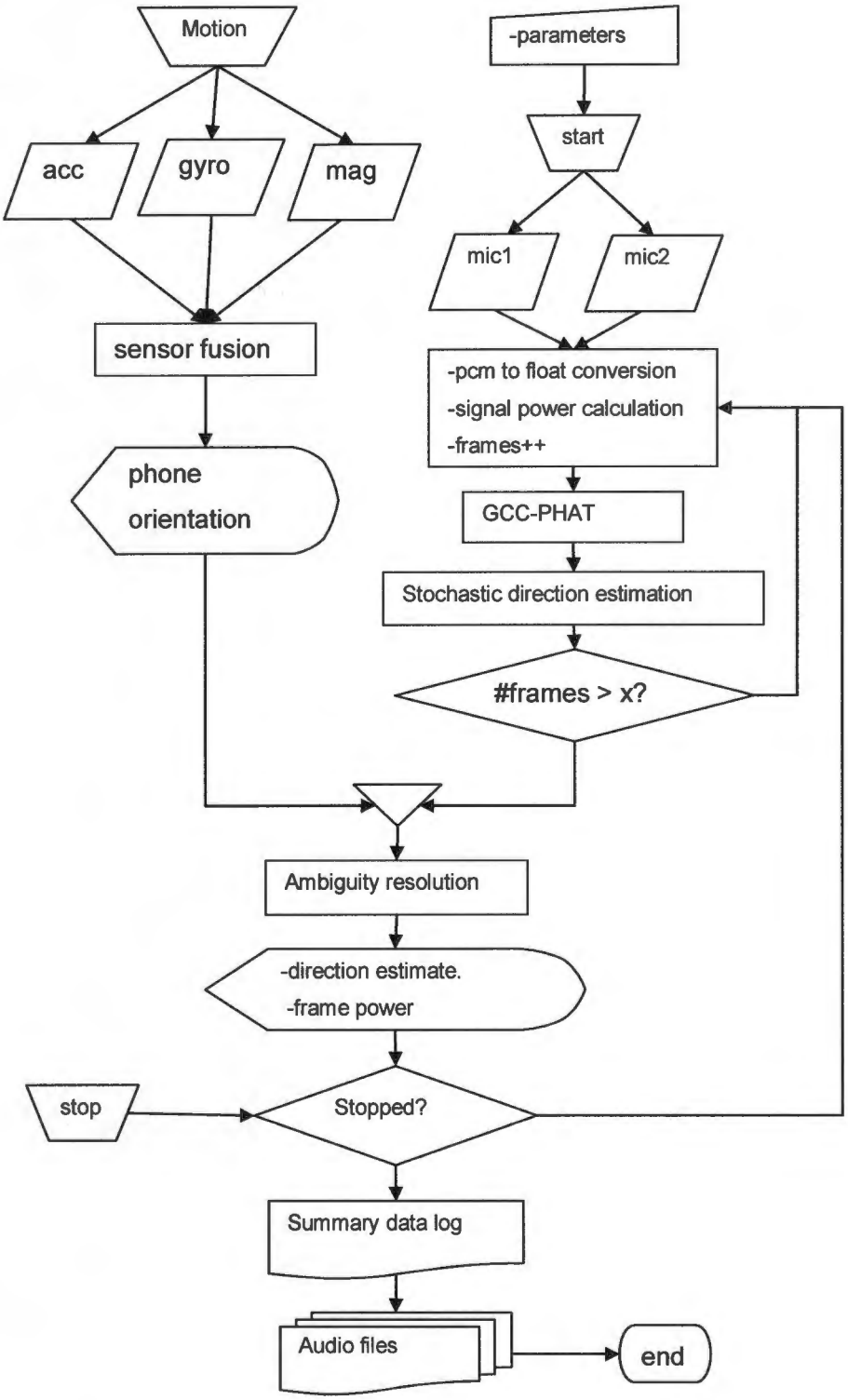


Figure 4.6-1 Flow diagram for detailed design

4.7 Signal acquisition and conversion

Sound signals are captured by on board microphones and converted to Pulse Coded Modulation format by the ADC. The Android SDK abstracts the underlying hardware coding requirements from the developer by providing a simple software interface. This interface is in form of a class that takes in various parameters as listed in table 4.7-1. The AudioRecord class is used to provide this service and detailed specifications may be found in[48]. The class constructor and sample usage are as shown below:

AudioRecord(int audioSource, int sampleRateInHz, int channelConfig, int audioFormat, int bufferSizeInBytes)

```
recorder = new AudioRecord(MediaRecorder.AudioSource.CAMCORDER,
                           RECORDER_SAMPLERATE,
                           RECORDER_CHANNELS,
                           RECORDER_AUDIO_ENCODING, bufferSize);
```

{1}

Table 4.7-1 AudiRecord class definition

Parameter	Value	Purpose
audioSource	MediaRecorder.AudioSource.*CAMCORDER*	Enables stereo recording using main microphone and camera microphone
sampleRateInHz	*RECORDER_SAMPLERATE = 44100*	Sound Sampling rate set at 44100Hz as per reasons

		discussed in section 4.3
channelConfig	*RECORDER_CHANNELS = AudioFormat.CHANNEL_IN_STEREO = 12*	
audioFormat	*RECORDER_AUDIO_ENCODING = AudioFormat.ENCODING_PCM_16BIT = 2*	The format in which data is read out the ADC. We use PCM 16bit because it is guaranteed to work on all devices and provides for a better sampling resolution.
bufferSizeInBytes	bufferSize = 8192 bytes	The size of the buffer to which the audio data is written to during sampling.

The code snippet in {1} shows the declaration and initialization of an AudioRecord instance called 'recorder'. With this instance initialized, one can then call member functions to start the recorder, read the data, stop the recorder, and finally release it.

public int read (short[] audioData, int offsetInShorts, int sizeInShorts)

The 'read' function writes the data into a byte array indicated here as 'audioData'. The stereo data is written into the byte array as interleaved pairs. The first pair being from one microphone and the second from the other as indicated in the figure below

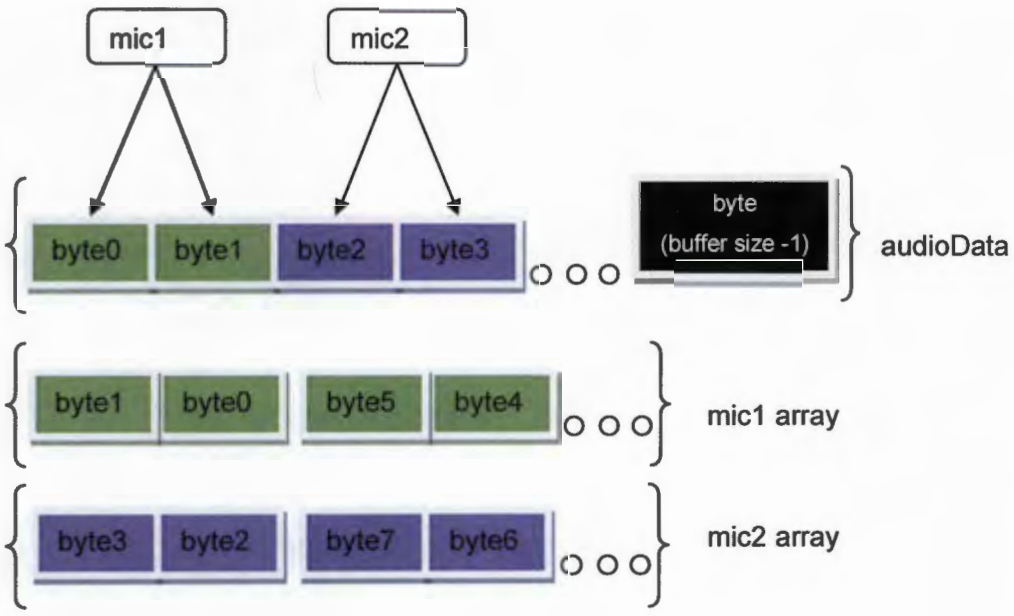


Figure 4.7-1 Byte extraction from ADC

The data from the audioData buffer has to be un-interleaved and placed accordingly into separate arrays. One needs to take note that the interleaved data is in little Endean format and hence byte order needs to be reversed as indicated in Figure 4.7-1. Once the data is extracted in rightful byte order, it is methodically converted and normalized into floating point formats which are conventionally used for sound signal processing. The figure below illustrates the conversion process of 2 bytes to a float

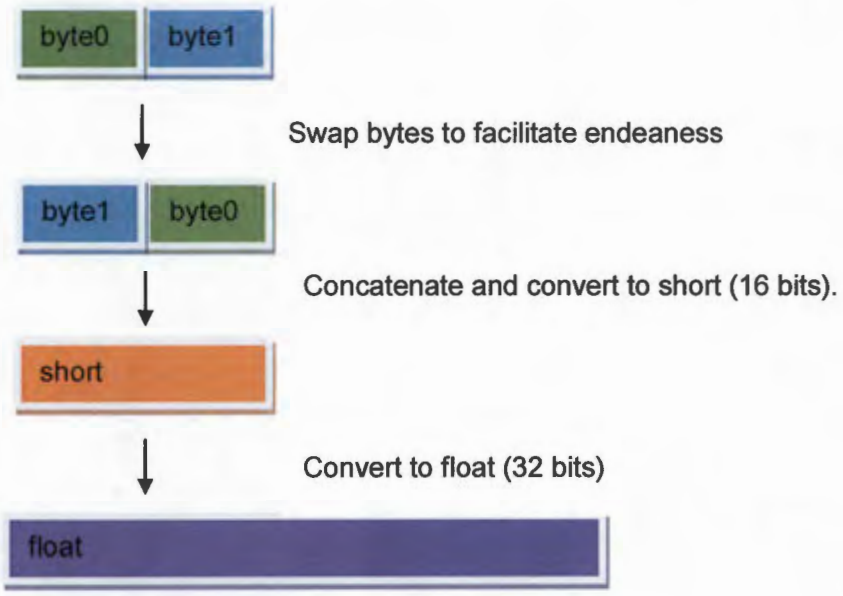


Figure 4.7-2 byte conversion to float

```
for(int i=0; i<(bufferSize2 - 4); i+=4) //run through the raw audio data buffer
{
  //type cast then normalised with limit of 32768 if negative else 32767
  temp_short = (short)(data[i] | (data[i+1]<<8));
  if(temp_short < 0){
      tempdata =  ((float)temp_short)/((float)32768);
  }
  else{
     tempdata = ((float)temp_short)/((float)32767);
  }

 temp_short = (short)(data[i+2] | (data[i+3]<<8));
 if(temp_short < 0){
      tempdata2 = ((float)temp_short)/((float)32768);
  }
  else{
     tempdata2 = ((float)temp_short)/((float)32767);
  }

  mic1[c] = tempdata;
  mic2[c] = tempdata2;
 // mic2[c+dT] = tempdata + tempdataPrev;
  //test line, comment out after tests.
  //use dT box on user interface to vary this delay for correlation
  //tests


  mic1Power += tempdata*tempdata;  //power levels
  mic2Power += tempdata2*tempdata2; //power levels
  c++;

}
```

Figure 4.7-3 screen shot of java implementation section for byte conversion

4.8 GCC-PHAT

Once both microphone signals have been placed in their respective arrays and zero padded to avoid cyclic convolution, they are transformed to the frequency domain by means of FFTs. One of the resulting FFTs is conjugated before it is multiplied with the other. Taking the conjugate ensures positive contribution to the integral by aligned peaks or troughs with imaginary components.

Given the time series signals are converted to X1[f], and X2[f] respectively, the pseudo code for GCC-PHAT is as listed below.

$$X_1[k] = \mathrm{FFT}(x_1(t), \mathrm{fftsize})$$

$$X_2[k] = \mathrm{FFT}(x_2(t), \mathrm{fftsize})$$

$$G_{12}[k] = X_1[k]X_2^*[k], \text{ where } * \text{ denotes conjugate}$$

$$G_{\mathrm{denom}}[k] = \max\left(|G_{12}[k]|, 1e^{-6}\right)$$

$$G_r[k] = \frac{G_{12}[k]}{|G_{12}[k]|}$$

$$R_{12}^{\mathrm{raw}}[k] = \mathrm{FFT}^{-1}\{ G_r[k]\}$$

$$R_{12}^{\mathrm{phat}}[k] = \mathrm{FFTshift}\{ R_{12}^{\mathrm{raw}}[k]\}$$

$$R_{12}^{\mathrm{phat}}[\tau] = \mathrm{argMax}\{R_{12}^{\mathrm{phat}}[k]\}$$

The time domain signals are converted to frequency domain using an FFT libgdx library developed by 'badlogic games' [49]. The library is designed to perform high speed FFTs on android devices.

4.9 Stochastic Direction estimation from GCC-PHAT result

Even though GCC-PHAT bears the advantage of sharpening correlation peaks, it faces the disadvantage of being sensitive to finite observation interval errors particularly where signal power is lowest[15]. So, for a system expected to be used in realistic environments including indoor environments where reverberations and air conditioning systems have a large presence, the system is likely to suffer from erratic direction estimations due to reverberations and low signal power conditions.

This is where this research aims to make contribution to this field of study, by providing a means to filter out the wrong estimates caused by low signal power, and to ignore or suppress the effect of 'ghost' sources that occurs as a result of reverberations.

As a solution, a simple histogram technique is proposed that uses the GCC-PHAT result to determine DOA per sound frame, and histogram the corresponding highest signal-power from either microphone for that particular frame. The hypothesis is that by using power level, direction estimates with low signal power would contribute less to the overall histogram poll. The overall DOA estimation is made by choosing the DOA in the histogram with the highest value/peak. The histogram values are accumulated over a given number of frames (dT). The number of frames is set as a critical variable parameter to be studied in order to determine what value best optimises for accuracy and response time. For example, setting this parameter to 1 would mean that the DOA estimate displayed to the end user, is made over only one frame, making the system response very fast but most likely un-visually pleasant with a low accuracy rate. This is because visual response time in humans is slower than auditory response times [50] and since it is auditory data that we are presenting in a visual context, we would need to identify suitable values for this parameter that would provide an ideal visual perception of dominant sound source directions.

The algorithm is hence proposed as follows:

1. Initialise an array whose length is equivalent to the sum of valid number of lags and leads one microphone signal can have over the other. let's call it DOA_array
2. For each sound frame, determine the power contained in each microphone signal. Choose the microphone signal with the highest power to be assigned to a variable called max_power.

3. Acquire the direction estimate of the sound frame from the GCC-PHAT module and use it to index the corresponding direction index in the DOA_array.
4. Add the max_power value to the value at this index
5. Proceed to next sound frame and repeat from 2 until the preset number of frames has been reached.
6. Select the index with the highest values.
7. Add this estimate to the DOA annotation vector
8. Reset the DOA_array and repeat from 2.

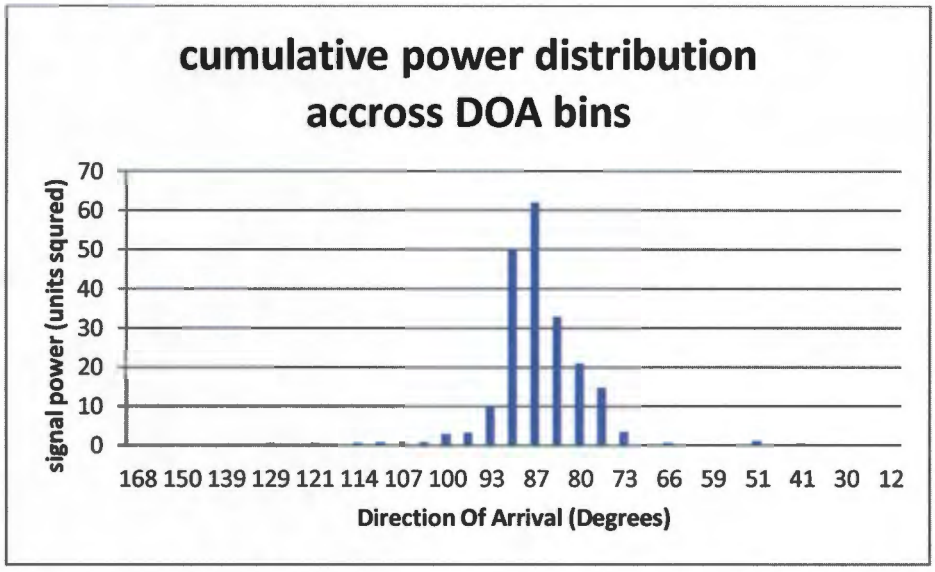


Figure 4.9-1 illustration of expected out come

The estimated directions along with the accumulated power value are then passed to the plotting module.

4.10 Orientation data

The sensor fusion demo by Paul Lawitzki [47]was adopted for this project to provide orientation data.

Lawitzki developed a phone orientation tracking algorithm aimed at tracking head-positions for his Masters project. The system was developed and implemented on a Samsung Galaxy s2. The system fuses data from the phone's MEMS onboard sensors to periodically estimate orientation of the phone.

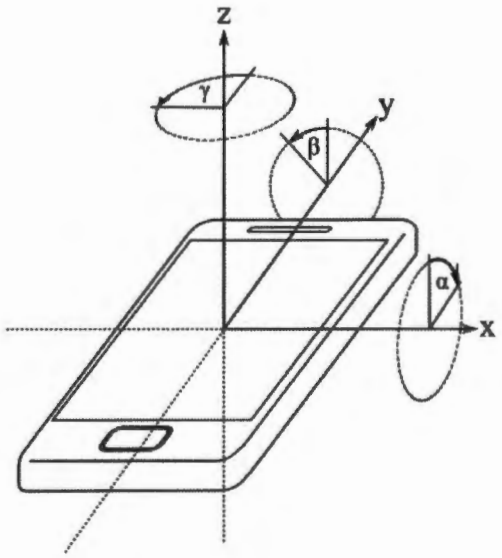


Figure 4.10-1 Standard coordinate system for android systems. illustration acquired from [47]

A complimentary filter was used to fuse data from the 3 MEMS sensors (gyroscope, accelerometer, and magnetometer) in order to balance out the downsides of each sensor.

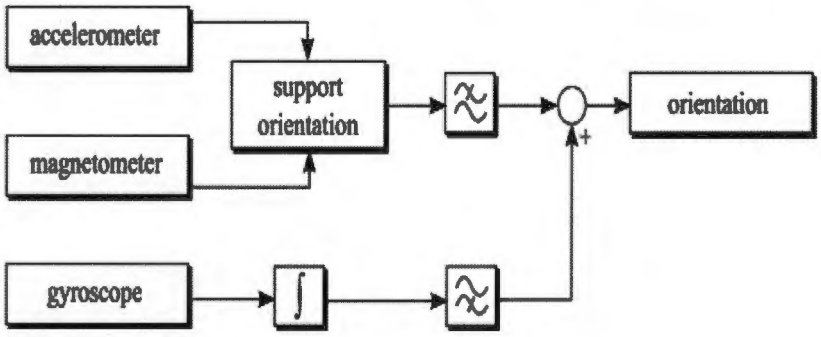


Figure 4.10-2 complementary filter implementation as illustrated in [47]

In order to make the system more flexible, one of the modules is modified as indicated in fig 4.7-5 to use 3 global variables for azimuth, pitch and role values in the switch case statement. This enables one to either use orientation data calculated from magnetron and accelerometer sensors, gyroscope, or all three fused together under sensor fusion. The variables are; 'fAzimuth' for azimuth, 'fPitch' for pitch, and finally fRoll for roll.

```
public void updateOreintationDisplay() {
    switch(radioSelection) {
    case 0:
        fAzimuth = accMagOrientation[0] * 180/Math.PI;
        fPitch = accMagOrientation[1] * 180/Math.PI;
        fRoll = accMagOrientation[2] * 180/Math.PI;

        mAzimuthView.setText(d.format(fAzimuth) + '°');
        mPitchView.setText(d.format(fPitch) + '°');
        mRollView.setText(d.format(fRoll) + '°');
        break;
    case 1:
        fAzimuth = gyroOrientation[0] * 180/Math.PI;
        fPitch = gyroOrientation[1] * 180/Math.PI;
        fRoll = gyroOrientation[2] * 180/Math.PI;

        mAzimuthView.setText(d.format(fAzimuth) + '°');
        mPitchView.setText(d.format(fPitch) + '°');
        mRollView.setText(d.format(fRoll) + '°');
        break;
    case 2:
        fAzimuth = fusedOrientation[0] * 180/Math.PI;
        fPitch = fusedOrientation[1] * 180/Math.PI;
        fRoll = fusedOrientation[2] * 180/Math.PI;

        mAzimuthView.setText(d.format(fAzimuth) + '°');
        mPitchView.setText(d.format(fPitch) + '°');
        mRollView.setText(d.format(fRoll) + '°');

        break;
    }
}
```

Figure 4.10-3 screen shot of orientation data code

4.11 Ambiguity resolving

In order to resolve ambiguity, rotation data is utilised from the orientation module in the following algorithm

1. Take initial DOA estimate from the SSL module. Assume that the value is either positive or negative.
2. At the same time, take the initial orientation data readings
3. When a new DOA is made by the SSL module, establish the rotation made by the phone during the elapsed period from initial readings. Add this value to both assumed initial DOA estimates. This yields two expected new-DOAs new-A and new-B.
4. Take the new DOA estimated by the SSL module and compare it to both new-A, and new-B. The expected new-DOA it most closely approximates is taken as the actual DOA

```
private void ambiguityResolver(){
    Qria=Qrfa;  //initial orientation angle
    Qrfa = (int)fAzimuth; //current orientation angle
    Qra = Qria - Qrfa;  //Rotation made from initial to current angle

    Qsslia = Qa; //initial estimated ssl position
    Qssla = angleRad[inda];  //current ssl position

    Qxpva= Qsslia - Qra;
    Qxnva = Qsslia*(-1) - Qra;

    Qpva = Math.abs(Qssla - Qxpva);
    Qnva = Math.abs(Qssla - Qxnva);

    if(Qpva < Qnva){
        Qa = angleRad[inda];
    }
    else{
        Qa = angleRad[inda]*(-1.0f);
        fpositions[inda] = vala*(-1.0f); //flip column to represent back
    }


}
```

Figure 4.11-1 ambiguity resolving code

4.12 Results Display

To display estimated DOA results, the system uses a canvas running on a separate thread to continuously redraw lines using trigonometric functions that use the DOA angle in radians and associated signal power from the SSL module as parameters. Since the values are stored in global variables, the display module is able to access the values without interfering with the DOA estimation process. The two modules hence perform calculations on separate threads, which tremendously improves throughput of the system to the point of being relatively real time.

```
//*******************surface view stuff*****************
protected void onDraw(Canvas canvas) {
        canvas.drawRGB(255, 255, 255);

    }

    protected void DrawOnSurface(Canvas canvas) {
        canvas.drawRGB(255, 255, 255);

        paint.setColor(Color.RED);
        painta.setColor(Color.GREEN);
        paintb.setColor(Color.MAGENTA);
        paintc.setColor(Color.RED);
        paintL.setColor(Color.BLACK);
        paint.setStrokeWidth(4);
        painta.setStrokeWidth(4);
        paintb.setStrokeWidth(3);
        paintc.setStrokeWidth(3);
        paintL.setStrokeWidth(2);

        float  scale=0.125f;
```

Figure 4.12-1 canvas display initialization code

The signal power accumulated for each cluster used to estimate a DOA, is converted from amplitude^2 to decibels using the equation

$$power_{dB} = 10\ log_{10}(amplitude^2/0.0000025)$$

This leads to a dB range of [0,105] given the settings from section 4.4. The figure below shows the android canvas code that uses the dB power from a cluster to set the length and color of a line displayed on the phones screen to indicate the DOA and related power.

```
if(AvframePower_dB <= 65){
    paintx = painta;
     scale=0.004f;
}
else{
    if(AvframePower_dB > 65 && AvframePower_dB < 80){
        paintx = paintb;
         scale=0.006f;
    }
    else{
        paintx = paintc;
         scale=0.008f;
    }
}

int h = canvas.getHeight();
int w = canvas.getWidth();

float ix =0, iy=0, sx =0, sy =0;
float dB = (float) AvframePower_dB;


ix = w/2; iy =h/2; //canvas centre

canvas.drawLine(0, iy, (float)w, iy, paintL);
canvas.drawLine(ix, 0, ix, (float)h, paintL);

sx = ix + (float) (scale*dB*ix*Math.sin(Qa)); //ambiguity resolution
sy = iy - (float) (scale*dB*iy*Math.cos(Qa)); //ambiguity resolution
canvas.drawLine(ix, iy, sx, sy, paintx);
```

Figure 4.12-2 canvas display code

The line colors are divided into three groups to provide quick visual warning when the sound levels progress into regions that are related to high risk.

