Acoustic Source Angle and Position estimation using Microphone Arrays

A THESIS

to be submitted by

Pusuluri Ramakrishna Chaitanya

EE08B046

for the award of the degree

of

BACHELOR OF TECHNOLOGY

and

MASTER OF TECHNOLOGY

under the guidance of

Prof. Nitin Chandrachoodan

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

CHENNAI-600036

CERTIFICATE

This is to certify that the thesis titled **"Acoustic Source Angle and Position estimation using Microphone Arrays",** submitted by Mr. Pusuluri Ramakrishna Chaitanya, to the Indian Institute of Technology Madras, Chennai for the award of the degree of Bachelor of Technology and Master of Technology, is bonafide record of research work done by him under my supervision. The contents of this thesis, in full or parts, have not been submitted to any other institute or university for the award of any degree or diploma.

Dr. Nitin Chandrachoodan
Project Guide
Assistant Professor
Dept. of Electrical Engineering
IIT Madras, Chennai-600036

Place: Chennai

Date: 8^{th} May 2013

ACKNOWLEDGMENT

I would like to express my deepest gratitude to Dr. Nitin Chandrachoodan for giving me this opportunity to work under his able guidance. Without his invaluable advice and help, I wouldn't have been able to complete my project. Working for this project has improved my technical skills and broadened my understanding of many aspects of Electrical Engineering. I would also like to thank Dr. Nitin Chandrachoodan for letting me work as a Teaching Assistant to him during the course of my project.

Many thanks to Dr.Vinita Vasudevan for being on my committee and reviewing my work.

I wouldn't forget the endless hours of discussion, both technical and otherwise, with my lab mates : Amit, Karthik, Seetal and Satish, Surya and Vikas, that have been extremely helpful in my work.

Finally, I would like to express my gratitude to my parents for being with me always and making me who I am.

Chaitanya R Pusuluri

Abstract

Time delay estimation (TDE)-based algorithms for estimation of direction of arrival (DOA) have been most popular for use with speech signals. This is due to their simplicity and low computational requirements. Though other algorithms, like the steered response power with phase transform (SRP-PHAT), are available that perform better than TDE based algorithms, the huge computational load required for this algorithm makes it unsuitable for applications that require fast refresh rates using short frames. In addition, the estimation errors that do occur with SRP-PHAT tend to be large. This kind of performance is unsuitable for an application such as video camera steering, which is much less tolerant to large errors than it is to small errors.

We use Generalized Cross Correlation (GCC) technique, in this project to estimate the time delays between microphones in a microphone array. Using this information for a pair of micropones, the 2 dimensional angle at which the sound source is present is estimated. This is experimented both on CPU and also a microcontroller (memory constrained) and the results are presented. The sensitivity analysis with respect to error in speed of sound, array geometry, SNR is studied. This analysis is also done for 3D position estimation using a square shaped Four microphone array.

Contents

List of Tables

List of Figures

Abbreviation

CC	Cross Correlation
GCC	Generalized Cross Correlation
PHAT	PHAse Transform
ML	Maximum Likelihood
ASDF	Average Square Difference Function
LMS	Least Mean Square
SCOT	Smoothed Coherence Transform
MXS	Matching X and S
MSX	Matching S and X
AMDF	Average Magnitude Difference Function
TDE	Time Delay Estimation
DOA	Direction Of Arrival
TDOA	Time Difference Of Arrival
ECM	Electret Condenser Microphone
SNR	Signal to Noise Ratio

1 Introduction

1.1 Motivation for project

Direction of Arrival (DOA) on the basis of the received sound signals from an array of microphones has many practical applications in everyday life. DOA estimates from microphone arrays can be used to steer the camera towards the speaker in a conference or a long distance TV based classroom [50]. To this end, presently there are commercially available systems in which a user or a group of users can steer the camera towards him/them using a manual switch. But these systems are quite expensive in terms of manpower and extra hardware. So an automatically steered camera would be of great use. Most conferences and classrooms consist of a single speaker. But he/she may be moving. So, the problem mainly boils down to tracking single sound source which is moving. A comprehensive tracking system using video data has been proposed by Wren et.al. [51]. However, it is very algorithmically complex and computationally intensive that a computer has to be dedicated for this. Methods based on acoustic data are typically far simpler with respect to computational complexity.