4.13 Source counting & separation

From single source tests, a pattern emerged in the cumulative power distribution arrays that alerted us to the possibility of performing source counting and hence separation by creating an extension to the already proposed SSL algorithm. This extension would have to computationally determine how many Gaussian-like peaks are present in the cumulative power distribution array the positions at which these peaks are centred would indicate the estimated DOA. These estimated DOAs are then used to create corresponding wav files to which sound Frames with closely associated DOAs are written in post processing. The algorithm is listed in point form as follows;

1. Take Power accumulation array and smooth the data using a 3 point moving average.

$$x = \frac{x_{-1} + x + x_{+1}}{3}$$

2. Pick out max value from smoothed data
3. populate a peaks' vector with the indices of peaks/maxima from the smoothed data that meet the criteria of being at least $\frac{1}{4} \times$ max _value
4. Create wav files labelled and index-able based on the contents of the peaks' vector
5. Use DOA annotation vector to write frames being read from raw data file into suitable wav files based on the estimated DOA associated with them.

5 Experiments, Results, & Analysis

This chapter is split up into subsections covering experiments outlined in chapter 3. Each experiment subsection is further split into three subsections:

1. Setup and expected hypothesis
2. experiment results
3. Analysis and conclusion

5.1 Experiment 1: GCC-PHAT Auto-correlation tests

This experiment was designed to verify if the implementation of GCC-PHAT on the android device yielded results similar to those in literature. The tests make use of component re-use by utilising the drawing canvas to plot the time domain correlation array, and using the dT input box to enter the simulated lags.

5.1.1 Setup and expected hypothesis

GCC-PHAT as explained in chapter 2.2.2 is expected to yield sharp peaks at points relating to a delay between a received pair of signals in ideal conditions with high signal to noise ratios. To establish if the correlator works, auto-correlations are performed using each channel. Each channel is duplicated into the second channel where a simulated lag is created by shifting the duplicated signal based on a user set value. The procedure is detailed below:

1. In the program, set
$$x_2(t) = x_2(t)$$
$$x_1(t) = x_2(t+\tau)$$
2. Control τ using the input box for dT. It can be varied in discrete intervals from 0 to 17. For this test, the values 0, 8, and 17 are used.
3. Compare GCC-PHAT outputs with those of GCC by commenting out the PHAT weighting from the program.
4. Repeat from step 1 to 3 using
$$x_1(t) = x_1(t)$$
$$x_2(t) = x_1(t+\tau)$$

Expected results

For GCC-PHAT with $x_1(t)$ being used as the reference signal, a single dominant peak is expected to be seen at points relating to the number of samples by which $x_2(t)$ lags $x_1(t)$. All these points except for the zero lag, should fall on the left side of the centre point of the display since the display axis for this experiment has been rotated -90° from the sensor axis. Similarly, in the case of $x_2(t)$ being used as the reference signal, the dominant peak should be observed on the right side to indicate number of samples by which $x_1(t)$ lags $x_2(t)$.

Results from GCC without PHAT weighting should exhibit broadened peaks with the global maxima positioned at positions equivalent to the number of samples by which $x_1(t)$ lags or leads $x_2(t)$.

5.1.2 Experiment results

The figure below is a screen shot of both GCC-PHAT and GCC for comparison.

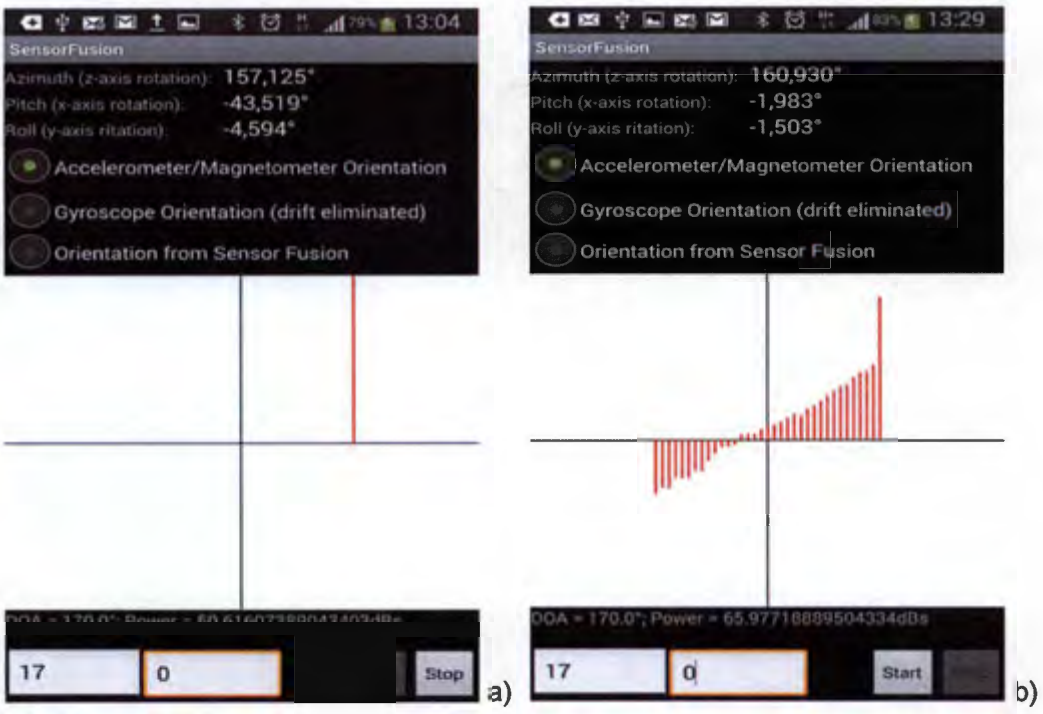


Figure 5.1.2-1 a) GCC-PHAT and b) un-weighted GCC with simulated $x_2(t)$ lead of 17 samples

In the figure above, signal $x_2(t)$ leads signal $x_1(t)$ by 17 samples. GCC-PHAT appears to have a single dominant peak whilst GCC has residue peaks alongside the main peak.

The next figure is a pair of screen shots showing the correlation outputs when signal $x_2(t)$ is simulated to lead $x_1(t)$ by 8 samples.

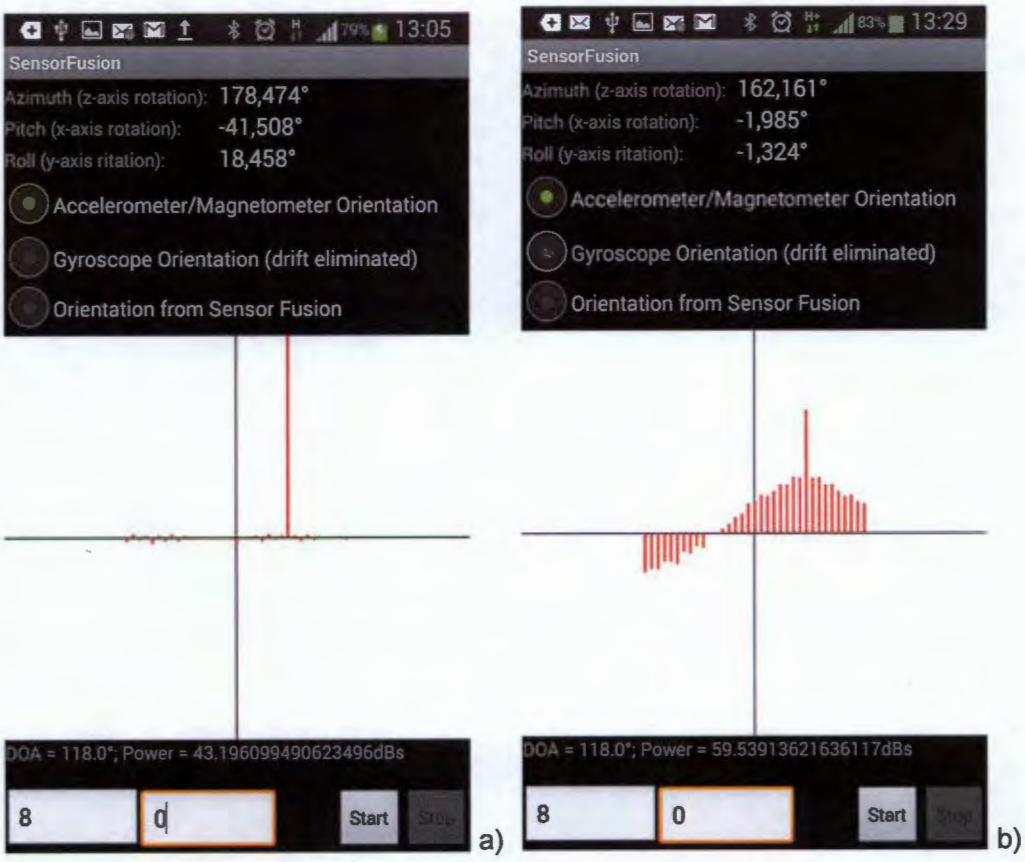


Figure 5.1.2-2 a) GCC-PHAT and b) un-weighted GCC with simulated $x_2(t)$ lead of 8 samples

The outputs for both the weighted and un-weighted version of GCC are seen to indicate a lead of 8 samples as expected.

The auto-correlation below, results from auto-correlating either $x_2(t)$ or $x_1(t)$ by itself without any simulated shifts as has been the case of the results thus far.

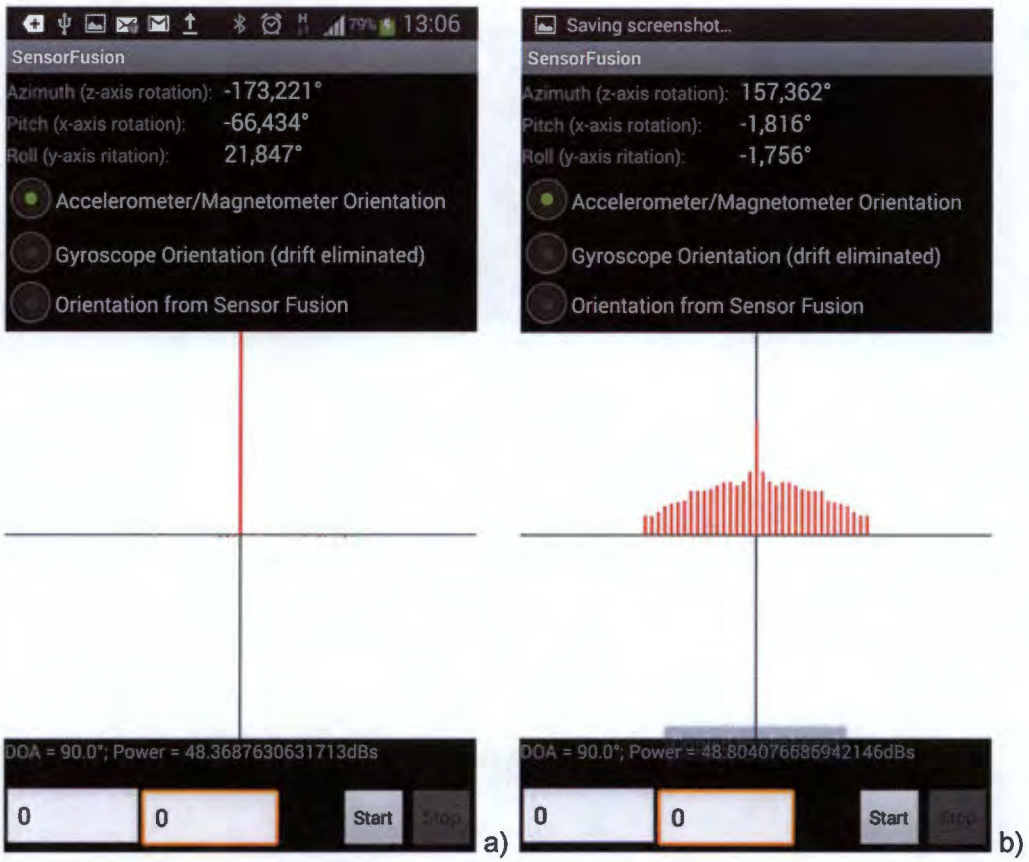


Figure 5.1.2-3 Auto-correlation by a) GCC-PHAT and b) un-weighted GCC.

Both methods peak at zero, with GCC-PHAT maintaining a single dominant peak. GCC exhibits a symmetrically broadened peak centred at the zero lag/lead mark.

For the next two figures, $x_2(t)$ has been simulated to lag $x_1(t)$ by 8 samples and 17 samples respectively.

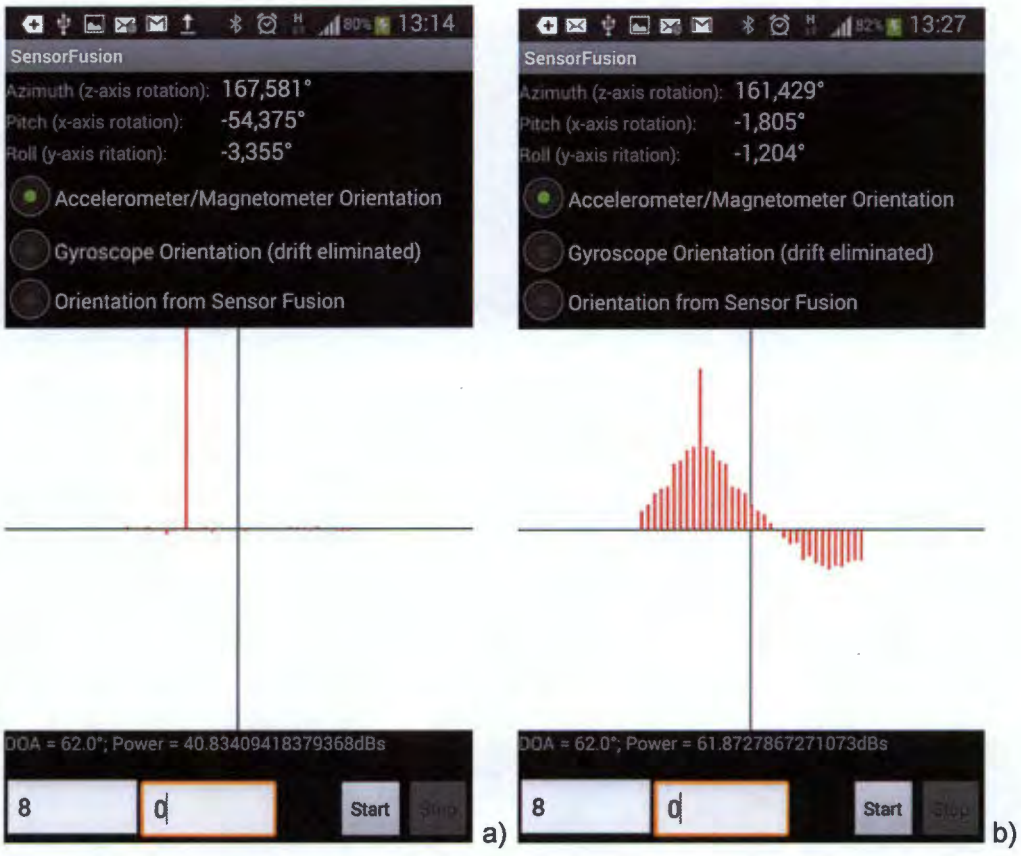


Figure 5.1.2-4 a) GCC-PHAT and b) un-weighted GCC with simulated $x_2(t)$ lag of 8 samples

A lag of 8 samples is obtained for both GCC-PHAT and GCC results displayed above.

Finally, the results for a simulated lag of 17 samples are shown below in the final auto-correlation test.

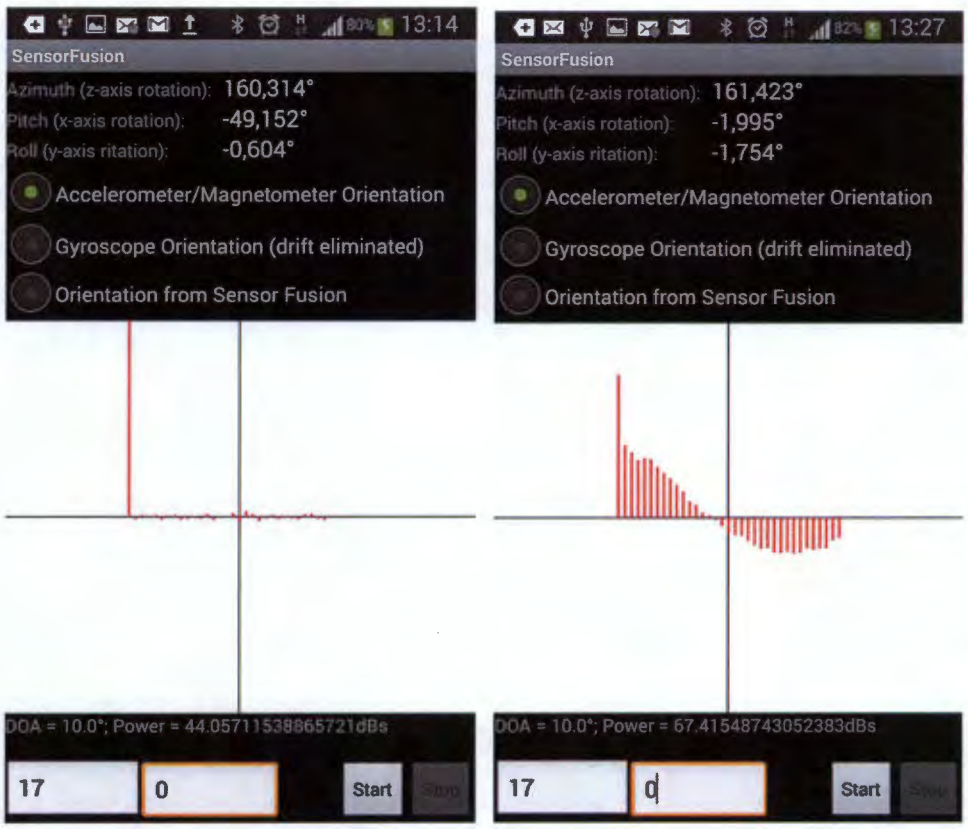


Figure 5.1.2-5 a) GCC-PHAT and b) un-weighted GCC with simulated $x_2(t)$ lag of 17 samples.

The final results exhibited similar results as all other cases in the sense that the exhibited lag was equivalent to the expected lag.

5.1.3 Analysis & conclusion

All GCC-PHAT tests exhibited sharp correlation peaks for all simulated lags and leads. This was in agreement with literature which states that GCC-PHAT yields sharp peaks in environments of high SNR. The GCC screen shots were merely used as a means to emphasise the effects of PHAT weighting.

This test acted as a unit test to verify whether the GCC-PHAT module yielded satisfactory results. The results attained from the tests were very satisfactory with 100% accuracy, indicating that the basis SSL module was fit to be integrated into the system.

5.2 Experiment 2: GCC-PHAT processing time

It was imperative that the processing time of GCC-PHAT be analysed to ensure that the system is capable of performing in real time. This test analyses the time taken to process the critical points of the GCC-PHAT algorithm keeping in mind that the processing speed of the processor used is 1.4GHz as mentioned in section 4.1.

5.2.1 Set up and expected hypothesis

Timing code is embedded in the system to establish processing time taken over a region being analysed. The procedure for acquiring elapsed time is shown below:

startTime = **android.os.SystemClock.uptimeMillis**();

{critical section}

stopTime = **android.os.SystemClock.uptimeMillis**();

$\boldsymbol{timeTaken = stopTime - startTime;}$

The critical sections to be analysed for the GCC-PHAT algorithm are explained as follows:

- PCM to 2 float array conversions: This is the conversion of 23ms worth of sound from PCM to two normalised float arrays representing signals $x_1(t)$ and $x_2(t)$.
- FFT processing: Time taken to convert one time domain signal to the frequency domain.

  $X_1[k] = FFT(x_1(t), fftsize)$
- Multiplication and PHAT weighting: This is the section in which one of the frequency domain signals is conjugated, multiplied by the second frequency domain signal, and their product weighted by PHAT.

  $G_{12}[k] = X_1[k]X_2^*[k]$, where $*$ denotes conjugate

  $G_{denom}[k] = \max(|G_{12}[k]|, 1e^{-6})$

  $G_r[k] = \frac{G_{12}[k]}{|G_{12}[k]|}$

- iFFT processing: This is the time taken to perform the inverse Fourier transform of the correlation.
- GCC-PHAT overall processing: This is the total time taken from PCM to float conversion up until the resultant iFFT array is mapped into a smaller bounded correlation array.

The processing times listed above are logged into a modified version of the csv log file used for experiment 5.

5.2.2 Experiment results

The table below lists the average processing time recorded for each critical section listed in the previous section. Each average was made over 361 samples.

Table 5.2.2-1 Average processing times for GCC-PHAT critical sections

CRITICAL SECTIONS	PROCESSING TIME [milliseconds]
PCM to 2 float array conversions	0.39ms
FFT processing (one FFT)	2.42ms
Multiplication and PHAT weighting	1.81ms
iFFT processing	3.19ms
GCC-PHAT overall processing	10.32ms

5.2.3 Analysis & conclusion

The overall processing time for GCC-PHAT is found to be almost half of the time represented by a sound frame. FFT conversions constitute the most time which totals to about 5ms for both signals.

It can hence be concluded that the GCC-PHAT algorithm is executed at 2 X real time, which makes it feasible to be integrated into a Histogram algorithm to filter out wrong estimates caused by low SNR, noise, and reverberation.

5.3 Experiment 3: DOA display tests

This set of tests was designed to ensure that the correlation results were mapped appropriately to the corresponding directions on the mobile device's display as designed in section 4.12. The tests were also meant to ensure that decibel changes in received sound, were properly integrated into the DOA display to act as a visual warning system for: when someone is exposed to high levels of sound that might be harmful to the auditory system; when a fire alarm or horn is sounded; when an active sound source such as a car is approaching a user.

5.3.1 Set up and expected hypothesis

The procedure for these tests is largely identical to that of experiment 1. Only difference being that only GCC-PHAT is used and the DOAs displays are tested for three different dB settings:

- power < =65dB -> region A
- 65< power<80dB -> region B
- power>=80dB -> region C

Sound sources producing dB levels falling in region A, should result in the DOA line turning green. For region B, the line should turn magenta, and lastly, for region C, the line should turn red.

5.3.2 *Experiment results*

The first set of results is from a sound source simulated to be at DOA=170° as shown in the figure below.

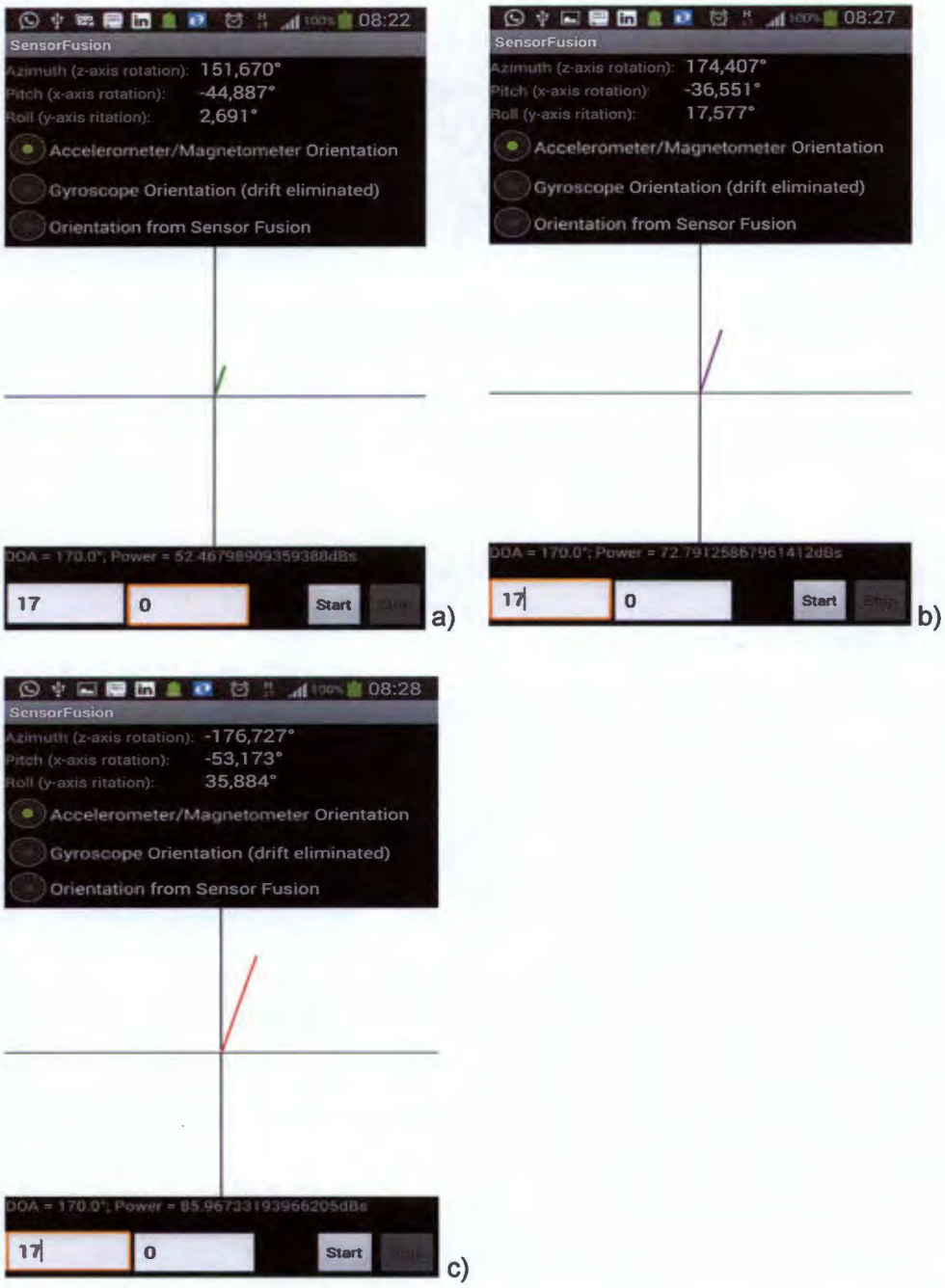


Figure 5.3.2-1 Display result for a) region A, b) region B, and c) region C at DOA=170°

The second set of results illustrates the system's display when a sound source is simulated to be located at 118°.

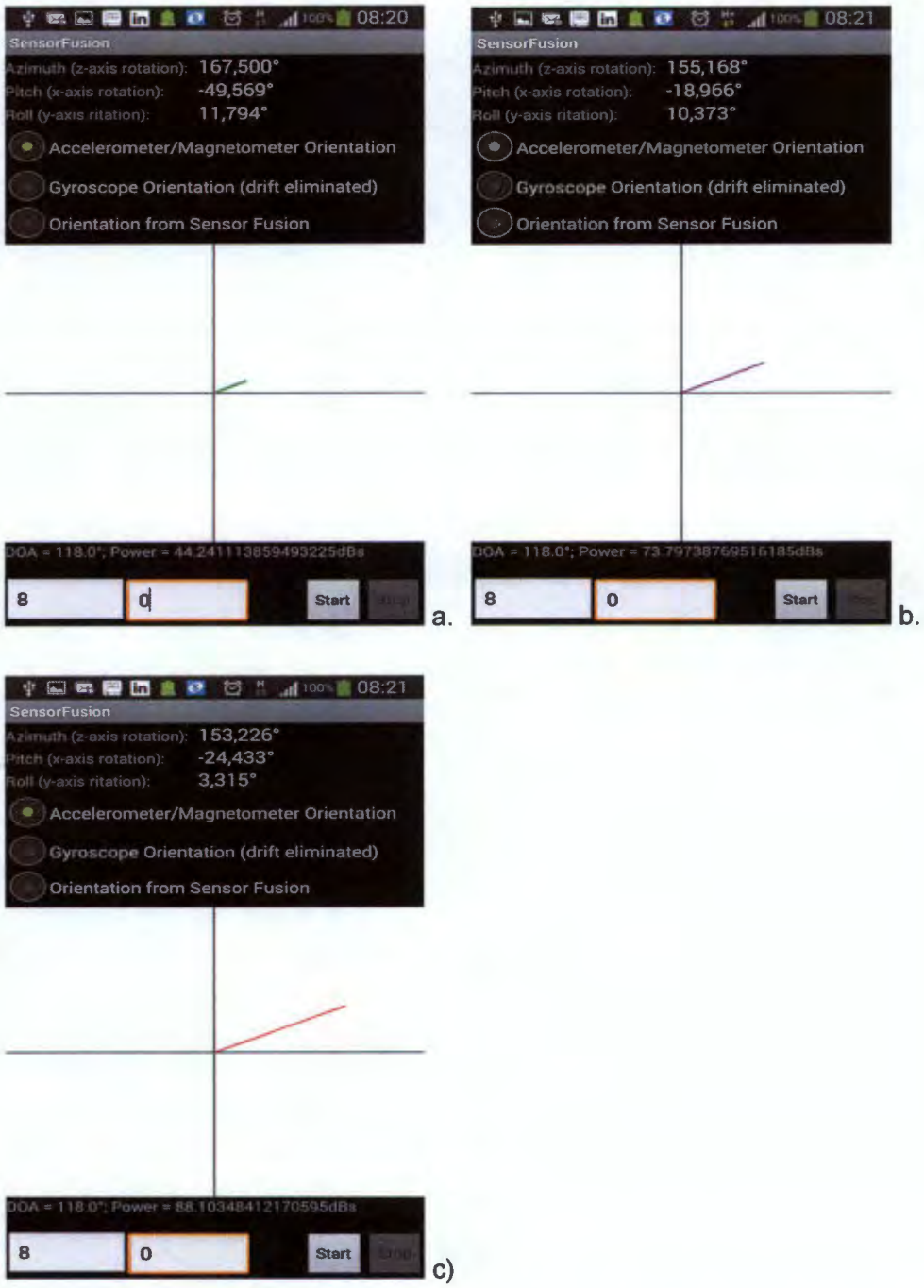


Figure 5.3.2-2 Display result for a) region A, b) region B, and c) region C at DOA=118°

The third set of results illustrates the system's display when a sound source is simulated to be located at 90°.

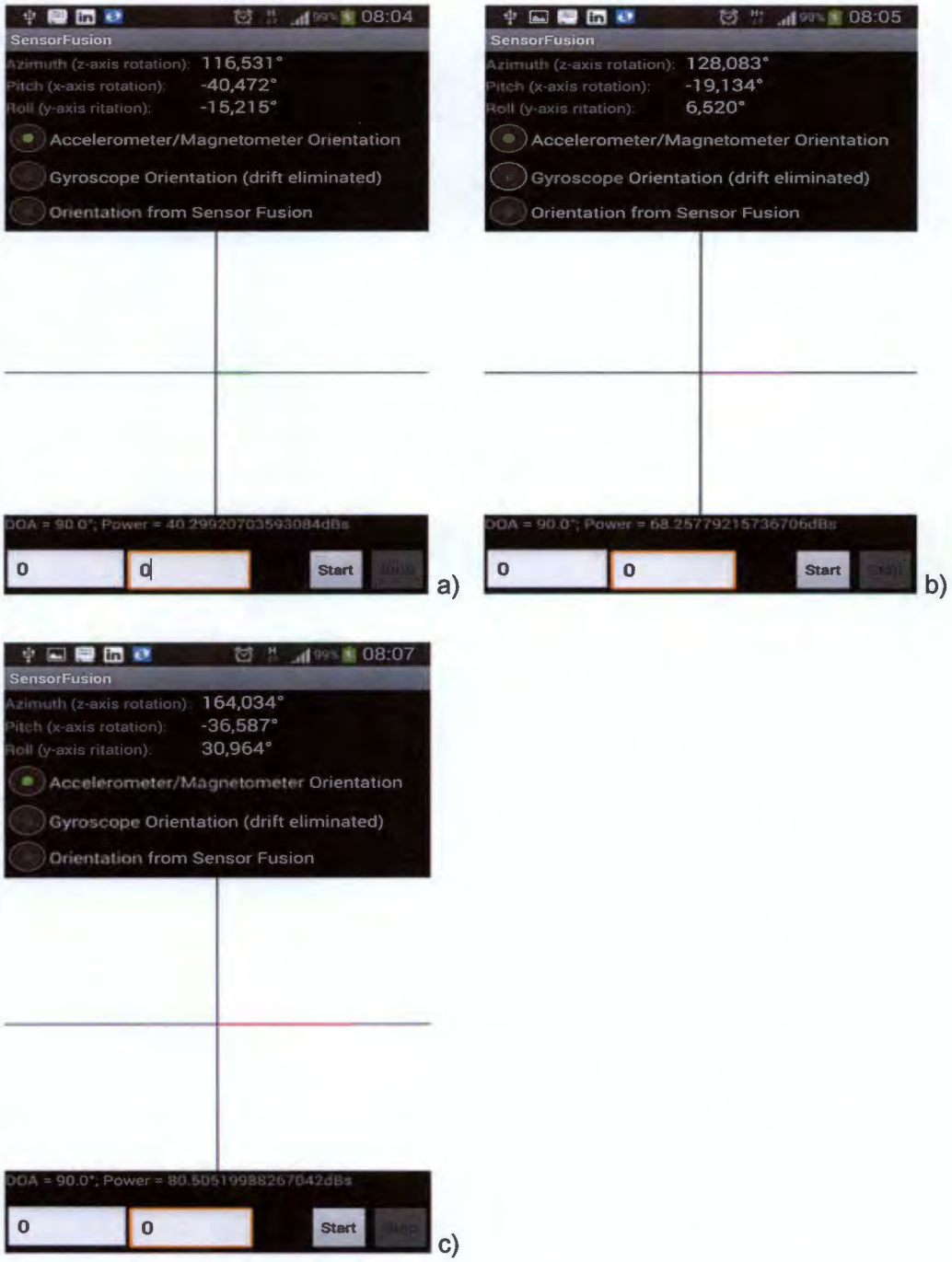


Figure 5.3.2-3 Display result for a) region A, b) region B, and c) region C at DOA=90°

The fourth set of results illustrates the system's display when a sound source is simulated to be located at 62°.

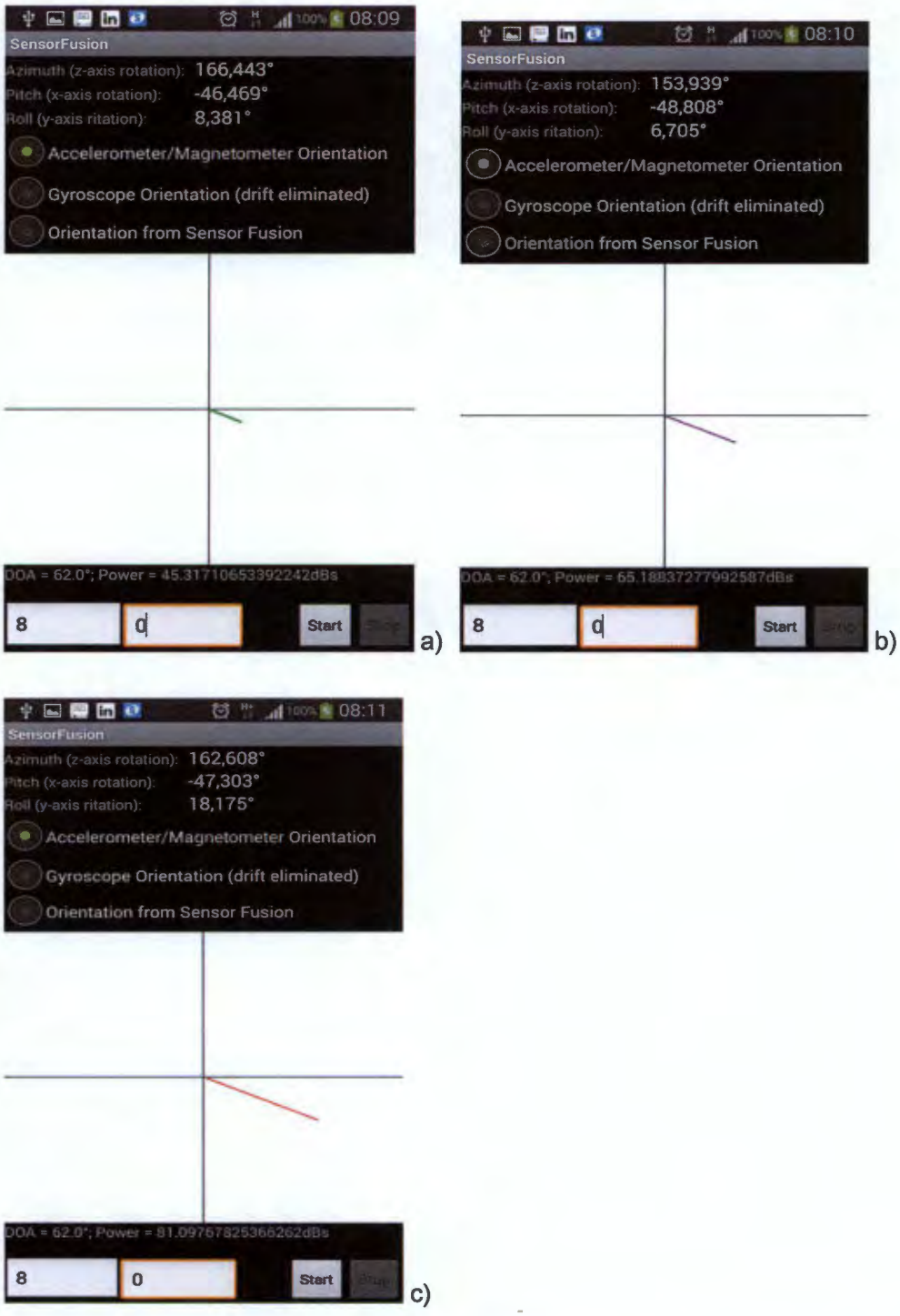


Figure 5.3.2-4 Display result for a) region A, b) region B, and c) region C at DOA=62°

The final set of results illustrates the system's display when a sound source is simulated to be located at 10°.

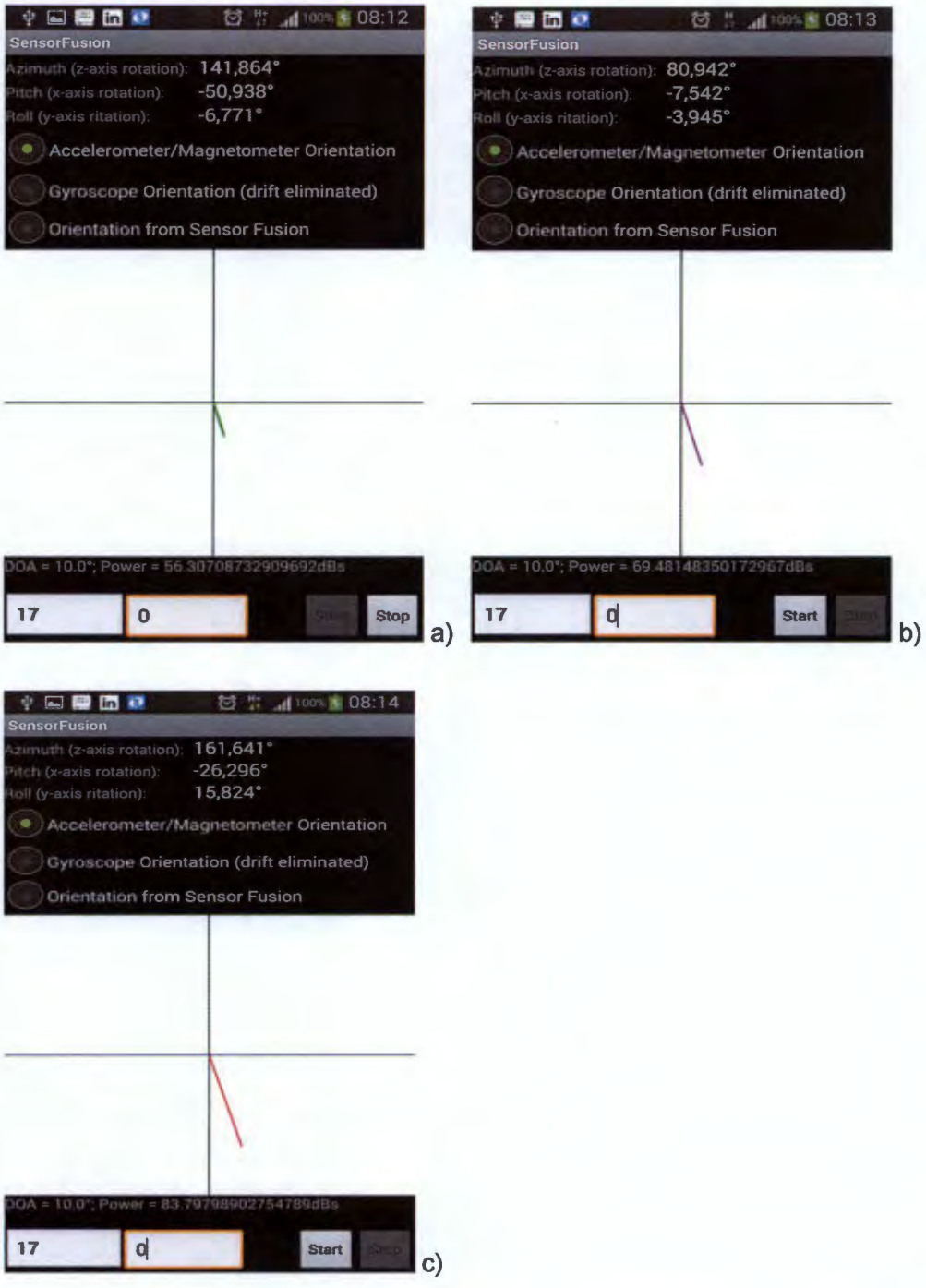


Figure 5.3.2-5 Display result for a) region A, b) region B, and c) region C at DOA=10°

5.3.3 Analysis & conclusion

All tests for this set of experiments yielded expected results. The DOA display line lengths and colours changed proportionally as the sound levels were adjusted.

5.4 Experiment 4: Phone Orientation tests

Phone orientation tests were meant to test orientation data generated by the module designed by Lawitzki[47] . These tests were used to determine which option, from the sensor options below, to choose for the ambiguity resolution module:

- Accelerometer/Magnetometer orientation
- Gyroscope orientation with drift eliminated
- sensor fusion orientation

5.4.1 Set up and expected hypothesis

Since azimuth localisation is the key objective of this project, only rotations around the phone's z-axis are assessed in this test. The procedure was to rotate the phone 360° while logging the orientation data produced by the selected sensor options. Three tests were carried out, each covering a different option from the list stated above. The graphed results are expected to exhibit an 'S' plot symmetric about the line Y = 'start point'.

5.4.2 Experiment results

Three graphs were produced representing results from each sensor option

The first figure shows results when the option of Accelerometer/Magnetometer orientation is selected.