1.2 Background

In the human body, the system formed by the ears and the brain can receive, process and detect the location of the source to a certain range of accuracy [5]. From19th century only, people have tried to build devices that mimic this feature. One of the popular ways to implement this is to use the microphone arrays. However, in a number of interesting ways, sound location can be found in the nature. In several animals, for location of objects around them, sound location is known to be used rather than eye vision. For instance, animals like bats or dolphins perform a technique called echolocation. These animals can perceive their environment by emitting sounds and processing the echoes caused by the reflections [6]-[7]. It was proved that human beings can also develop this capability and that it can be applied to blind people [8].

One of the first purposes of sound location was for military applications, as many other inventions all along the history. A method called sound ranging was developed. It obtains the coordinates of a hostile by the its gun firing sound [9]. This way, even without direct vision, the enemy could be detected. This technique, started out in World War I and has been used during the whole 20th century. But later, new technologies like gun detective radars replaced the sound ranging. Even so, some armies still use it in their operations [10].

Besides the military application, many other studies about sound location have been carried out. Some of the most important are the techniques used in noise or echo cancellation [11]-[13]. These can be very important, depending on the type of the environment whether it is closed or open. In other works, the aim is to minimize the error [14] and improve the efficiency. The advancement in upcoming technology is commonly the number of microphones used [15]-[19] or the way the signals are processed after the sound is captured [20]-[23]. Furthermore it is noteworthy that some radio localization systems have been implemented. In these the principle is the same as in sound location but the signals are radio waves instead of acoustic waves [27].

1.3 Methods for estimating DOA

As we see in the subsequent sections, the whole project is based on the calculation of time delay between received signals at different microphones.

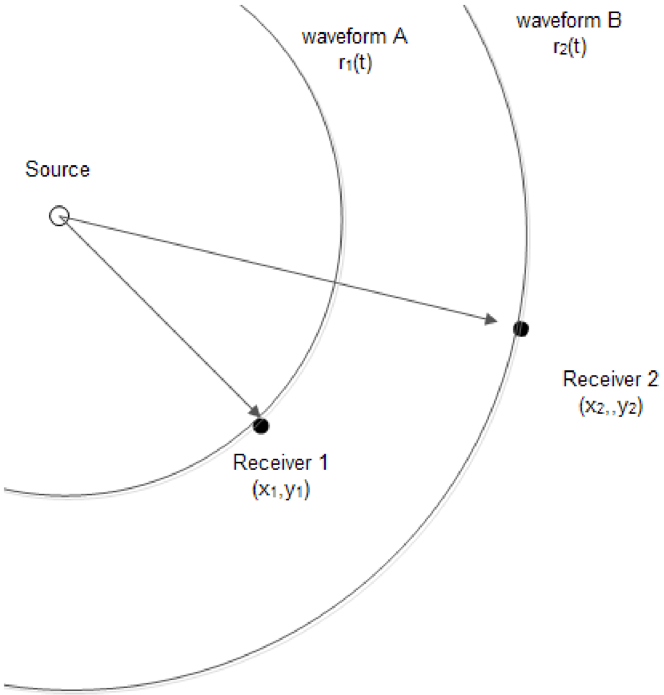


Figure 1.1: he source and microphones in 2D

From the figure 1.1, it is clear that the wave forms the receivers are exposed to are both identical in shape but one is delayed version of another. If d is the time delay between the two waveforms, then we can say, in the absence of any adding waves like noise,

$$r_2(t) = r_1(t-d)$$

Using the time delay d for different pairs of microphones in an array, we can calculate the angle and location of the source with respect to the microphone array, as shown in sections 4 and 5. There are many algorithms to estimate the time delay d [28]. The cross-correlation(CC) technique is a popular solution technique. Many other TDE methods were developed using CC. Basically, the CC method estimates the time argument that corresponds to the peak in the correlation curve of the two microphone outputs. To improve the peak detection and time delay estimation, many weighing functions are proposed to be used after CC [29]. We can also use the filtered versions of signals to estimate the peak of the cross correlation. This method is called the Generalized Cross Correlation (GCC) [29]. The GCC method, proposed by Knapp and Carter in 1976, is the most popular technique for TDE due to their accuracy and moderate computational complexity. The role of filter or weighing function in GCC is to ensure a sharp peak in cross correlation, so as to have a good time delay resolution. There are many techniques used to select the weighting function; such as the Roth Processor, the Smoothed Coherence Transform (SCOT), the Phase Transform (PHAT), the Eckart Filter, and the Maximum Likelihood (ML) estimator [28,29]. These correlation-based methods yield ambiguous results when the noises at the two sensors are correlated with the desired signals. To overcome this problem, higher-order statistics methods were employed [28,30].

There are also some other algorithms used to estimate the time-delay. The matching x and s (MXS) and matching s and x (MSX) methods [31] compare the cross-correlation of the received signals and with the auto-correlation of the reference signal. Algorithms based on minimum error: average square difference function (ASDF) and average magnitude difference function (AMDF), seek position of the minimum difference between signals and [31]. Adaptive algorithms such as LMS can also be introduced into the TDE [33]. In these algorithms, the delay estimation process is reduced to a filter delay that gives minimal error. Nowadays, many other methods are employed in the TDE, such as, MUSIC [34], ESPRIT [45], and wavelets [35].

The following subsection described five TDE methods.

1.4 TDE methods

There are several TDE algorithms each with its own advantages in computation complexity, hardware implementation, and precision. Five of these commonly used TDE methods are described here.

1.4.1 Cross Correlation (CC) method

Here, the cross correlation curve of the two received signals is computed. Then maximum peak in the curve is estimated, which represents the time delay.

$$R_{r_r r_2}(t) = E[r_1(t) r_2(t-\tau)]$$

$$d_{cc} = arg_t max\,[R_{r_r r_2}(\tau)]$$

1.4.2 Phase Transform Method (PHAT method)

This is a CC method with some weighing function (making it a GCC technique). The PHAT received considerable attention due to its ability to avoid spreading of the peak of the correlation function [43]. This can be mathematically expressed as :

$$R_{r_1 r_2}(\tau) = \int \psi_p(f) G_{r_1 r_2}(f)\, e^{2\pi j f \tau} df$$

$$where\ \psi_p(f) = \frac{1}{|G_{r_1 r_2}(f)|}$$

$$d_{PHAT} = arg_t max\,[R_{r_r r_2}(\tau)]$$

where $G_{r_1 r_2}(f)$ is the cross-spectrum of the received signals and $\psi_p(f)$ is the PHAT weighing function.

1.4.3 Maximum Likelihood method (ML method)

The ML (is identical to the HT method proposed by Hannan and Thomson [28]) is another important method within the GCC family since it gives the maximum likelihood solution for TDE problem. The ML weighting function $\psi_{ML}(f)$ is chosen to improve the accuracy of the estimated delay by attenuating the signals fed into the correlator in spectral region where the SNR is the lowest. The ML estimator is popular because of its optimality under appropriate conditions. For uncorrelated Gaussian signal and noise and non reverberant environments, the ML estimator of time delay is asymptotically unbiased and efficient in the limit of long observation intervals [39]. The ML method can be mathematically represented by:

$$R_{r_1r_2}(\tau) = \int \psi_{ML}(f) G_{r_1r_2}(f)\, e^{2\pi jf\tau} df$$

$$where\ \psi_{ML}(f) = \frac{1}{|G_{r_1r_2}(f)|} \frac{|\,\gamma_{r_1r_2}(f)\,|^2}{|1-\gamma_{r_1r_2}(f)\,|^2}$$