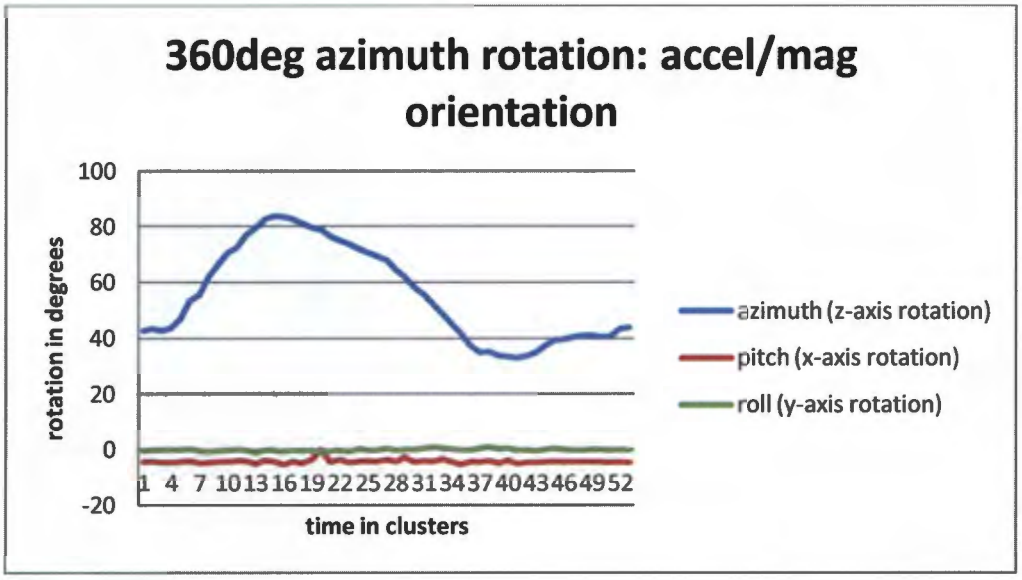


Figure 5.4.2-1 Aaccelerometer/Magnetometer orientation

The data appears to form an 'S' shape in the graph above but it is not symmetrical about the line y = 43, which is the start point of rotation for this test.

The second figure shows results in which gyroscope orientation data is used.

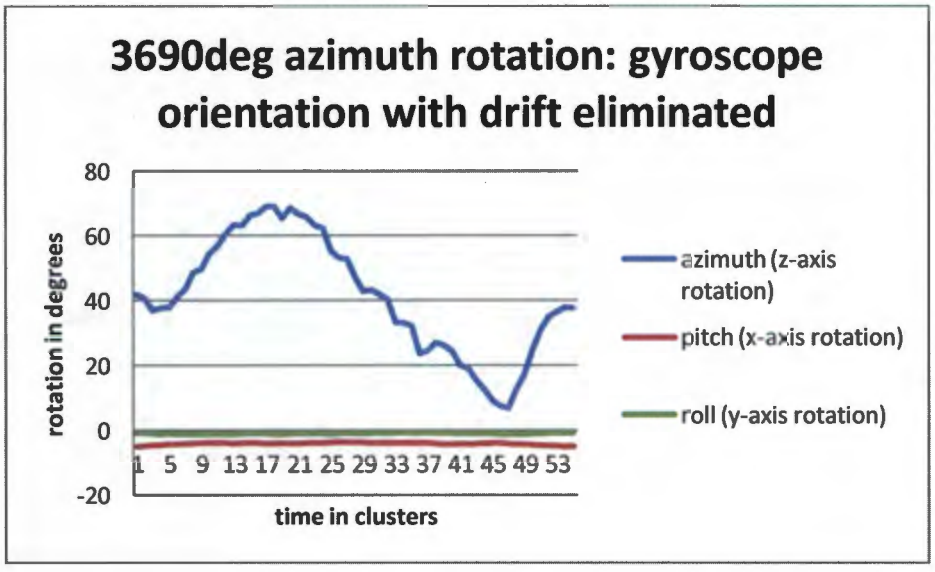


Figure 5.4.2-2 Gyroscope orientation with drift eliminated

The data from these results maps out a fairly symmetrical 'S' shape about the line y= 43. The rotation data does appear to return back to the starting value of 43

The last case of sensor fusion produces similar results to those of the gyroscope option.

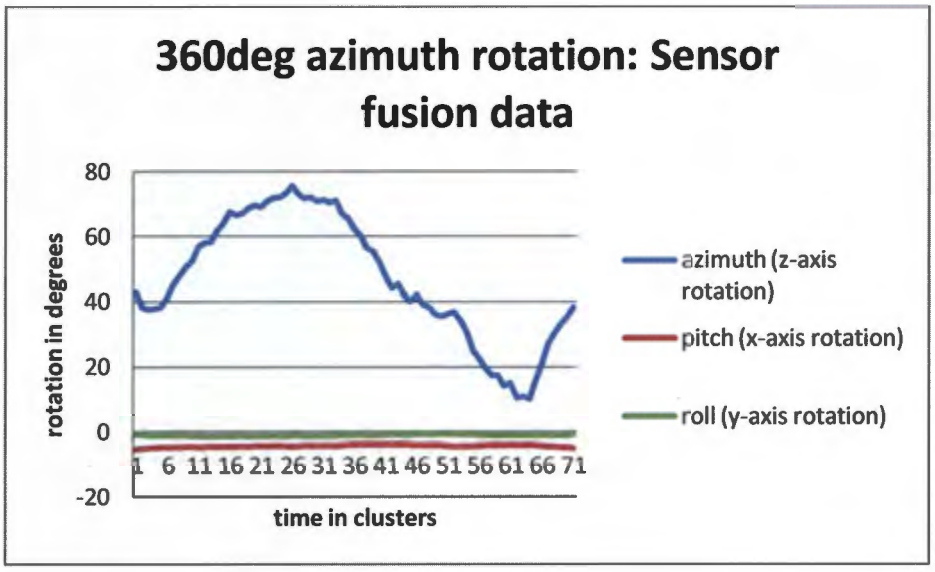


Figure 5.4.2-3 sensor fusion orientation

5.4.3 Analysis & conclusion

Accelerometer/Magnetometer orientation data appears to have non symmetry, indicating that for one of the two 180° sections, the rotation would have larger or smaller rotations as compared to the other. This error would most likely compound itself into wrong ambiguity resolution when a large rotation is made between the two sections.

Both gyroscope orientation data and sensor fusion orientation results appear as expected with data forming a symmetrical 'S' shape about their axis of rotation. The sensor fusion orientation should hence be chosen due to its balanced benefits as explained in [47].

5.5 Experiment 5: Determining optimum dT value for stochastic Histogram algorithm.

The main purpose of this set of experiments was to establish which value of the parameter 'dT' yielded the highest rate of DOA estimation accuracies. dT is the parameter that represents the number of time frames over which a histogram poll is made to estimate the DOA of the most active source, in a close semblance to real time. This parameter is crucial for mapping real time active sound sources into suitable visual perception, bearing in mind that the human audio response is faster than the visual response.

The test values of dT were arrived at by first setting a maximum limit that would throughput at least one DOA estimation response every 1second. The maximum value was calculated to be 43 and was subsequently halved while rounding off to the nearest prime number. The minimum value was set to 1, which is equivalent to taking the DOA estimate from GCC-PHAT without the histogram method. dT values were hence selected as 43, 23, 11, 5, and 1.

Taking note that each time frame is equivalent to 23ms worth of sound as derived in section 4.4, the table below relates how many clusters would be needed for each dT value in order to make up 5 seconds worth of DOA estimations.

Table 5.5-1 Number of clusters needed to analyze 5 seconds worth of sound data per dT setting.

dT	Number of clusters
1	217
5	44
11	20
23	10
43	5

5.5.1 *Set up and expected hypothesis*

A 5mX4m conference room was selected as the venue for these tests. The phone was placed at a fixed position in the room, with freedom of rotation around its z-axis. A tripod rotation mount shown in fig5.1.1-1 was used to provide 360° rotation about the phone's z-axis

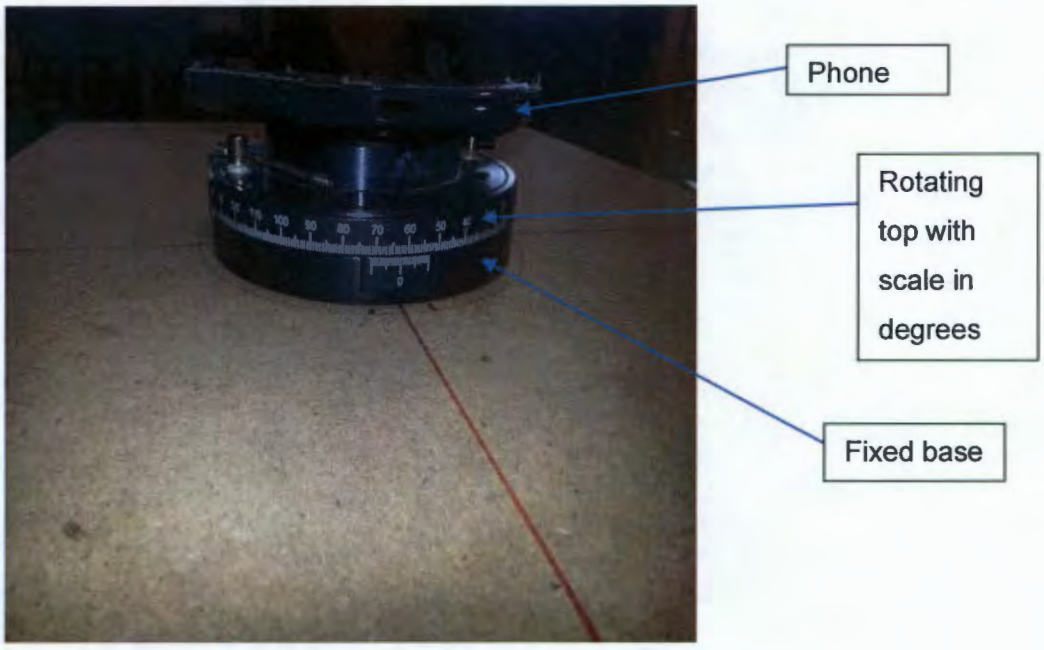


Figure 5.5.1-1 360° rotation mount used for direction alignment

The following procedure was used to run the experiment:

1. Set the sound source at a test distance {0.5m, 1m, or 2m} away from the phone.
2. take baseline noise readings without any sound activity from the main sound source for all test angles {170°, 130°, 90°, 50°, 10° }
3. Activate and calibrate sound source levels to 60dB. Set sound source to starting point.
4. Rotate the phone to the desired angle: {170°, 130°, 90°, 50°, or 10° }
5. Vary 'number of frames' parameter dT to either 1, 5, 11, 23, or 43.
6. Take sound source localisation measurements for a minimum duration of 5 seconds and repeat from step 5 until all dT values are exhausted
7. Repeat from step 4 until all test angles are utilised

8. Alter source distance in step 1 and repeat all steps. (source distance = 0.5m, 1m, 2m)

The rotating mount in fig 5.5.1-1 was used in order to minimise set-up times and errors that would have resulted if the sound source had to be re-positioned at different angles indicated in step 4 of the procedure. The phone was rotated around the z-axis as indicated from fig 5.5.1-2 to 6

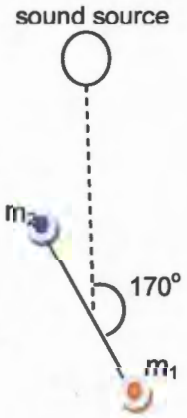


Figure 5.5.1-2 DOA = 170°

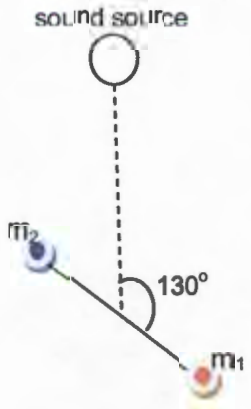


Figure 5.5.1-3 DOA = 130°

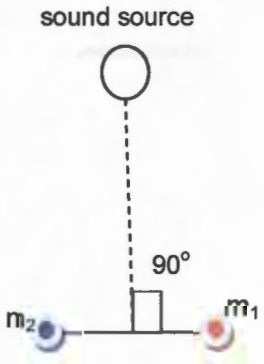


Figure 5.5.1-4 DOA = 90°

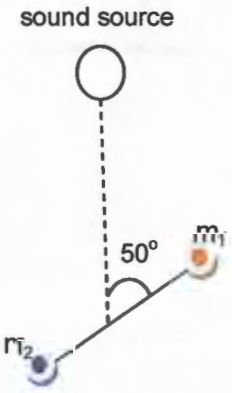


Figure 5.5.1-5 DOA = 50°

Figure 5.5.1-6 DOA = 10°

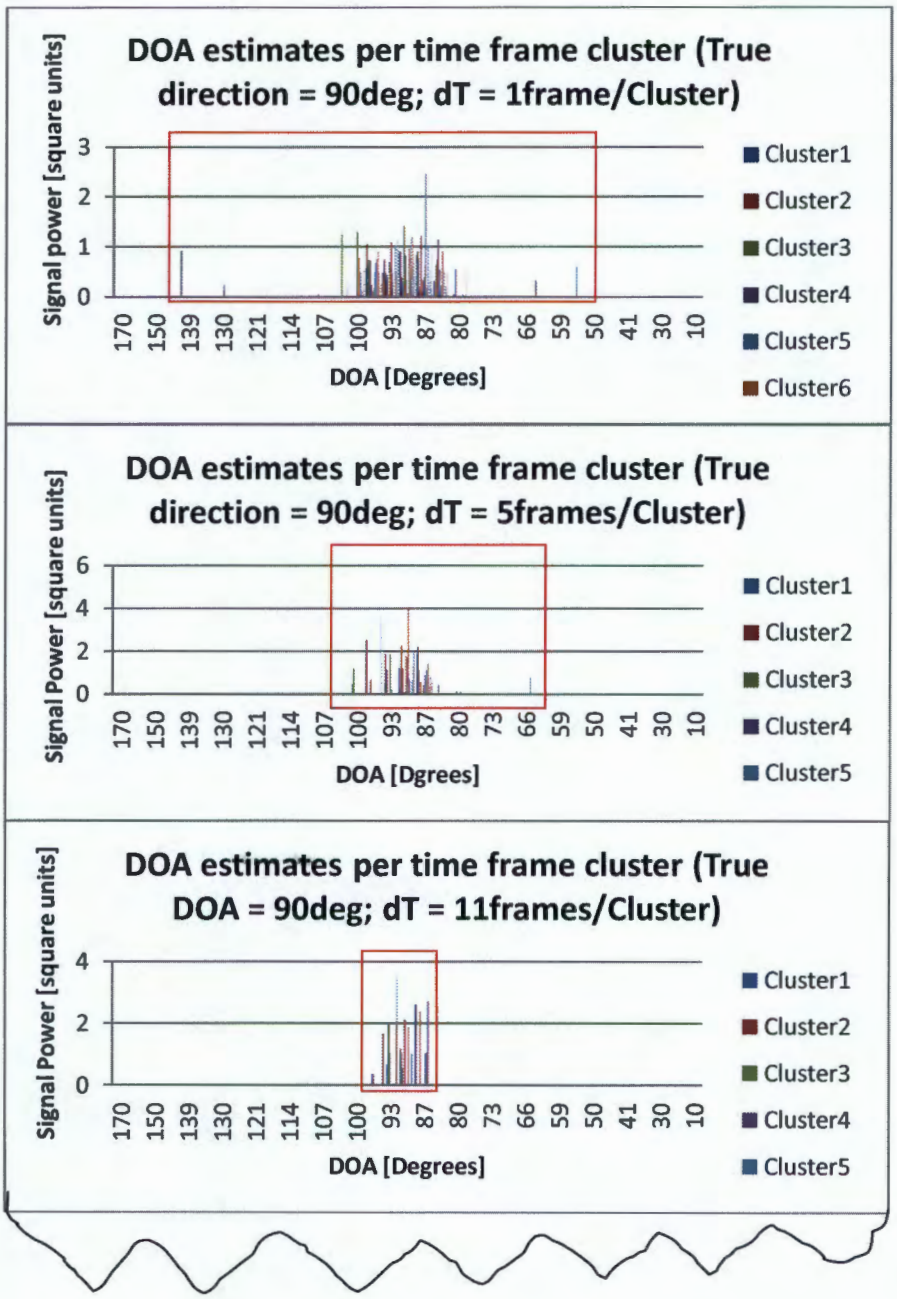
For testing purposes, the system was designed to write a CSV log file indicating the displayed DOAs per time-frame cluster along with the average power levels received by each microphone and the average processing time for each time frame in the cluster. In addition, the log file also includes an array of the power accumulated in each DOA bin from the point the system is started up until it is stopped.

The type of sound used for these experiments was a pop song by Keri Hilson titled "Turn me on". The song was selected due to its diverse composition of sounds.

5.5.2 Experiment results

Each row in the log file is a record made after a cluster of time frames has been processed to yield a DOA to be displayed to the user. As mentioned before, the cluster size is governed by the parameter dT, and the number of clusters needed to analyse 5 seconds worth of data are tabulated in table 5.5-1. The first 35 columns represent DOA bins.

Fig 5.5.2-1 is a collage of graphs showing the estimated DOA by each cluster for different dT values when the sound source is 90°, 0.5m ahead of the sensor axis.



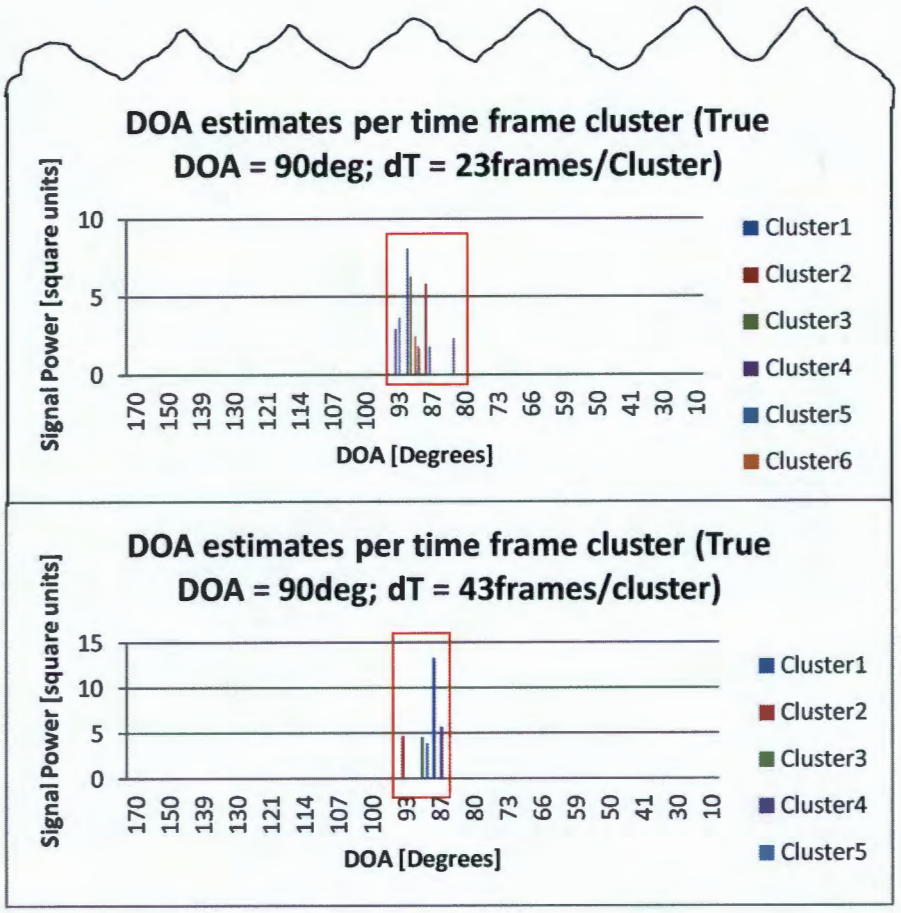


Figure 5.5.2-1 collage of DOA estimations obtained from each cluster for a range of dT values when sound source is located 90deg, 0.5m ahead of the phone.

The red windows are a visual aid used to highlight the breadth by which the DOA estimates are spread across various bins, for different values of dT. In order to quantitatively analyse the accuracy of the system, one needs to establish the percentage of accurate DOA estimates for each dT setting across the different test distances and angles. Two categories of accuracy were utilised to analyse the system in order to choose the optimum value of dT:

- **Pin-point accuracy**, which for this project was defined as a measure of the percentage of clusters from a sample space, that exhibit the expected DOA.
- **$\pm$ 1 sample accuracy**, which was defined as a measure of the percentage of clusters from a sample space, that exhibit the expected DOA to within $\pm$ one sample tolerance. Depending on the region of DOAs, $\pm$ 1 sample

translates to different sized DOA beams as calculated from resolutions in table 4.3-1.

PIN-POINT ACCURACY

The next three graphs show the pin-point accuracy rates of DOA estimation for each of the selected test distances, test angles and dT values.

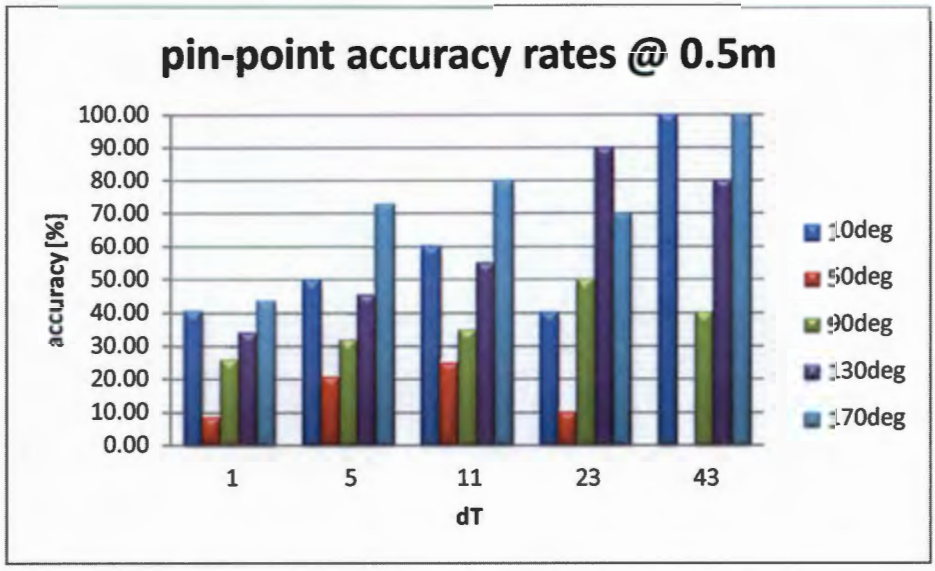


Figure 5.5.2-2 pin-point accuracy results for source placed at 0.5m away from phone

The clustered column graph clusters DOA results in dT bins along the x-axis. In the graph above, the '50deg' series is seen to have the lowest accuracy rates for all dT values, whereas the '170deg' series generally has the highest accuracy rates amongst the various dT values.

Fig 5.5.2-3represents the results of pin point accuracies when the sound source is placed 1m away from the phone

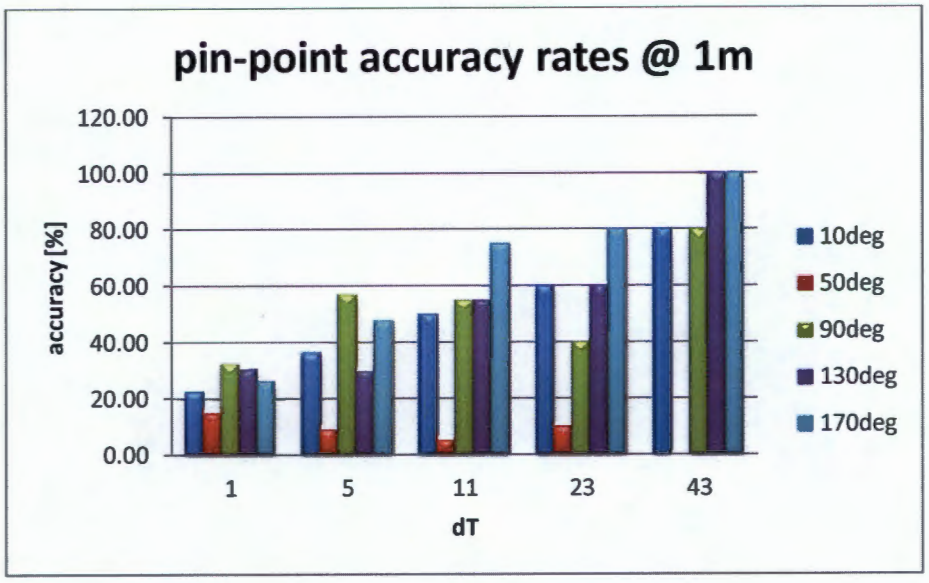


Figure 5.5.2-3 pin-point accuracy results for source placed at 1m away from phone

As for the 0.5m results, the results in the graph above indicate that the lowest accuracy rates were achieved when the sound source was positioned at a 50deg angle ahead of the microphone axis.

The next graph is for results where the sound source is placed at 2m away from the phone.

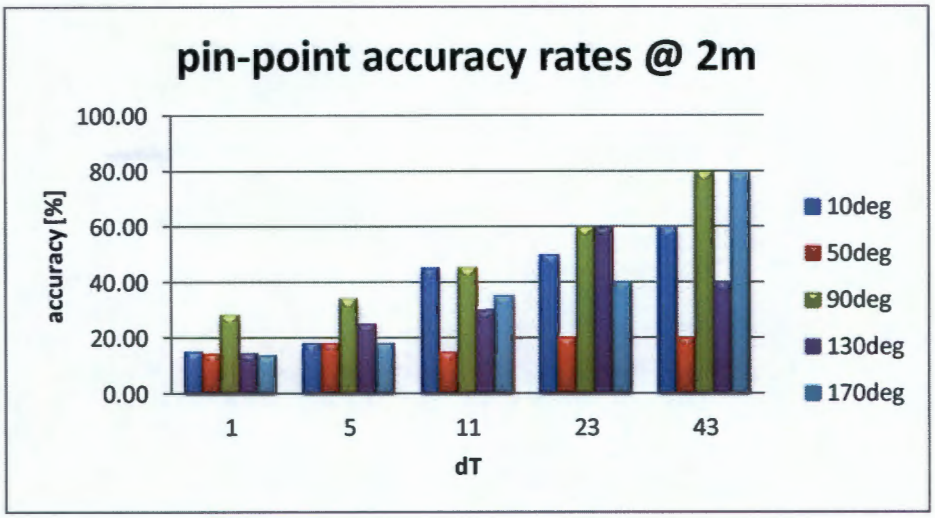


Figure 5.5.2-4 pin-point accuracy results for source placed at 2m away from phone

From a quick glance, it is noted that for all 3 graphs in fig 5.5.2-2 to4, it is evident that the accuracy rates improve for most DOAs as the value of dT is increased. The table below shows the pin-point accuracies averaged across the three distance settings.

Table 5.5.2-1 average pin-point accuracy rates for each dT and DOA across the 3 distance categories

Average Pin-point accuracy rates		dT				
		1	5	11	23	43
DOA	10deg	26.11	34.85	51.67	50.00	80.00
	50deg	12.44	15.91	15.00	13.33	6.67
	90deg	28.73	40.91	45.00	50.00	66.67
	130deg	26.27	33.33	46.67	70.00	73.33
	170deg	27.96	46.21	63.33	63.33	93.33

The data from table 5.5.2-1 is plotted in the graph below to provide a visual representation for further analysis.

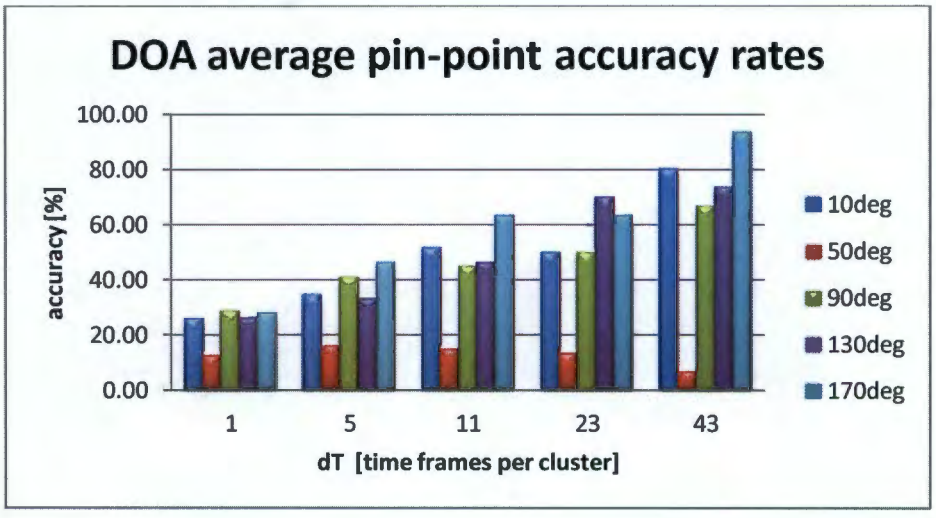


Figure 5.5.2-5 average pin-point accuracy rates for each dT and DOA across the 3 distance categories

Preliminary analysis from the data in table 5.5.2-1 indicates that the highest pin-point accuracy was achieved for test cases where the sound source was at 170° and the dT value was set to 43.

To compare the effect of dT on accuracies amongst the different distances (0.5m, 1m, 2m), an average is taken for each dT column and graphed for each distance in the figure below.

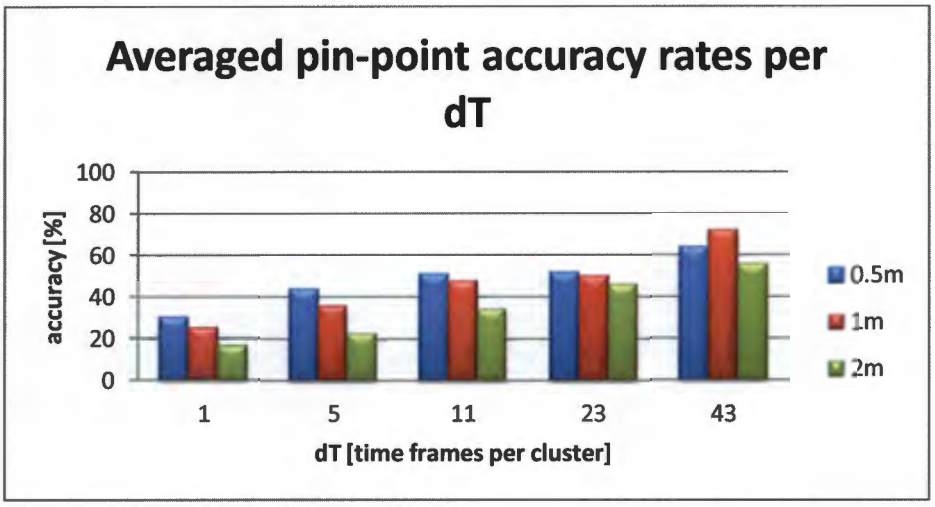


Figure 5.5.2-6 pin-point accuracies averaged over all DOAs for each dT setting at different test distances {0.5m, 1m, 2m}

The graph above shows that for lower dT values, the system had the highest pin-point accuracy when the sound source was 0.5m away from the phone. The graph also shows that the highest accuracy rates were achieved when dT was set to 43 for all distances.

To determine the overall accuracies for all the dT values used for these tests, the averages used to generate fig 5.5.2-6 are averaged to establish a single accuracy value for each dT setting. The results are graphed in the figure below

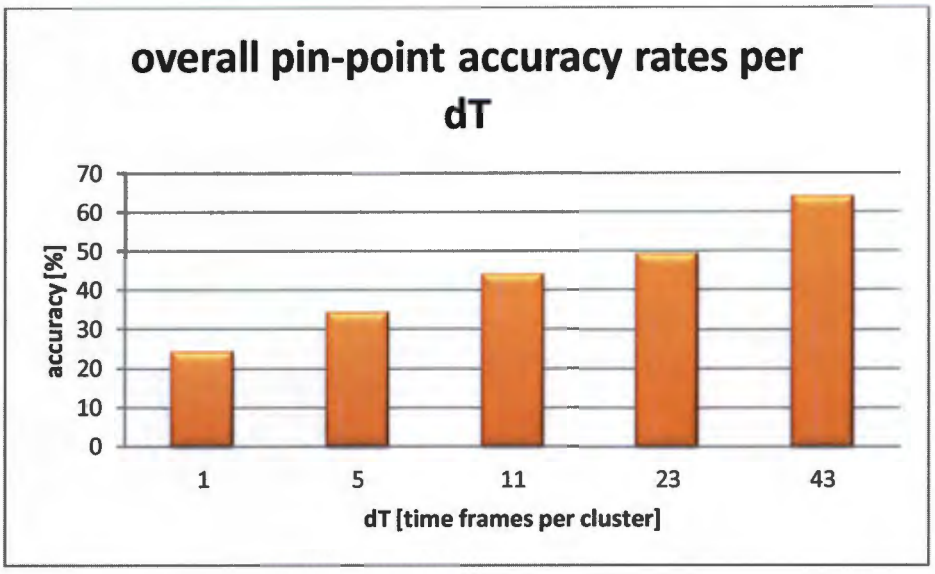


Figure 5.5.2-7 overall system pin-point accuracy rates across all test scenarios

From the figure above, the highest overall pin-point accuracy was attained from dT = 43. An almost linear trend is observed in the increase of accuracy rates as the dT values are increased. A detailed analysis of the results is made in section 5.5.3

± 1 SAMPLE ACCURACY

The pin-point results yielded an overall average of 64% when dT was set to 43. However, in some instance, the end user may be comfortable with having a general DOA range rather than pin-point results. This warranted the need to repeat analysis similar to that of the pin-point results, but with a tolerance of ± 1 sample.

The graph below is generated from the same data as fig 5.5.2-2 and represents a scenario in which the sound source is approximated to within ± 1 sample of the pin-point DOA bin. For example, if the sound source is expected to be estimated within the 18th DOA bin (which represents DOA=90deg), this analysis would rather expect the sound source to be estimated in either the 17th, 18th, or 19th bin.

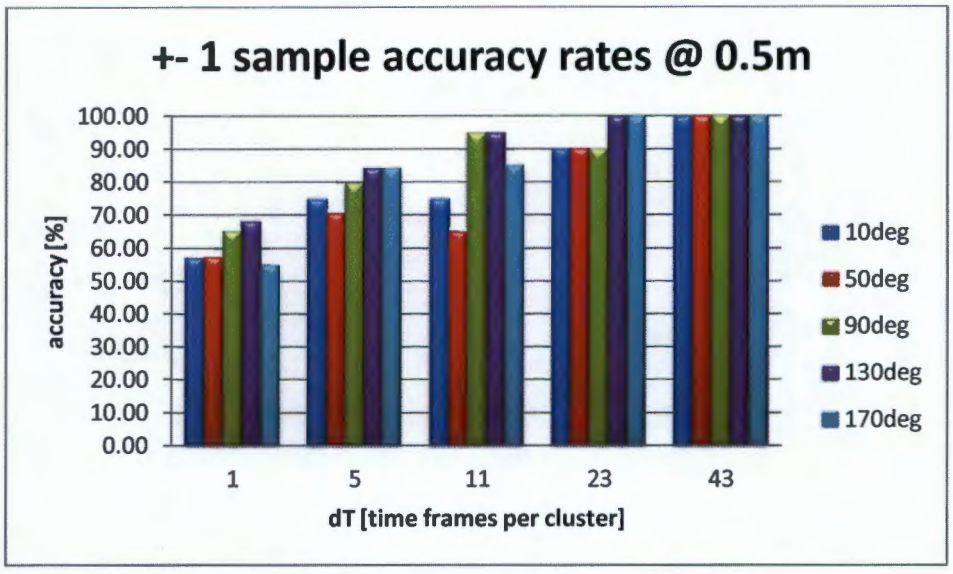


Figure 5.5.2-8 ± 1 sample accuracy results for source placed at 0.5m away from phone

The results in the figure above bear an evident improvement over the results in fig 5.5.2-2. It is worth noting the tremendous improvement for accuracies of series '50deg'.

Fig 5.5.2-9 below is the counterpart to fig 5.5.2-3 and bears similar improvements as those noted for fig 5.5.2-8.

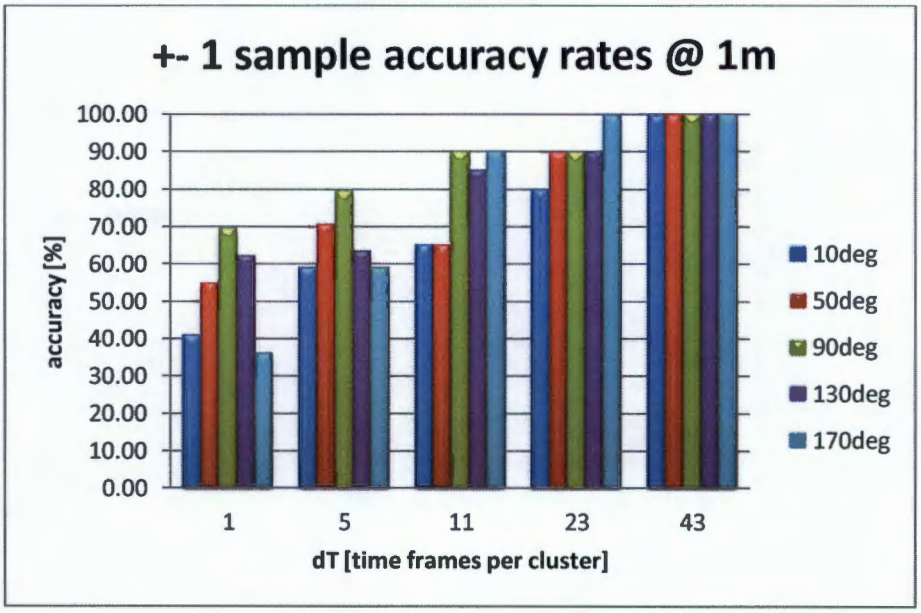


Figure 5.5.2-9 ± 1 sample accuracy results for source placed at 1m away from phone

The accuracies are observed to improve somewhat linearly for most DOAs as the dT value is increased.

For the case of the tests were the source was placed 2m away from the phone, the graph below highlights improvement over its counterpart in fig 5.5.2-4, but exhibits less improvement compared to the other two graphs of 0.5m, and 1m in the same category.

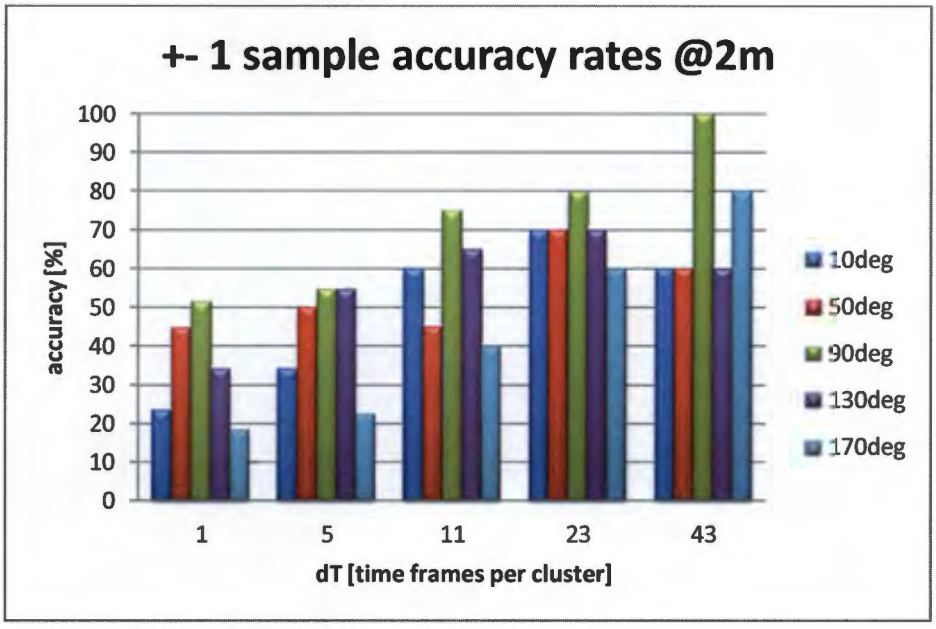


Figure 5.5.2-10 ± 1 sample accuracy results for source placed at 2m away from phone

DOA = 90deg exhibits the highest accuracies across all dT settings for the graph above and maintains an almost linear increase in accuracy as the dT value is increased.

As in the case of the pin-point results, the table below was generated to analyse the average accuracy for each DOA per dT averaged across the different test distances.

Table 5.5.2-2 average ± 1 sample accuracy rates for each dT and DOA across the 3 distance categories

Average percentage Pin-point accuracy rates		dT				
		1	**5**	**11**	**23**	**43**
DOA	**10deg**	40.40	56.06	66.67	80.00	86.67
	50deg	52.23	63.64	58.33	83.33	86.67
	90deg	62.06	71.21	86.67	86.67	100.00
	130deg	54.84	67.42	81.67	86.67	86.67
	170deg	36.41	55.30	71.67	86.67	93.33

The resulting graphed averages from the table above are shown in the figure below.

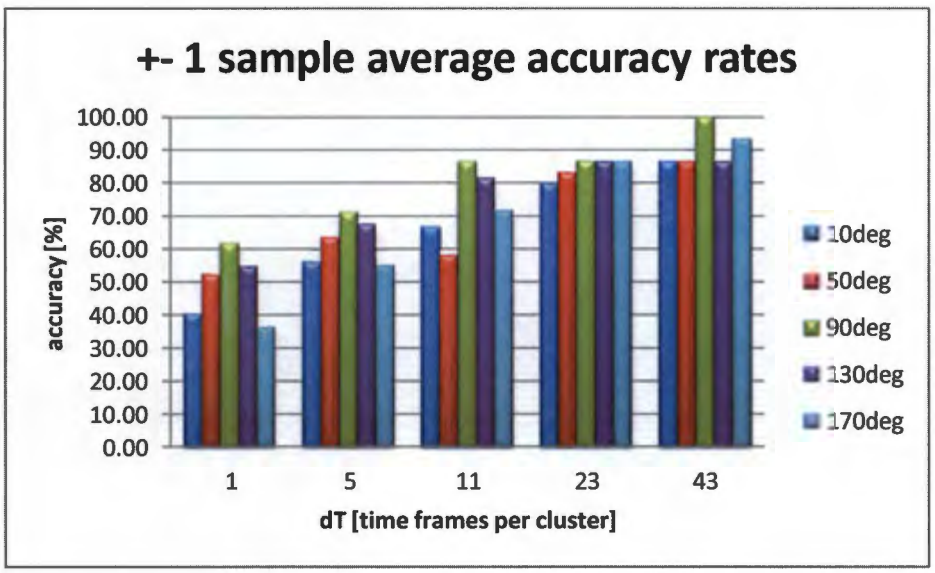


Figure 5.5.2-11 average ± 1 sample accuracy rates for each dT and DOA across the 3 distance categories

In this case, the highest average accuracy was achieved by DOA = 90deg when dT was set at 43.

The graph below shows data averaged per dT column for all three test distances.

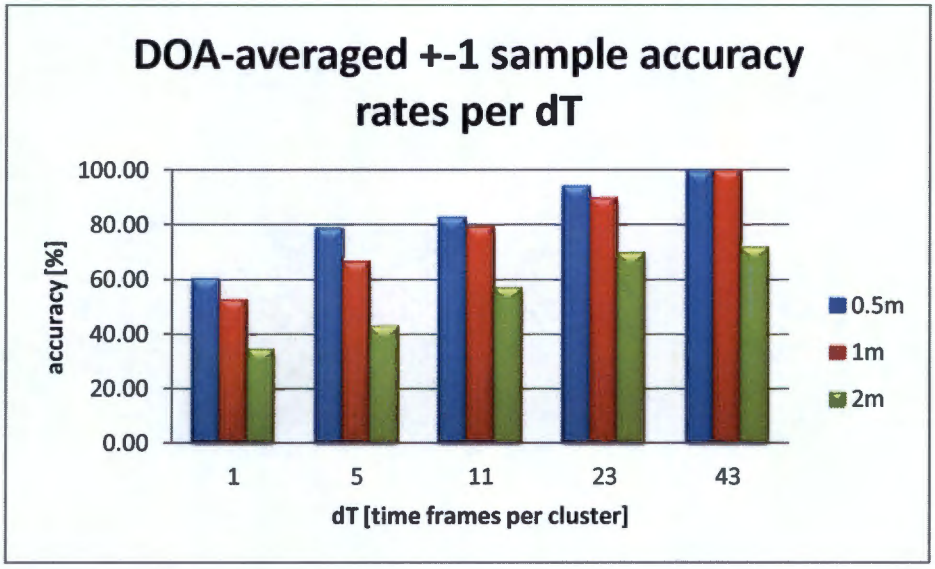


Figure 5.5.2-12 ± 1 sample accuracies averaged over all DOAs for each dT setting at different test distances {0.5m, 1m, 2m}

The resulting data from the graph above is further averaged for each dT bin to establish a single accuracy value for each dT setting. The results are shown the figure below.

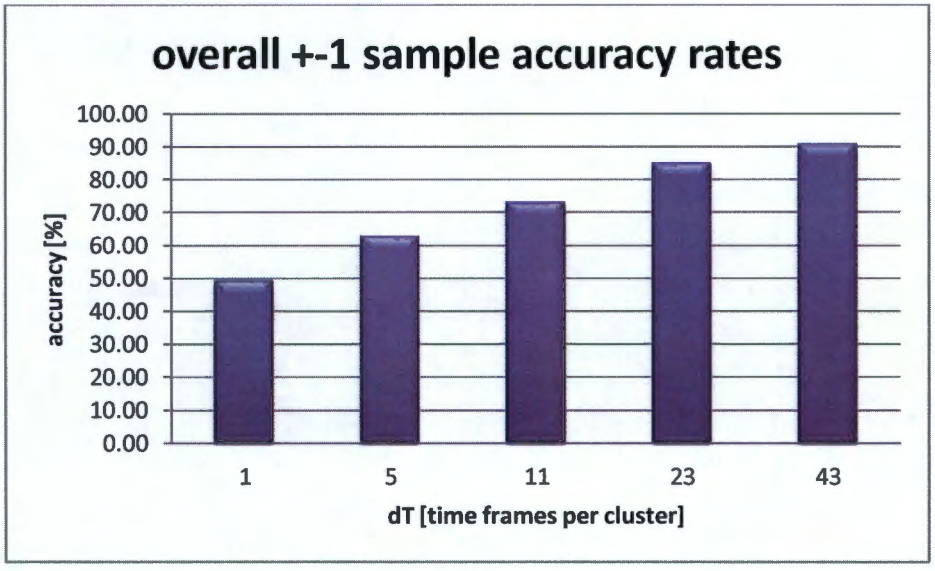


Figure 5.5.2-13 overall system ± 1 sample accuracy rates across all test scenarios

The trend of the graph above is similar to that of the graph in fig 5.5.2-7, however, there is an overall improvement in accuracy rates over the latter.