$$d_{ML} = arg_t max\,[R_{r_rr_2}(\tau)]$$

where

$$|\,\gamma_{r_1r_2}(f)\,|^2 = \frac{|G_{r_1r_2}(f)|^2}{G_{r_1r_1}(f)G_{r_2r_2}(f)}$$

is the magnitude coherency squared and $\psi_{ML}(f)$ is the ML weighing function. As ML contains the term $\frac{|\,\gamma_{r_1r_2}(f)\,|^2}{|1-\gamma_{r_1r_2}(f)\,|^2}$, greater weight is placed on frequency bands that give near-unity coherence. In contrast, frequency bands in which the coherence is near zero are deemphasized [28]. The ML processor weights the cross-spectral phase according to the estimated cross-spectral phase when the variance of the estimated phase error is the least.

1.4.4 Average Square Difference Function (ASDF) method

The ASDF method is based on finding the position of the minimum error square between the two received noisy signals and considering this position value as the estimated time delay [44].

$$R_{r_1r_2}(\tau) = \frac{1}{N}\sum_{n=0}^{N-1}[r_1(n) - r_1(n+\tau)]^2$$

$$d_{ASDF} = arg_t min\,[R_{r_rr_2}(\tau)]$$

The favorable performance of the ASDF-based estimator is due to the fact that it gives perfect estimation in the absence of noise while the direct correlation does not [44]. Also, it is apparent that ASDF requires no multiplication, which is the most significant practical advantage over the other methods. Also, this technique does not require the knowledge of the input spectra.

1.4.5 Least mean square (LMS) adaptive filter method

The LMS adaptive filter is a finite impulse response (FIR) filter which automatically adapts its coefficients to minimize the mean square difference between its two inputs; a reference signal and a desired signal. It does not require any a prior knowledge of the input spectrum.

If $r_1(n)$ is the reference signal and $r_2(n)$ is the desired signal, then LMS output $y_{12}(n)$can be expressed as:

$$y_{12}(n) = W_{12}^T(n)X_{12}(n)$$

where T denotes transpose and

$$X_{12}(n) = [r_1(n), r_1(n-1), ..., r_1(n-L+1)]^T$$

is the filter state consisting of the most recent samples of the reference signal. The vector $W_{12}(n)$ is the L-vector of filter weights at instant n. The error output is

$$e_{12}(n) = r_2(n) - y_{12}(n)$$

The weight vector is updated every sample according to

$$W_{12}(n+1) = W_{12}(n) + \mu e_{12}(n) X_{12}^*(n)$$

where * denotes the complex conjugate and is a feedback coefficient that controls the rate of convergence and the algorithm stability. The algorithm adapts the FIR filter to insert a delay equal and opposite to that existing between the two signals. In an ideal situation (no additive noise), the filter weight corresponding to the true delay would be unity and all other weights would be zero [33]. When in the case where the additive noise exists, the filter weight corresponding to the true delay would be the maximum value compared to the other weights.

1.5 The method used in the project

Having researched about various TDE methods, the GCC method with band pass filters was chosen to be the apt method for our project. For 2D angle estimation and position coordinate estimation, this has been the method used in the project. But when it comes to using the redundancy by having extra microphones, the LMS method has been used. This method has shown to improve results by using extra microphones.

The GCC method is first programmed using Matlab. This is used for all the simulations in the subsequent sections. It is also used in the practical implementation of 2D angle estimation using the data acquisition by FPGA. Later, C program for the same technique has been developed for using with the memory constrained interface, i.e. Stellaris Evalbot.