CUMULATIVE POWER HISTOGRAMS

The cumulative power distribution graphs were a resultant of plotting the cumulative power histograms that were generated by the system through accumulating the signal power estimated for each DOA bin throughout the system's run time.

The following graphs were results from the test cases where the sound source was placed 0.5m away from the phone at a 90deg DOA. Similar graphs from 1m and 2m tests are listed in appendix B.

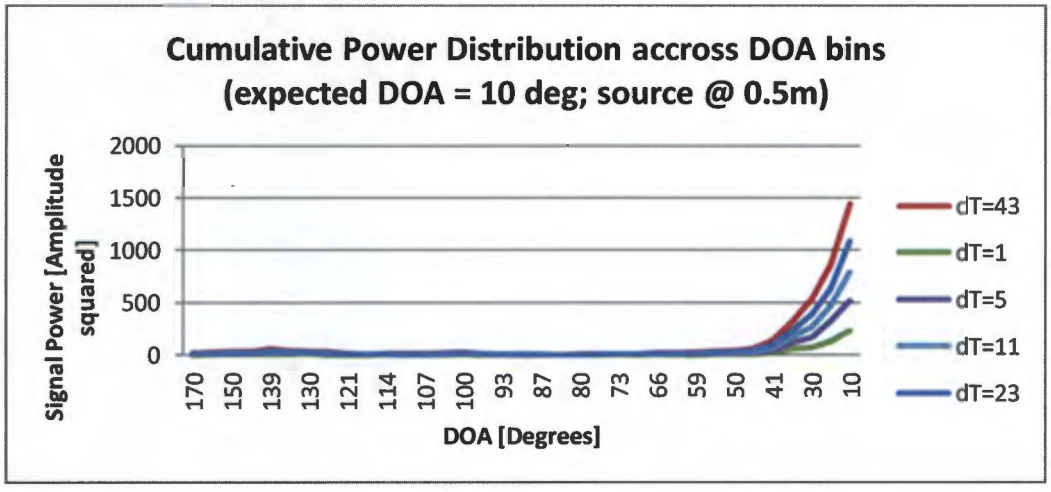


Figure 5.5.2-14 stacked accumulated power histograms peaking at DOA=10deg

All histograms peaked at the expected DOA in the figure above. The power level appears to increase as the dT is increased.

For the case of expected DOA=50deg, the histograms were all found to peak at DOA=46deg as shown below.

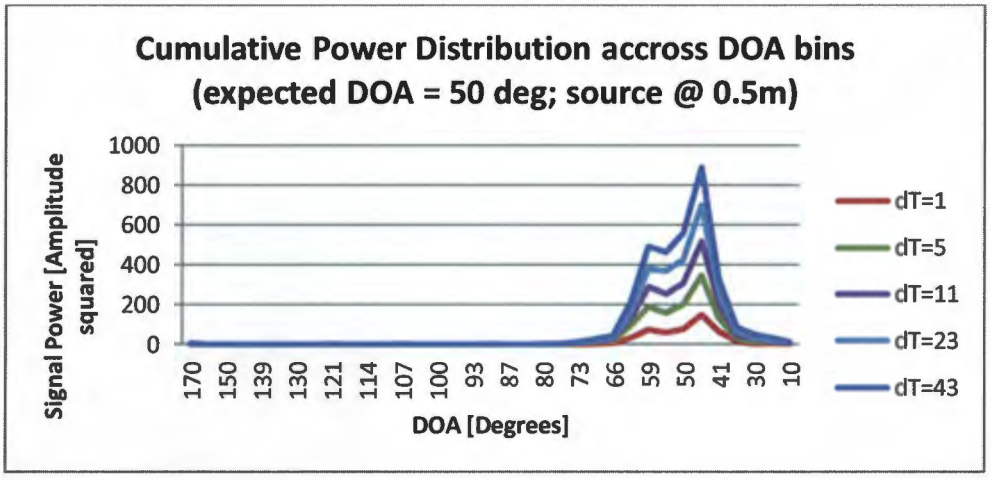


Figure 5.5.2-15 stacked accumulated power histogram peaking at DOA=46deg

This result persisted across measurements taken at 1m and 2m, and is discussed further in the analysis.

For the next three graphs, the stacked histograms are seen to peak at the respective expected DOAs.

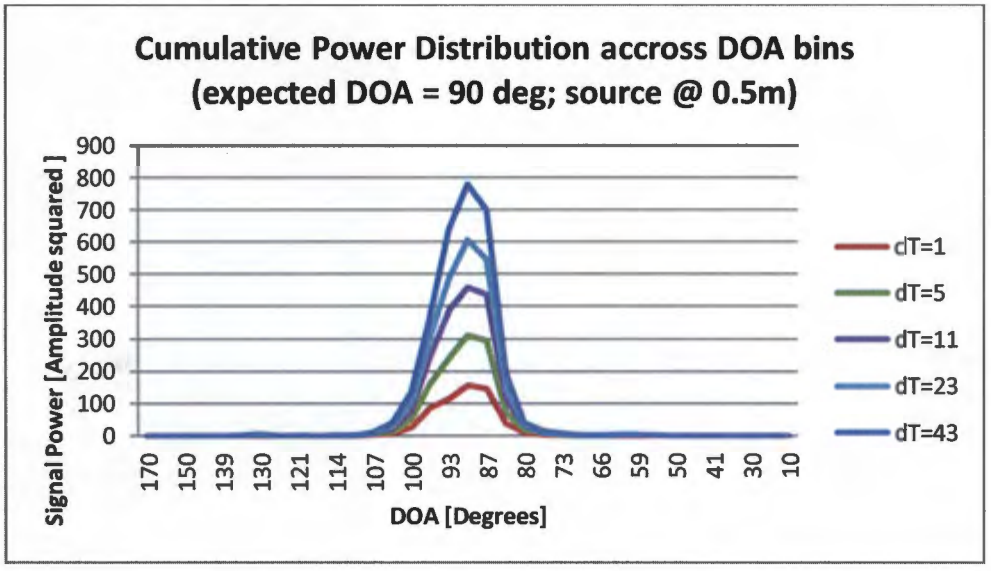


Figure 5.5.2-16 stacked accumulated power histograms peaking at DOA=90deg

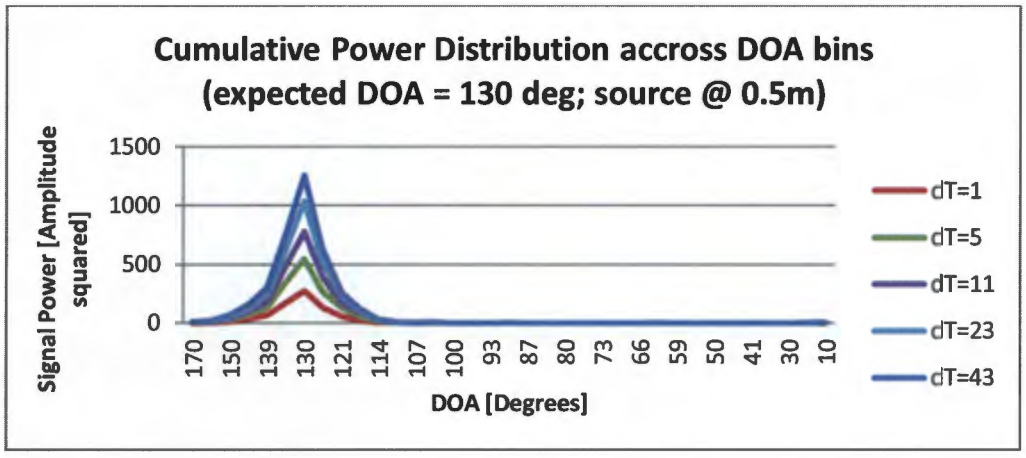


Figure 5.5.2-17 stacked accumulated power histograms peaking at DOA=130deg

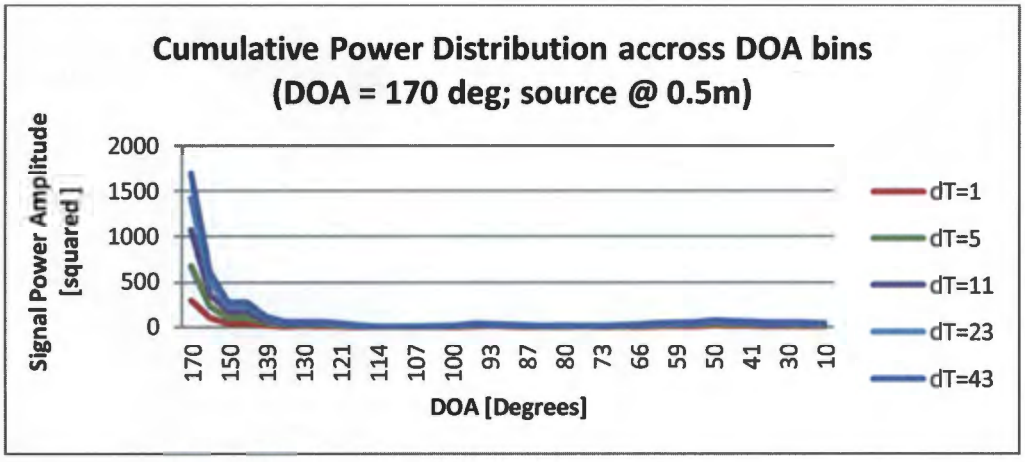


Figure 5.5.2-18 stacked accumulated power histograms peaking at DOA=170deg

A trend in which accumulated signal power increases as dT increases, is observed for all histograms from fig 5.5.2-14 to fig 5.5.2-18.

5.5.3 Analysis & conclusion

Two approaches were used for determining which value of dT produced the highest rate of accuracy:

- Pin-point accuracy analysis, and;
- ± 1 sample accuracy analysis.

For pin-point accuracies, the general trend across all distance test cases was that the accuracy seemed to increase linearly as the value of dT was increased.

Fig 5.5.2-5 was a result of calculating dT average accuracies as a function distance. This led to a matrix of DOA by dT pin-point accuracies. Calculating dT accuracies as a function of DOA led to fig 5.5.2-6. Both of these graphs indicated an increase in accuracy rates as dT was increased, with the maximum results achieved with dT set to 43. In order to simplify quantitative comparison amongst the five dT values, the dT pin-point accuracy rates were calculated as function of both DOA, and distance of measurement. This resulted in fig 5.5.2-7 which showed the highest overall pin-point accuracy to have been 64% when dT is set to 43.

A similar analysis is performed for scenarios when the end user may be interested in a broader DOA rather than pin-point accuracy. In this case, the analysis is repeated exactly the same way as for pin-point accuracies with difference being that a ± 1 sample tolerance in accuracy is used. The resulting overall accuracy for such a scenario is contrasted against the overall accuracies of pin-point results in the figure below.

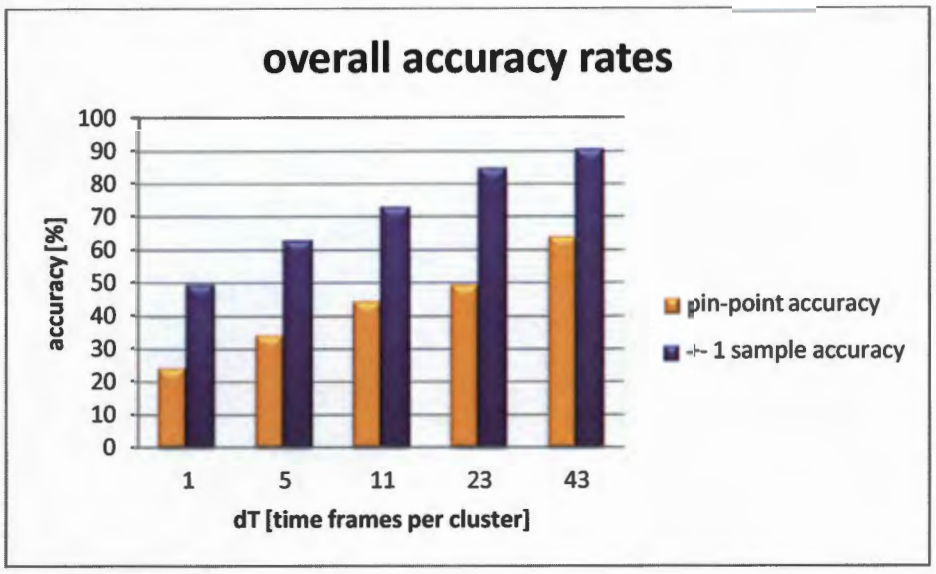


Figure 5.5.3-1 contrast between overall accuracy rates from pin-point and ± 1 sample tolerance analysis

In both approaches, the trend shows a linear increase in accuracy rates as the value of dT is increased. The pin-point accuracy approach yields a maximum of 64% accuracy at dT= 43. The corresponding maximum ± 1 sample tolerance accuracy is 90.67% when dT = 43. One can therefore conclude that the highest accuracy rates for the system are achieved when dT is set to dT = 43.

The results from the cumulative power histograms consistently exhibited Gaussian-like distributions centred on the estimated DOA bins. This led to the hypothesis that the system could be used to perform source counting and localisation of multiple sound sources.

Analysis of cumulative power histograms for test cases in which the expected DOA was 50deg, showed that across all test distances, the sound source was rather estimated at 46deg, a -1 sample away from the expected DOA. The precision of this anomaly indicated that it was most likely due to multipath propagation effects. It further helped explain why pin-point accuracy rates were much lower for DOA=50deg test cases than any other DOAs in the pin-point analysis.

5.6 Experiment 6: Source counting tests

5.6.1 Set up and expected hypothesis

Three sound sources were used to verify the assumption of the systems' ability to count present sound sources while estimating their Direction of arrivals using the proposed histogram algorithm of signal power accumulation across the GCC-PHAT estimated DOA bins.

The procedure for this experiment is as follows:

1. Set 3 sound sources 1 metre away from the test phone at 135°, 90°, and 45°
2. Calibrate the sound level for each sound source to emulate 60dB reception at the phone.
3. Set dT to 43.
4. activate all sources and take measurements
5. Swap positions of sound sources and repeat from step 3

Three sound sources of different types were used in order to cater for a more realistic situation. In addition to the pop music (M) that was used for the single source experiments, a 2Hz pulse (P) was used as the second source and a female speech (S) signal used as a third. The permutations for the source positions were PSM, SMP, and SPM.

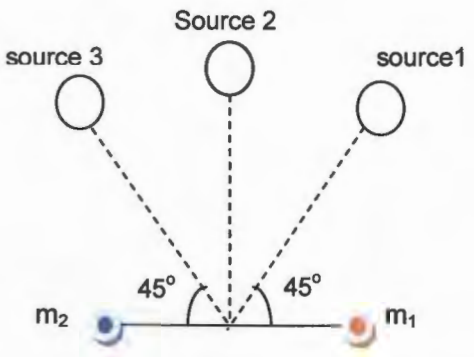


Figure 5.6.1-1 Source placement in relation to mic1 and mic2

5.6.2 *Experiment results*

The results of the experiments are presented in cumulative power distribution graphs showing the raw data together with the smoothed data that is used as a basis for the source counting and separation algorithm. Each graph is followed by a table tabulating the estimated DOAs for both data types (raw and smoothed), and the associated estimation errors compared to the reference expected DOAs. The expected DOAs were 134°, 90°, and 46° due to round off dictated by the discrete sample delays.

The first results are for the test case in which the pulse source was placed 135°; the speech source placed at 90°; the music source at 45°.

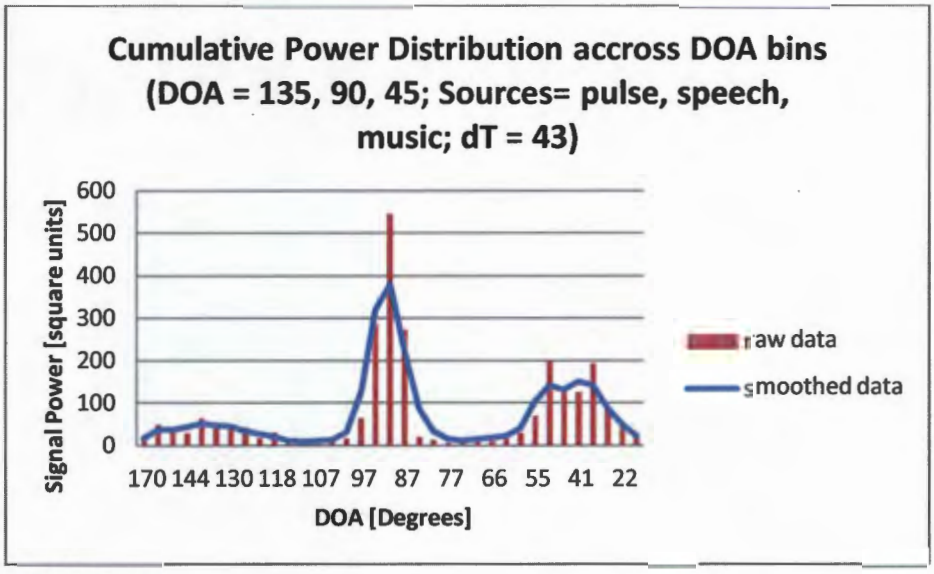


Figure 5.6.2-1 overlaid raw and smoothed data for PSM test with dT=43

Table 5.6.2-1 estimation error tabulation for raw Vs smoothed data PSM tests; dT =43

	Direction of Arrival [Degrees]		
	Pulse	Speech	Music
expected	134	90	46
Raw data	139	90	50
Smoothed data	139	90	41
Raw data error	5	0	4
Smoothed data error	5	0	-5

The results for this test case exhibited highest accuracies for sources placed at 90°. The maximum errors were 5° and -5° at 135° and 45° respectively.

The next set of results is for the speech source was placed 135°; the music source placed at 90°; and the pulse source at 45°.

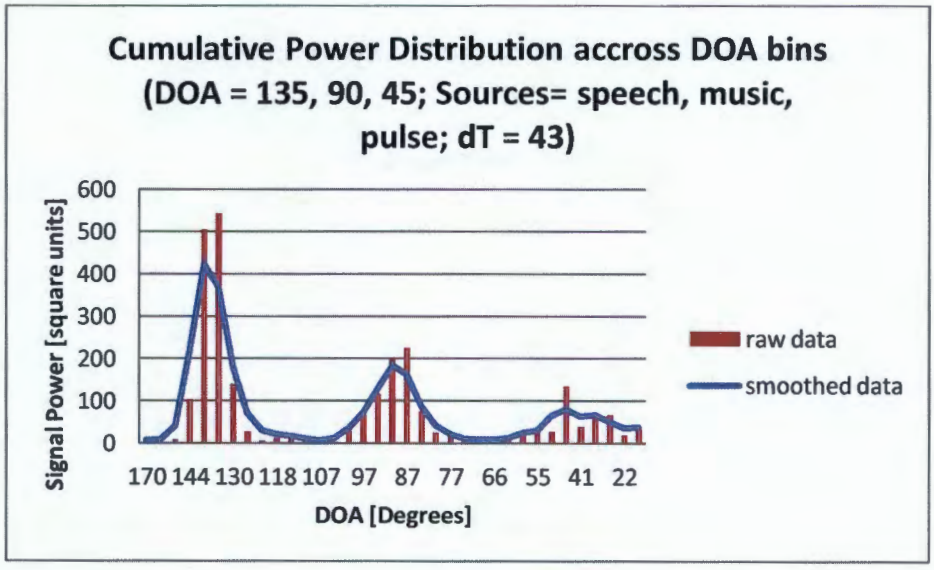


Figure 5.6.2-2 overlaid raw and smoothed data for SMP test with dT=43

Table 5.6.2-2 estimation error tabulation for raw Vs smoothed data SMP tests; dT=43

	Direction of Arrival [Degrees]		
	Speech	Music	Pulse
Expected DOA	134	90	46
Raw data	134	87	46
Smoothed data	139	90	46
Raw data error	0	-3	0
Smoothed data error	5	0	0

In this case, raw data exhibited less error for DOA=135 and 45 as compared to the smoothed data. The maximum error across all DOAs was registered as 5° by smoothed data at DOA=135°. This was most likely due to the smoothed data's peak being shifted by the sharp roll off from the peak created by the speech source.

The final set of results represents the test case in which the speech source was placed 135°; the pulse source placed at 90°; and the music source at 45°.

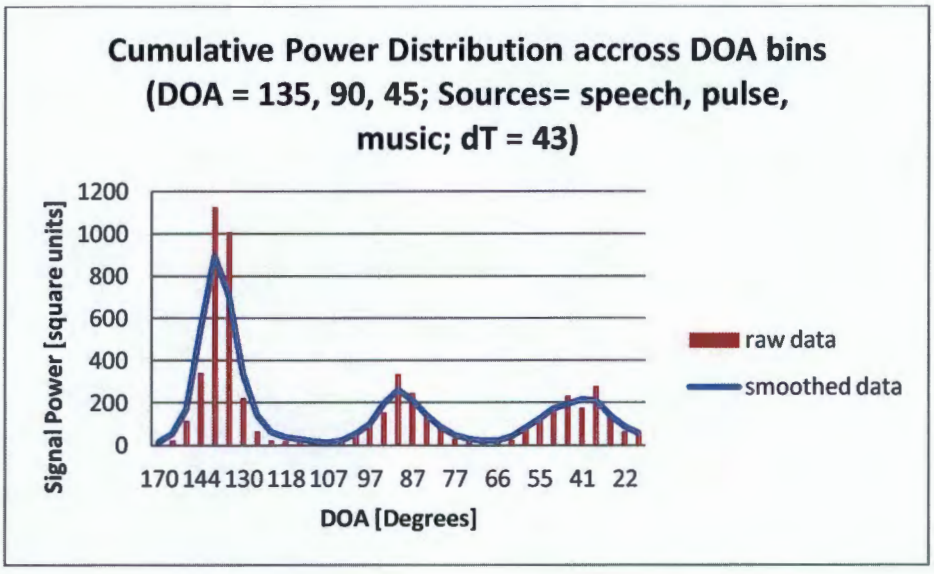


Figure 5.6.2-3 overlaid raw and smoothed data for SPM test with dT=43

Table 5.6.2-3 estimation error tabulation for raw Vs smoothed data SPM tests; dT=43

	Direction of Arrival [Degrees]		
	Speech	Pulse	Music
expected	134	90	46
Raw data	139	90	36
Smoothed data	139	90	41
Raw data error	5	0	-10
Smoothed data error	5	0	-5

DOA =135° and 90° yielded the same errors for both raw and smoothed data. The speech source exhibited the highest and narrowest peaks for all test cases.

5.6.3 Analysis & conclusion

Gaussian-like peaks were exhibited for the raw data in all test cases. The number of peaks exhibited in the data was proportional to the number of sources in the test environment.

The system had been hypothesised from results in experiment 5 to be able to generate Gaussian-like peaks centred at positions relative to the expected DOAs. The peaks exhibited by the raw data were centred at positions proportional to the expected DOAs. Smoothing the data helped eradicate noisy local maxima that created counting ambiguity.

From the graphs and tables in section 5.6.2, the following summarised analysis can be made:

- Speech signal yielded the highest cumulative signal power across all test cases. The peaks associated with the speech signal were narrower and higher than the other sound sources.
- The system was most accurate in localising sound sources placed at 90°, with smoothed data yielding 0% errors for all test cases
- The raw data had the least number of errors in DOA estimations for sound sources at 135°. However, some cases exhibited more than one local maxima in close proximity which made the data ambiguous for source counting. The effect of the local maxima is evident in estimation results for sources located at 45° where maximum errors of $\pm 10°$ were achieved.
- Smoothed data yielded peaks that were much easier to detect computationally, despite having maximum errors of +5° in the 135° test cases, and -5° for the 45° cases. All 90° estimates were 100% accurate.

5.7 Experiment 7: Front-back ambiguity resolution tests

These tests were designed to analyse whether it was possible to resolve front-back ambiguity using phone orientation data.

5.7.1 Set up and expected hypothesis

In order test the ambiguity resolution module, the sensor fusion orientation option should be selected from the phone orientation tracking module. dT should be set to the value recommended in experiment 5. A single sound source should be set fixed at a distance of 0.5m in front of the of the sensor axis.

While the sound source is active and the system has been started, the phone should be moved in pivot motions constrained by 30° oscillations.

Sound sources at the front side of the microphone are expected to be displayed as positive numbers in the log files, whereas sources at the back are expected to be indicated as negative values in the corresponding DOA bins in the log file. This would translate to having positive columns on a column graph for sources at the front side and negative columns for sources at the back side of the sensor axis.

For a source at the front, the experiment is expected to have 100% of all estimated DOAs being positive and in the region 55° to 93°; and for sources at the back 100% of the DOAs to be negative and within the range 87° to 134°

5.7.2 Experiment results

The figure below represents the estimations yielded by the system when the source is in front of the sensor axis.

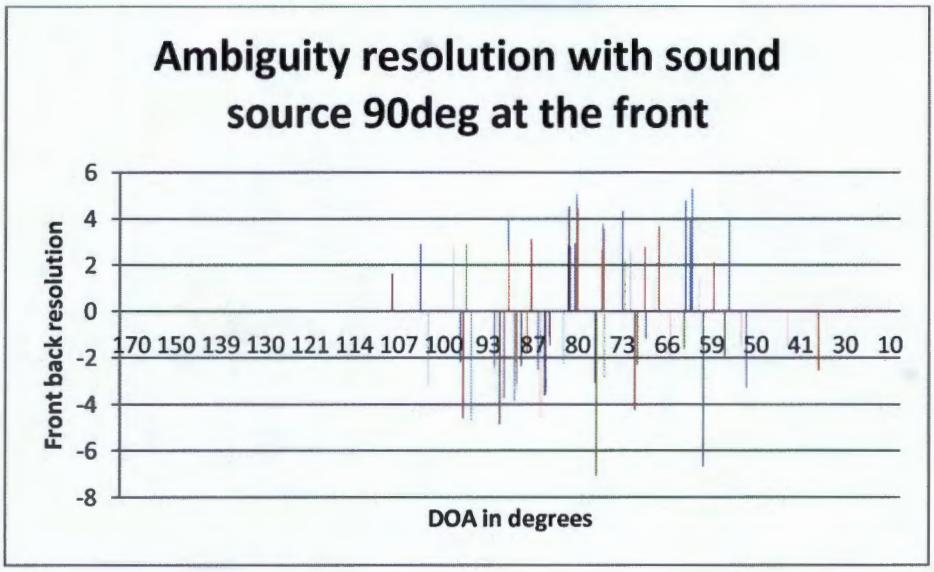


Figure 5.7.2-1 Front-back ambiguity resolution for sound source at 90deg in the front of sensor axis.

The figure below represents the estimations yielded by the system when the source is behind the sensor axis.

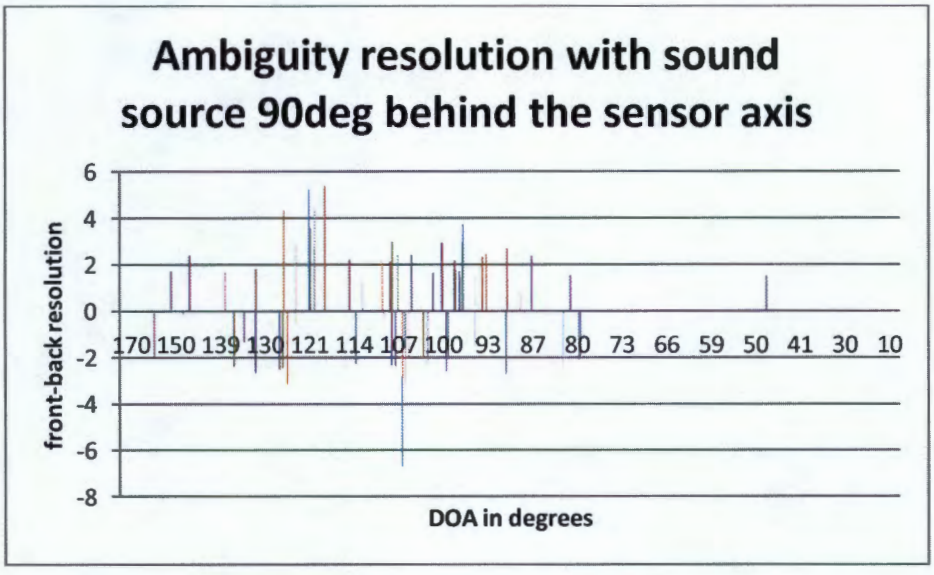


Figure 5.7.2-2 Front-back ambiguity resolution for sound source at 90deg behind the sensor axis.

5.7.3 Analysis & conclusion

From analysing 70 estimates for each of the test cases, both scenarios resulted in 27 samples falling within the expected range. This translated to an accuracy rate of 38.57% for each scenario. This value is too low to permit using this ambiguity resolution module in a final product. In addition to the low accuracy, using this module would require frequent phone rotations, which would not be user friendly.

5.8 Experiment 8: System processing time

The system processing time tests are performed in order to analyse the average time taken to yield a DOA estimate after the sound signals have been acquired by both microphones. A comparison amongst results from the best three dT settings established in experiment 5 are used to verify outcomes of the best dT setting.

5.8.1 Set up and expected hypothesis

Experiment 5 showed that dT=43 produced the highest accuracy in DOA estimations by the system. This was followed by dT=23 and dT =11 respectively. In order to establish the average processing time taken by the system to display a DOA estimate, the same procedure from experiment 2 is utilised, taking the critical section to be the entire processing carried out from the time the first sound frame is acquired till the last sound frame in a cluster, determined by dT, has been fully processed to yield DOA estimate to be displayed on the user interface.

5.8.2 Experiment results

The average processing times for the three dT values are tabulated in the table below.

Table 5.8.2-1 Average DOA processing time per dT setting

dT	Time in Seconds
11	0.278
23	0.557
43	1.020

The results are plotted in the graph below with a polynomial trend line to guide future work when optimising processing time against accuracy.

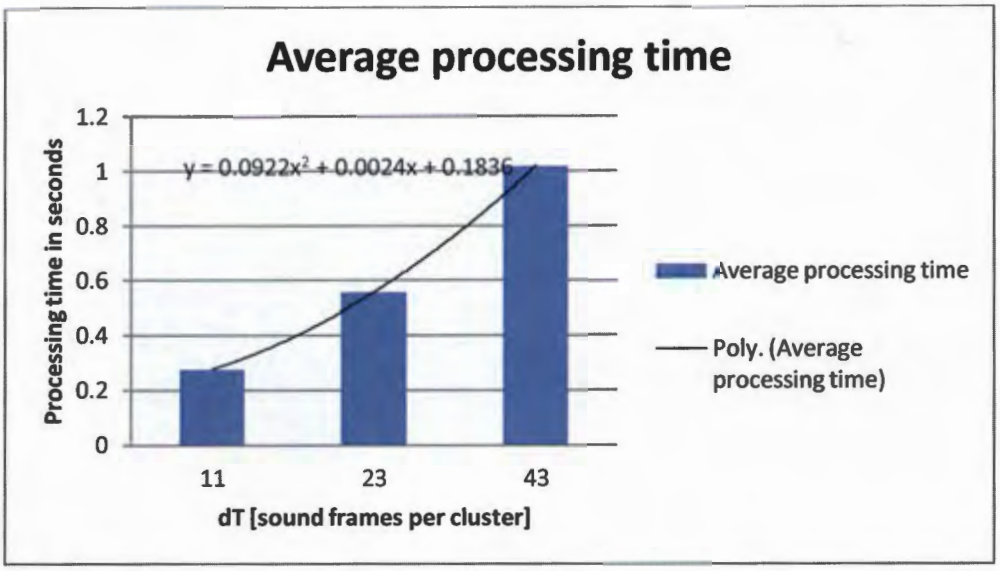


Figure 5.8.2-1 Average DOA processing time per dT setting

5.8.3 Analysis & conclusion

Based on results from experiment 5, dT setting of 43 sound frames per cluster was selected as the dT value yielding the highest accuracy rates. The amount of time taken to process 43 sound frames into a DOA estimate was found to be 1.02seconds. The maximum time expected was 1 second. It can hence be concluded that the system processing time meets the set design requirement of having a throughput of at least one DOA estimate per second.

5.9 Experiment 9: Source separation

Source separation tests were meant to test whether it would be possible to perform source separation in the time domain as a continuation to the results attained from experiment 5 and 6. The idea to perform source separation in the time domain was inspired by the fact that most methods described in section 2.7, used a combination of frequency and time domain to perform source separation.

5.9.1 *Set up and expected hypothesis*

In these tests, the system is expected to create wav output files labelled according to the DOAs of the available active sound sources. The wav files are expected to contain the corresponding sound generated by a source in that direction

A basic test was carried out for 2 sound sources to see if the sources could be separated and clustered in files linked to the estimated DOA. At 90°, a sound source of pop music, and at 130°, a sound source of indie rock music.

5.9.2 *Experiment results*

The figure below is of a screen shot listing the output files from this test.

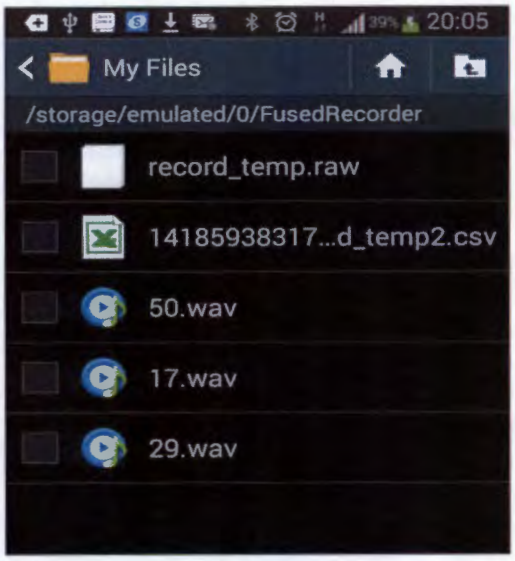


Figure 5.9.2-1 listing of system's outputs

The wav file titled 50, is the master file containing the un-separated sound sources. The '.raw' file is the data file containing the sound data in PCM form as recorded from the ADC buffers. The '.csv' file is the log file containing estimation data per dT cluster. The remaining wav files inherently represent the number of audible sound sources, the DOAs of the sound and the separated sounds.

The graphs below are a visual presentation of the 3 wav files that were generated in this experiment

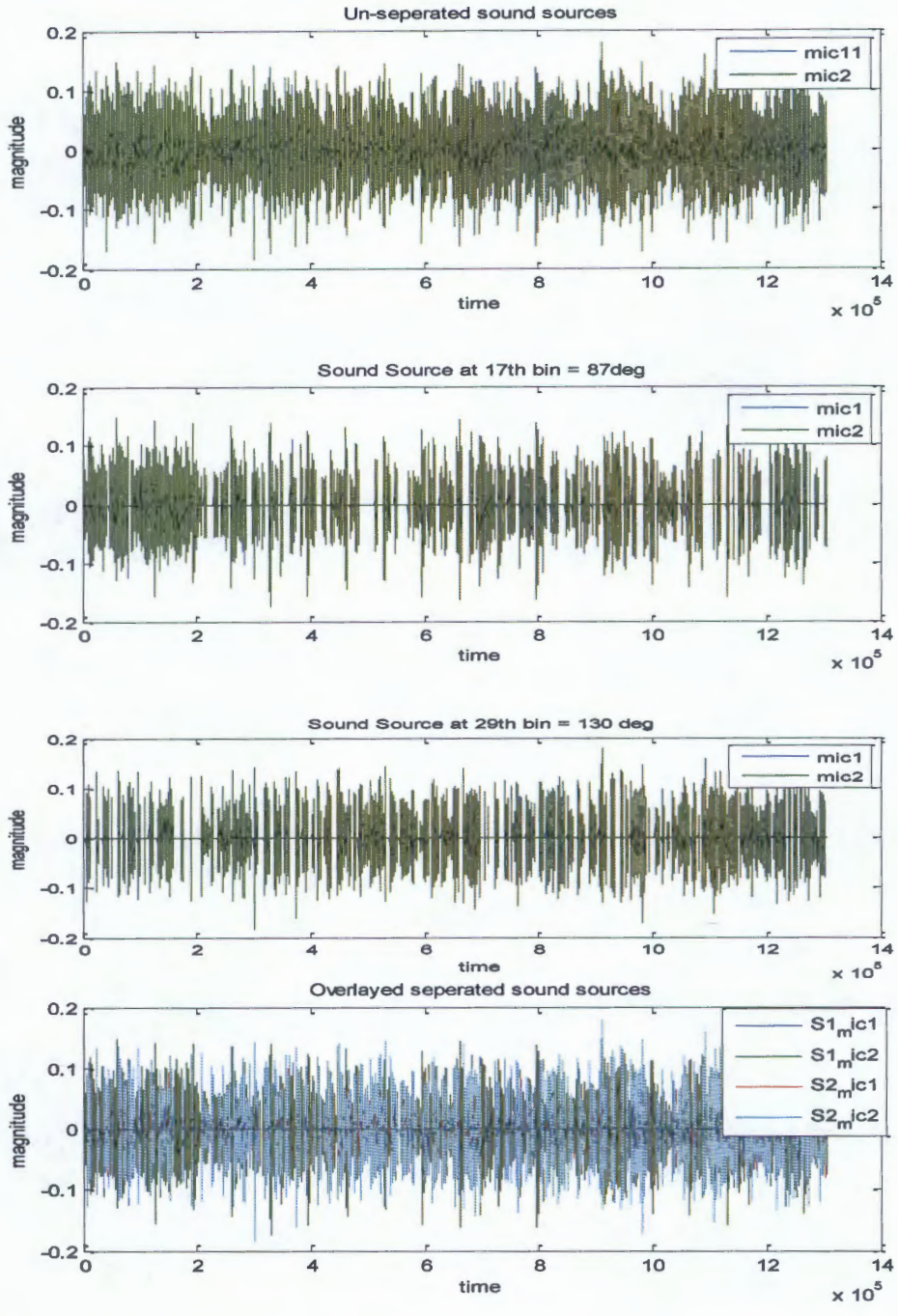


Figure 5.9.2-2 Matlab plots of un-separated sources , source1, source2, and the 2 sources super-imposed over each other to show that they make up the master file.

5.9.3 Analysis & conclusion

This format of output provides the three aspects of this project (sound source localisation, source counting, and source separation) in a tangible form that can be used by 3rd party systems either directly or through an API.

However, although it is evident that source separation does happen, the resultant audio files are of low quality due to regular 'quiet' spots created in each audio file as a result of the separation procedure. Whether the separated files provide meaningful data for a computer system is left as a topic for future work in Computer Scene Analysis.

One can listen to the wav files included in the tests' folder in the CD attached to this dissertation.

6 Conclusions, Recommendations, and Future work

A stochastic signal power accumulation algorithm (referred to as the stochastic histogram method in most parts of the text) was successfully designed and implemented on an android Galaxy S3 smart phone, using GCC-PHAT as underlying basis TDOA algorithm in order to perform azimuth sound source localisation.

During the phase of development, a critical parameter surfaced as part of the stochastic estimation algorithm, which necessitated a detailed analysis of the effect that changing this parameter had on the overall performance of the system. The research questions were concluded as follows:

6.1 What would be the most suitable algorithm for performing SSL on a smart phone?

From analysis of literature, GCC-PHAT was selected as the most suitable algorithm to use for sound source localisation on a smart phone. It is a fairly accurate algorithm in ideal conditions and is not computationally too demanding. Experiment 2 showed that GCC-PHAT processing time took an average of 10.32ms for 23ms worth of sound frames.

To improve estimation accuracy in realistic environments, the GCC-PHAT algorithm was modified by including a supplementary algorithm that used direction estimates produced by the GCC-PHAT algorithm, to cluster signal power received by the sensor most inclined towards the sound source.

Due to this modification, a critical parameter, dT, was developed to analyse how many time-frames needed to be accumulated in order to have the best results.

6.2 Is it possible to use orientation data to overcome localisation ambiguity caused due to microphone constraints?

Yes, it is possible to use orientation data to resolve ambiguity amongst DOAs estimated on the phone, however, from experiment 7; the proposed method was found to be highly inaccurate and impractical for a smart phone due to the constant rotations the user would need to make to achieve real time readings. The accuracy of ambiguity resolution was found to be 38.57% for both cases of sources at the back or front of the microphone axis.

6.3 What clustering value for parameter dT yields the best results?

From analysis in experiment 5, it was found that dT = 43 yielded the highest accuracies across all associated subsets of experiments constituting experiment 5.

6.4 What would be the accuracy of the sound source localisation algorithm?

Two accuracy analysis approaches were used in experiment 5: pin-point accuracy, which yielded 64%, overall accuracy for the system at dT= 43 as a function of both time and DOA; $\pm$ 1 sample tolerance analysis which yielded 90.67% overall accuracy for the same dT conditions.

6.5 What would be the time response of the proposed system?

When using dT = 43, the system averaged one DOA estimate per 1.02seconds including signal acquisition time. This throughput enables a real time response by the system.

6.6 Can source counting and separation be done within the same framework of the proposed algorithm?

The stochastic histogram algorithm was extended to perform source counting and source separation.

Source separation was performed in the time domain but yielded low quality separated sound source files.

Source counting in Experiment 6 showed that the cumulative power histogram generated by the stochastic histogram algorithm, has Gaussian-like peaks centred at DOA bins approximating the direction of active sound sources within an environment. Smoothing this histogram led to 100% counting rates for all test cases.

This was a critical step in laying foundation for future work that will aim at performing computational sound scene analysis with this work as the basis.

6.7 Recommendations

Rather than use orientation data for solving ambiguity, it would be more computationally and logically efficient to include a 3rd microphone to smart phones so the same system developed in this project be extended to use 2 microphone pairs in order to perform triangulation with the DOA estimates from both microphone pairs. This would add capability of real time localisation, ranging and tracking in 360° to the system.

The design considerations in section 4.3 can be used to establish a suitable trade off between maximum sampling frequency of a 3 channel ADC and microphone spacing over an L-shaped geometry that matches most smart phone dimensions.

Associated APIs would also have to be modified to facilitate these changes.

6.8 Future work

This project is meant to be developed further into an A.I. assisted computational sound scene analysis system. The sound source localisation, counting and separation functionality is to be developed into a system that enables smart phone users train A.I systems about different sound sources in their environments, using smartphones as auditory sensory devices. The auditory processing functionality could be integrated with visual processing in order to teach A.I systems which objects produce which sounds under which circumstances. This crowd sourced approach to knowledge gathering would enable researchers to easily create vast centralised Big Data databases from which highly intelligent A.I systems could be built. The first system in mind being an A.I system capable of independently performing sound scene analysis using a smart phone as an avatar, in order to provide guidance to a visually or hearing impaired user. Another system that comes to mind is a system that tracks, assesses, and creates reports about security risks by observing its environment through audio and visual cues.

To simplify development of such systems, APIs and optimised libraries will have to be developed as components of an A.I framework that utilises among other things, smartphones as input/output sensors.

7 References

[1] J.-M. Valin, F. Michaud, and J. Rouat, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering," *Rob. Auton. Syst.*, vol. 55, no. 3, pp. 216–228, Mar. 2007.

[2] J. C. Murray, H. R. Erwin, and S. Wermter, "Robotic sound-source localisation architecture using cross-correlation and recurrent neural networks.," *Neural Netw.*, vol. 22, no. 2, pp. 173–89, Mar. 2009.

[3] A. Czyzewski, "Automatic identification of sound source position employing neural networks and rough sets," *Pattern Recognit. Lett.*, vol. 24, no. 6, pp. 921–933, Mar. 2003.

[4] D. Brungart and W. Rabinowitz, "Auditory localization of nearby sources. Head-related transfer functions," *J. Acoust. Soc. Am.*, vol. 106, no. 3 Pt 1, pp. 1465–1479, Sep. 1999.

[5] E. Grassi, J. Tulsi, and S. Shamma, "Measurement of head-related transfer functions based on the empirical transfer function estimate," in *Proc. ICAD*, 2003, pp. 119–122.

[6] U.S. Congress and Office of Technology Assessment, "Hearing Impairment and Elderly People–A Background Paper," OTA-BP-BA-30 (Washington, DC: U.S. Government Printing Office, May 1986).

[7] Pricess Alexandra Hospital South Brisbane Health District, "Deafness and Mental Health Guidelines for Working with People who are Deaf or Hard of Hearing," Brisbane, Australia, 2008.

[8] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Editio. Prentice Hall, 2010.

[9] B. Perry, "Memories of fear," in *Splintered reflections. Washington, DC: Basic Books*, 1999.

[10] B. Harrub, B. Thompson, and D. Miller, "The Origin Of Language And Communication," *J. Creat.*, vol. 17, no. 3, pp. 93–101, 2003.

[11] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Am.*, vol. 120, no. 5, p. 2421, 2006.

[12] S. Swanson and M. Taylor, "Greendroid: Exploring the next evolution in smartphone application processors," *Commun. Mag. IEEE*, no. April, pp. 112–119, 2011.