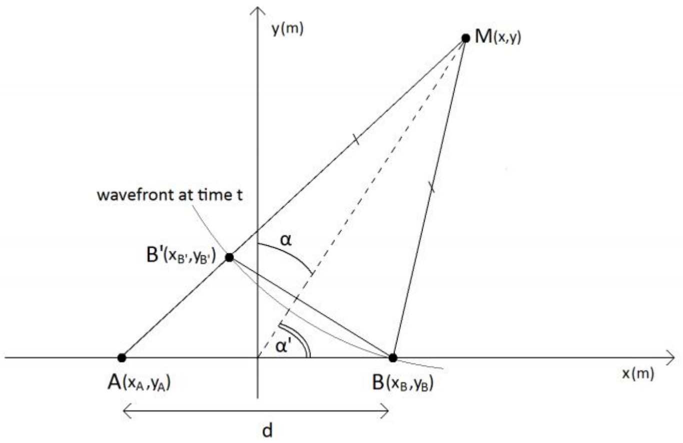
2 2D Angle Estimation

2.1 Applications

In many applications, the distance of the object may not be necessary but only the direction is needed. This enables to design less-complex systems, without giving up on good performance. For example on video conference, the system does not need to calculate the distance in which the source is emitting sounds. With the knowledge of the direction exclusively the camera can focus the speakers [2]. Other applications that require only the calculation of the direction are the audio surveillance systems. This kind of systems, used for intrusion detection [3] or gunfire location [4], determines the direction for locating the source, but ignores the distance.

2.2 Theory

As shown in the figure 2.1, let us assume an acoustic source located at M. There are two sound receivers MIC A and MIC B. Our aim is to find the angle α.This is the angle in the plane of the microphones, considering the perpendicular bisector of the microphones as the reference. As seen in the figure, there will be delay in arrival between the sound waves observed at both the receivers. We shall be using the time delay, designated by τ, for the calculation of the angle.



,

Figure 2.1: Theoretical derivation of angle

Let us see the mathematics behind finding the angle. As shown in the figure 2.1 , let the two microphones be located at A(x_A, y$_A$) and B(x_B, y$_B$). Let the source is at M(x,y). We are interested in finding the angle α. Here, the origin is the midpoint of AB. For our convenience, without loss of generality, let us assume, AB is on the $x-axis$.

Then the points A and B are ($-x_B$, 0) and (x_B, 0) respectively. Let the time delay between the received signals at the two microphones be τ. By the figure, we can see that,

$$B'M = BM$$

$$and\, so\, AB' = \tau v$$

where $v$is the speed of sound which is considered a constant in our environment.

$$AM = AB' + B'M$$

$$\Rightarrow \quad AM = AB' + BM$$

$$\Rightarrow \quad \sqrt{(x+x_B)^2+y^2} = \tau v + \sqrt{(x-x_B)^2+y^2}$$

$$Solving\, for y,\, we\, get,$$

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

Considering a vector of points for x and calculating y for respective x, and taking the derivative of this equation,

$$\frac{dy}{dx} = tan(\alpha')$$

and from the figure, we can see that,

$$\alpha = 90 - \alpha'$$

By this, he angle α can be found.

2.2.1 Four microphone setup

With the use of two more microphones, we can handle two issues:

1. Two microphone estimation doesn't account for the semi circle that the source is present. That is, the estimation is invariant to the sign of y-coordinate of the source. The extra two microphones are required to determine it.

2. 60^o to 90^o can be estimated using the extra microphones if they are placed at the right positions.

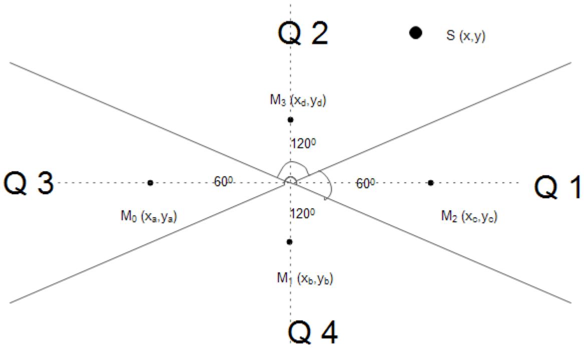


Figure 2.2: Addition of microphones

As shown in figure 2.2, $Q1$ and $Q3$ encompass 120^0out of the total circle and the remaining by $Q2$ and $Q4$. Let us consider $M_0$and $M_2$are the primary microphones. Considering the non-linear behavior of the algorithm at angles higher than 60^0 (with respect to the line M_1M_3), we use the additional two microphones $M_1$and $M_3$to find out the angle in this quadrant. For this, M_1M_3 should be on the perpendicular bisector of M_0M_2 and vice versa. The table shows the microphones used in different quadrants. Also, the additional microphones are useful to know which half circle (with respect to the line M_0M_2) the source is present. This can be done by checking the sign of the time delay between $M_0\,and\,M_2$.

Table 2.1: Quadrants and Microphones used

Quadrant	Microphone pair used to find the angle
$Q1$	M_1M_3
$Q2$	M_2M_4
$Q3$	M_1M_3
$Q4$	M_2M_4

2.3 Array Geometry and Constraints

In this section, we derive an important constraint on the geometry of a microphone array in general for a maximum frequency f_{max} of operation. Since we are using the cross correlation approach to find the time delay between received signals, we have to ensure that the phase delay between the received signals of any two microphones whose delay is of interest must fall in the $(-\pi, \pi)$ region. This is because, there can be a maximum of one time period difference between the two waves. For delays more than that, we perceive a corresponding delay in the valid range and so, get an incorrect result.

Let d be the distance between the microphones,v be the speed of sound and f_{max} be the maximum frequency of the signal that can be dealt with. Then, the maximum time delay between receptions of microphones occurs when the extra distance traveled by one of the waves is d.

Hence, for maximum frequency signal that can be viable, the limiting case will be when this distance is half wavelength of the wave (since we need both negative and positive halves).

That is, the time period of the wave is half the time delay occurred.

$$\frac{d}{v} = \frac{timeperiod}{2} = \frac{1}{2f_{max}}$$

Hence,

$$d = \frac{v}{2f_{max}}$$

This relation between microphone spacing and the maximum frequency is very important in hardware setup. Please note that only $0 - 180^0$ can be measured using two microphones.That is one half of a circle.

2.4 Simulations

The main aim of the project is to implement the angle estimation on a memory constrained module like microcontroller. Hence, to keep space and time expense as less as possible, only integer sample delays are calculated in the algorithm. It can be noted that this gives good enough results to be implemented on hardware. All these simulations are done with two microphone array with distances 5 cm, 10 cm and 15 cm and an angle sweep from 0 to 90^0. This gives us a perspective of what happens when we increase the array size.