[13] M. Meeker, "Internet Trends 2014 – Code Conference," 2014. [Online]. Available: http://www.kpcb.com/blog/2014-internet-trends. [Accessed: 05-Feb-2015].

[14] J. O. Smith, "Spherical Waves from a Point Source," *Physical Audio Signal Processing*. [Online]. Available: https://ccrma.stanford.edu/~jos/pasp/Spherical_Waves_Point_Source.html. [Accessed: 09-Dec-2014].

[15] C. H. Knapp and G. C. Carter, "The Generalised Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 2, pp. 320–327, 1976.

[16] M. a Akeroyd, "The psychoacoustics of binaural hearing.," *Int. J. Audiol.*, vol. 45 Suppl 1, no. Supplement 1, pp. S25–33, Jan. 2006.

[17] B. Yin, P. C. W. Sommen, and P. He, "Exploiting Acoustic Similarity of Propagating Paths for Audio Signal Seperation," *EURASIP J. Appl. Signal Processing*, vol. 2003, no. 11, pp. 1091–1109, 2003.

[18] R. F. Lyon, "A Computational Model of Binaural Localization and Separation," in *ICASSP 83*, 1983, pp. 1148–1151.

[19] A. Clifford and J. Reiss, "Calculating time delays of multiple active sources in live sound," in *Audio Engineering Society Convention 129*, 2010, pp. 1–8.

[20] P. R. Roth, "Effective measurements using digital signal analysis," *IEEE Spetrum*, vol. 4, no. 8, pp. 62–70, 1971.

[21] G. C. Carter, "The Smoothed Coherence Transform," in *Proceedings of the IEEE*, 1973, pp. 1497–1498.

[22] A. Umbarkar, "Improved Sound-based Localization Through a Network of Reconfigurable Mixed-Signal Nodes," Stony Brook University, 2010.

[23] A. Brutti, M. Omologo, P. Svaizer, and F. Bruno, "Comparison between different sound source localization techniques based on a real data collection," in *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*, 2008, pp. 69–72.

[24] H. T. Do, "Robust cross-correlation-based methods for sound-source localization and separation using a large-aperture microphone array," Brown University, 2011.

[25] S. T. Birchfield and R. Gangishetty, "Acoustic Localisation By Interaural Level Difference," in *Conference, Ieee International Processing, Signal*, 2005, vol. 2, no. March, pp. 2–5.

[26] A. Deleforge and R. Horaud, "The Cocktail Party Robot : Sound Source Separation and Localisation with an Active Binaural Head," in *Perception and Recognition*, 2012, pp. 431–438.

[27] M. Aytekin, E. Grassi, M. Sahota, and C. F. Moss, "The bat head-related transfer function reveals binaural cues for sound localization in azimuth and elevation," *J. Acoust. Soc. Am.*, vol. 116, no. 6, p. 3594, 2004.

[28] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile Approach to Spatial Filtering," *IEEE ASSP magazine*, pp. 4–24, Apr-1988.

[29] J. Chen, K. Yao, and R. Hudson, "Source localization and beamforming," *IEEE Signal Process. Mag.*, no. March, pp. 30 – 39, 2002.

[30] Y. Zheng and R. Goubran, "A microphone array system for multimedia applications with near-field signal targets," *IEEE Sens. J.*, vol. 5, no. 6, pp. 1395–1406, 2005.

[31] K. Varma, "Time-delay-estimate based direction-of-arrival estimation for speech in reverberant environments," Virginia Polytechnic Institute and State University, 2002.

[32] J. H. Dibiase, "A High-Accuracy , Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays," Brown University, 2000.

[33] C. Zhang and D. Florêncio, "Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings," *IEEE Trans. Multimed.*, vol. 10, no. 3, pp. 538–548, 2008.

[34] M. Omologo, "Acoustic event localization using a crosspower-spectrum phase based technique," no. 2, pp. 273–276, 1994.

[35] J. Benesty, S. Member, J. Chen, and Y. A. Huang, "On Microphone-Array Beamforming From a MIMO Acoustic Signal Processing Perspective," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 15, no. 3, pp. 1053–1065, 2007.

[36] L. Kneip and C. Baumann, "Binaural model for artificial spatial sound localization based on interaural time delays and movements of the interaural axis.," *J. Acoust. Soc. Am.*, vol. 124, no. 5, pp. 3108–19, Nov. 2008.

[37] S. Carlile, P. Leong, and S. Hyams, "The nature and distribution of errors in sound localization by human listeners," *Hear. Res.*, vol. 114, no. 1997, pp. 179–196, 1997.

[38] B. G. Shinn-cunningham, S. Santarelli, and N. Kopco, "Tori of confusion : Binaural localization cues for sources within reach of a listener," *J. Acoust. Soc. Am.*, vol. 107, no. 3, pp. 1627–1636, 2015.

[39] E. C. Cherry, "Some Experiments on the Recognition of Speech, with One and with Two Ears," *J. Acoust. Soc. Am.*, vol. 25, no. 5, pp. 975–979, 1953.

[40] A. R. a. Conway, N. Cowan, and M. F. Bunting, "The cocktail party phenomenon revisited: The importance of working memory capacity," *Psychon. Bull. Rev.*, vol. 8, no. 2, pp. 331–335, Jun. 2001.

[41] N. Roman and D. Wang, "Binaural sound segregation for multisource reverberant environments," in *ICASSP 2004*, 2004, pp. 373–376.

[42] C. Kim, K. Kumar, and R. M. Stern, "Binaural sound source seperation motivated by auditory processing," in *ICASSP 2011*, 2011, pp. 5072–5075.

[43] D. A. Effects, "On the use of spatial cues to improve binaural source seperation," in *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, 2003, pp. 1–5.

[44] Ö. Yılmaz and S. Rickard, "Blind Separation of Speech Mixtures via Time-Frequency Masking," *IEEE Trans. SIGNAL Process.*, vol. 52, no. 7, pp. 1830–1847, 2004.

[45] J. B. Allen and L. R. Rabiner, "Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[46] J. Amaral, "About Computing Science Research Methodology." 2011.

[47] P. Lawitzki, "Application of Dynamic Binaural Signals in Acoustic Games," Stuttgart Media University, 2012.

[48] Google Inc. and Open Handset Alliance, "Android Developers," 2015. [Online]. Available: www.developer.android.com/reference/android/media/AudioRecord.html. [Accessed: 02-Feb-2015].

[49] Digi-Phd, "Android Java: Simple fft example using Libgdx," 2014. [Online]. Available: http://digiphd.com/android-java-simple-fft-libgdx/. [Accessed: 02-Feb-2014].

[50] J. Shelton and G. P. Kumar, "Comparison between Auditory and Visual Simple Reaction Times," *Neurosci. Med.*, vol. 1, no. September, pp. 30–32, 2010.

Appendix A

Contents of CD

- Dissertation soft copy
- Experiment data
- Android Project file
- SoundSourceLocalisation_code

Appendix B

The figures below represent stacked cumulative power histograms for 1m and 2m test cases mentioned in section 5.5.2 on pg 94.

1m cumulative graphs

Fig B.1.1 to fig B.1.5 are from tests with sources placed at 1m distance angled at 10deg, 50deg, 90deg, 130deg, and 170 deg.

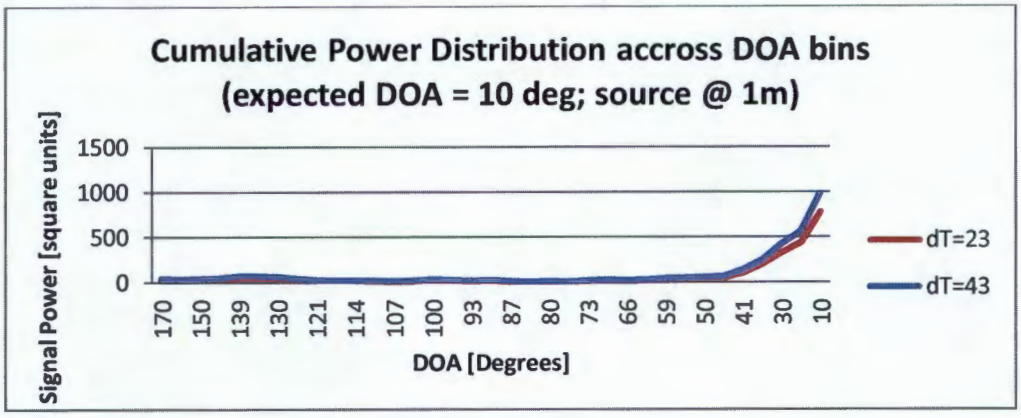


Figure B.1.1 stacked accumulated power histograms peaking at DOA=10deg

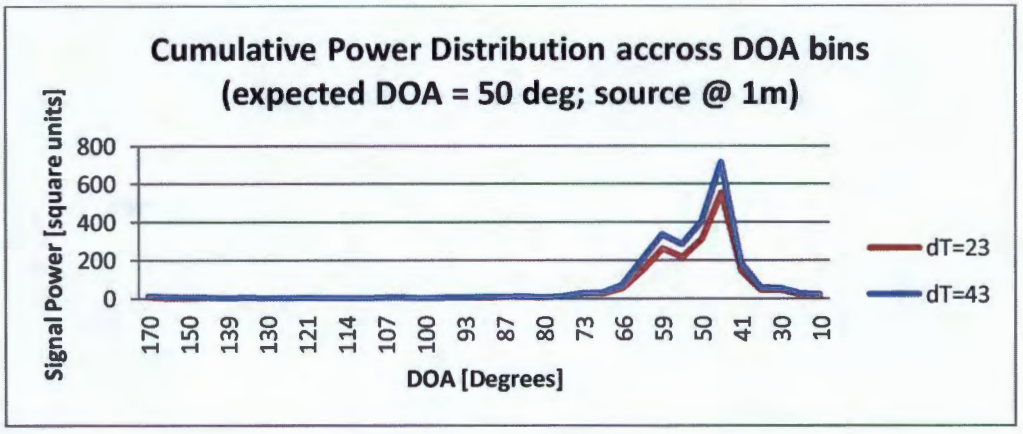


Figure B.1.2 stacked accumulated power histogram peaking at DOA=46deg

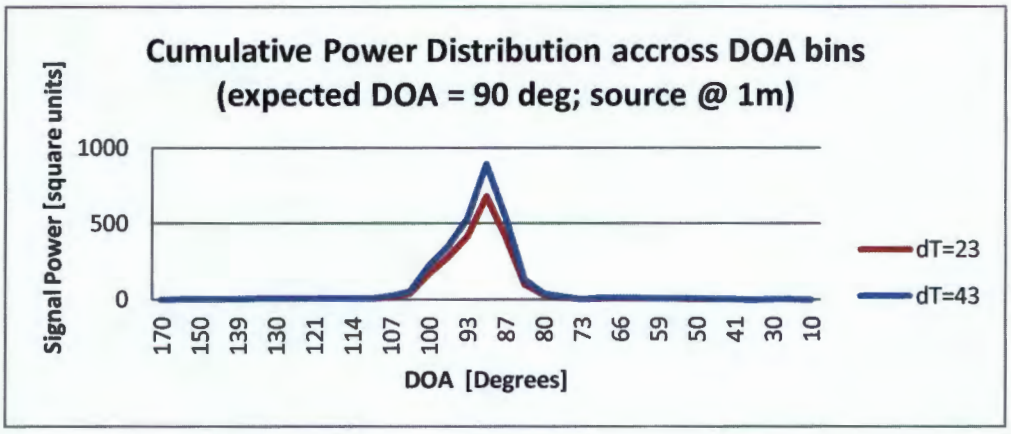


Figure B.1.3 stacked accumulated power histograms peaking at DOA=90deg

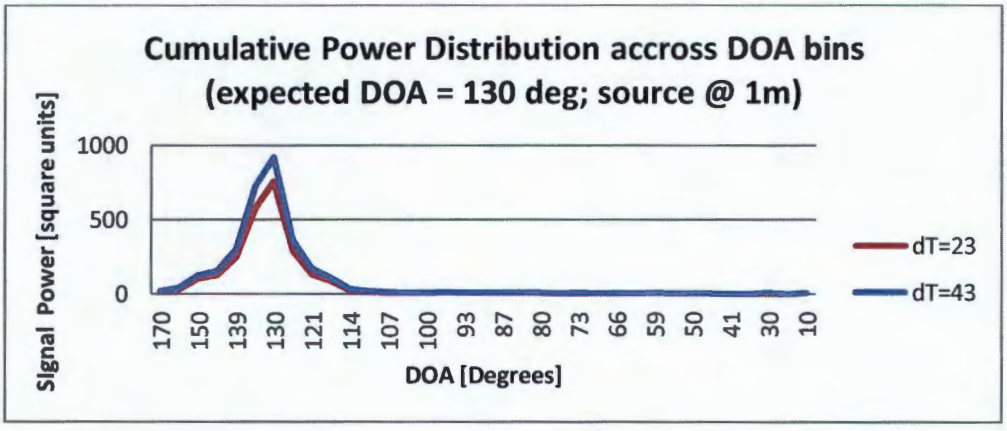


Figure B.1.4 stacked accumulated power histograms peaking at DOA=130deg

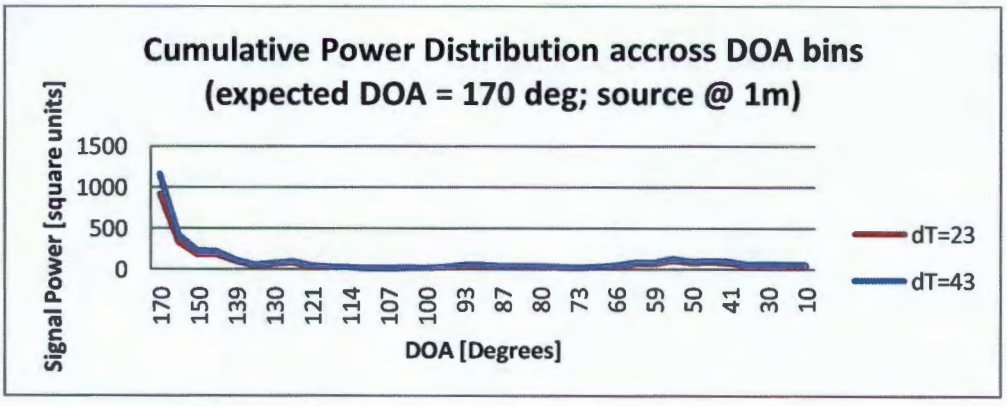


Figure B.1.5 stacked accumulated power histograms peaking at DOA=170deg

2m cumulative graphs

Fig B.2.1 to fig B.2.5 are from tests with sources placed at 2m distance angled at 10deg, 50deg, 90deg, 130deg, and 170 deg.

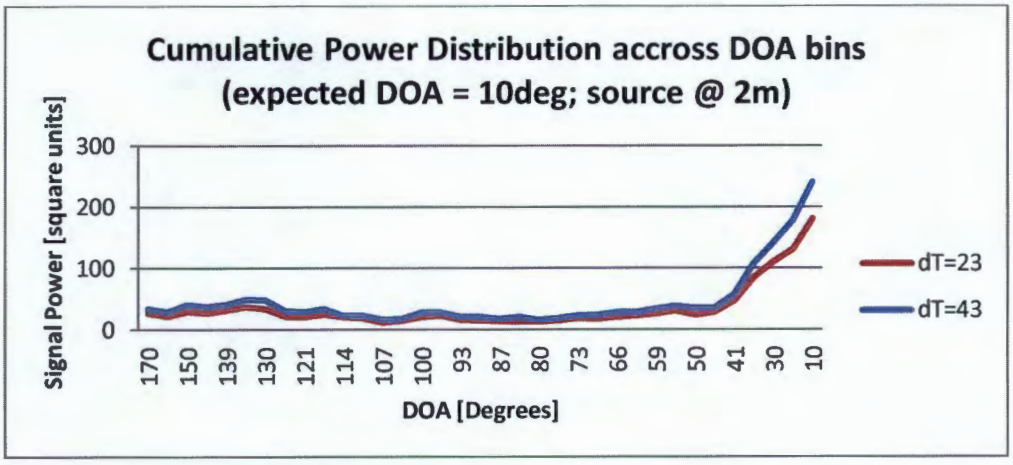


Figure B.2.1 stacked accumulated power histograms peaking at DOA=10deg

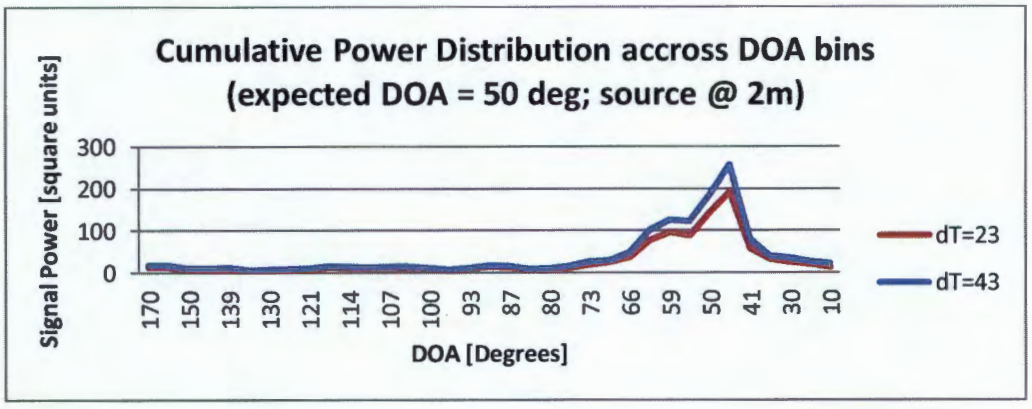


Figure B.2.2 stacked accumulated power histogram peaking at DOA=46deg

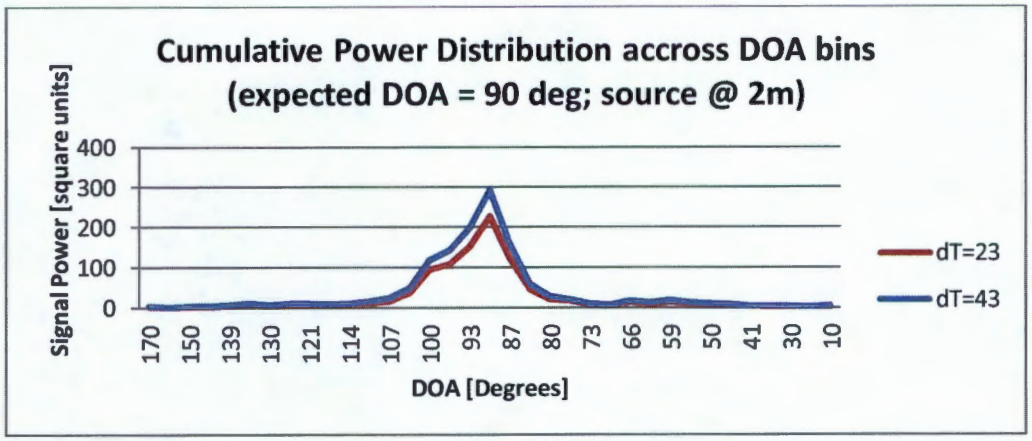


Figure B.2.3 stacked accumulated power histograms peaking at DOA=90deg

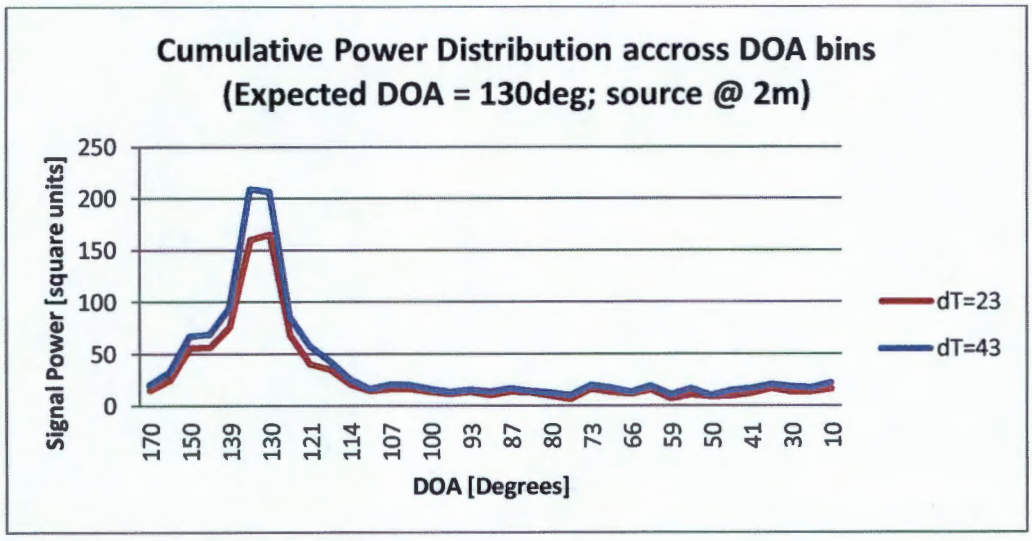


Figure B.2.4 stacked accumulated power histograms peaking at DOA=125deg for dT =23; and DOA=134deg for dT=43.

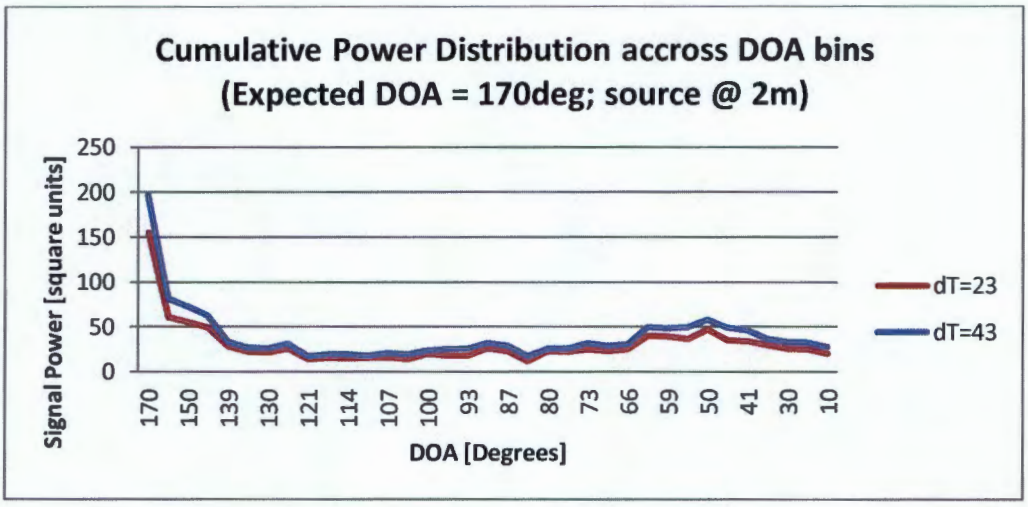


Figure B.2.5 stacked accumulated power histograms peaking at DOA=170deg