2.4.1 Non-linearity

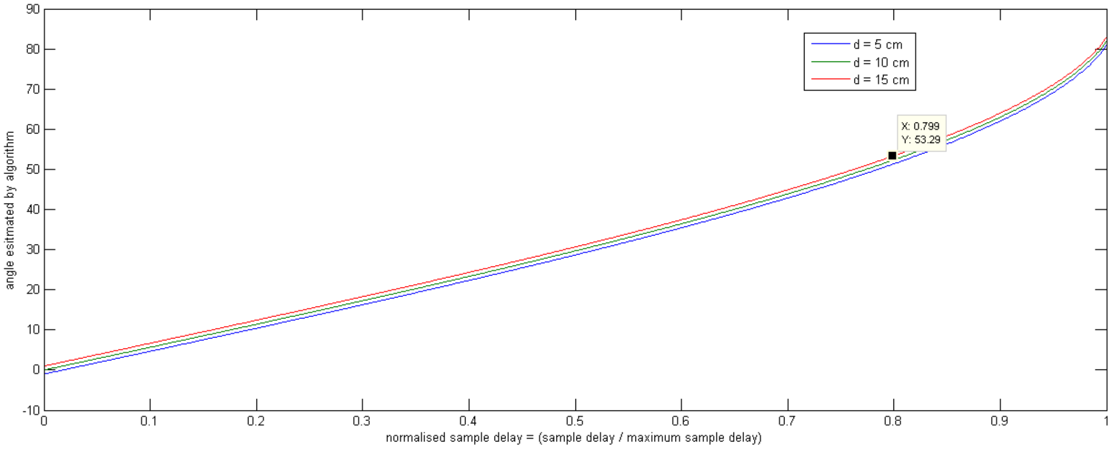


Figure 2.3: Angle vs delay in samples

As we can see from figure 2.3, the trigonometric solution is linear w.r.t. sample delay only up to about 50^0. This linearity ensures a uniform distribution of angle estimations w.r.t. sample delay. It is seen that the angle increases with normalized sample delay at approximately 6^0per 0.1. But there is an increase of angle from about $55^0$to 90^0per 0.2 variation in normalized sample delay. As the delay estimation is always prone to some error, this non linear region is less accurate than in linear region. Hence, this region has to be taken care of, which we will see later.

We also observe that the variation in angle is same w.r.t. normalized sample delay irrespective of the microphone spacing. Of course, the normalization factor(maximum sample delay possible) and the actual delay values differ in this aspect.

$maximum\, delay\, possible = (\frac{d}{v}) * F$ where,

$d = distance\, between\, microphones,$

$v = speed\, of\, sound = 340 m/sec,$

$F = Frequency\,of\,sampling = 192.3KSps\,(The\,sampling\,rate\,of\,Virtex5\,ADC)$

For $d = 5cm$, $maximum\,delay\,possible$ =$28.2794\,samples$

For $d = 5cm$, $maximum\,delay\,possible = 56.5588\,samples$

For $d = 5cm$, $maximum\,delay\,possible = 84.8382\,samples$

The non integer number of samples just indicate higher precision for accuracy in time delay calculation.

2.4.2 Sensitivity analysis w.r.t. distance between microphones

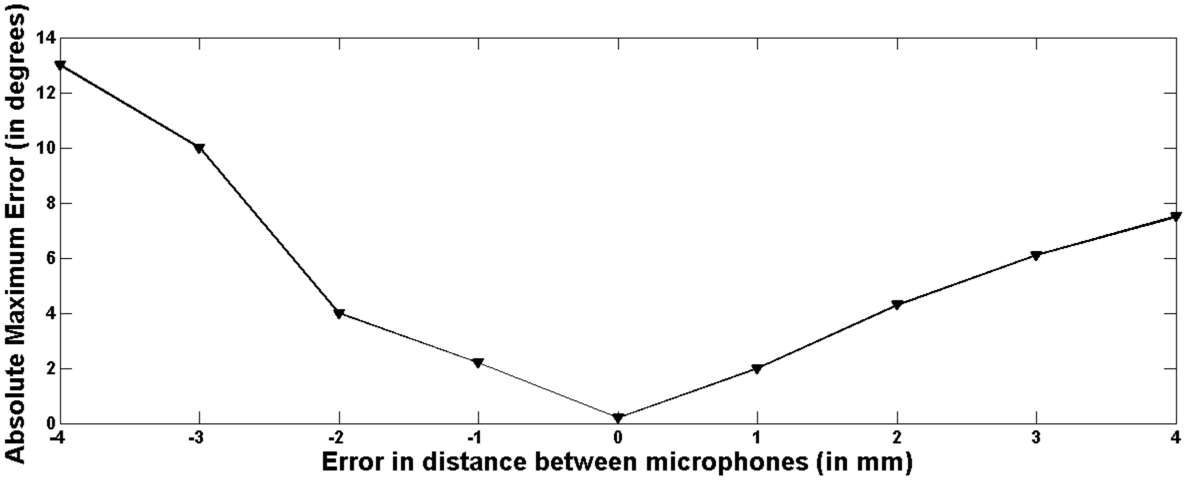


Figure 2.4: Absolute Maximum Error Vs. Error in d

Error in distance between microphones = Assumed distance (5 cm) - (unknown) actual distance.

From Figure2.4 it can be seen that the absolute maximum error in measurement increases as the error in distance between microphones increases. But it is not a symmetric figure with respect to zero error. That means, the error in estimation is higher for the same negative error in distance than that of positive error in distance.

2.4.3 Sensitivity analysis w.r.t. speed of sound

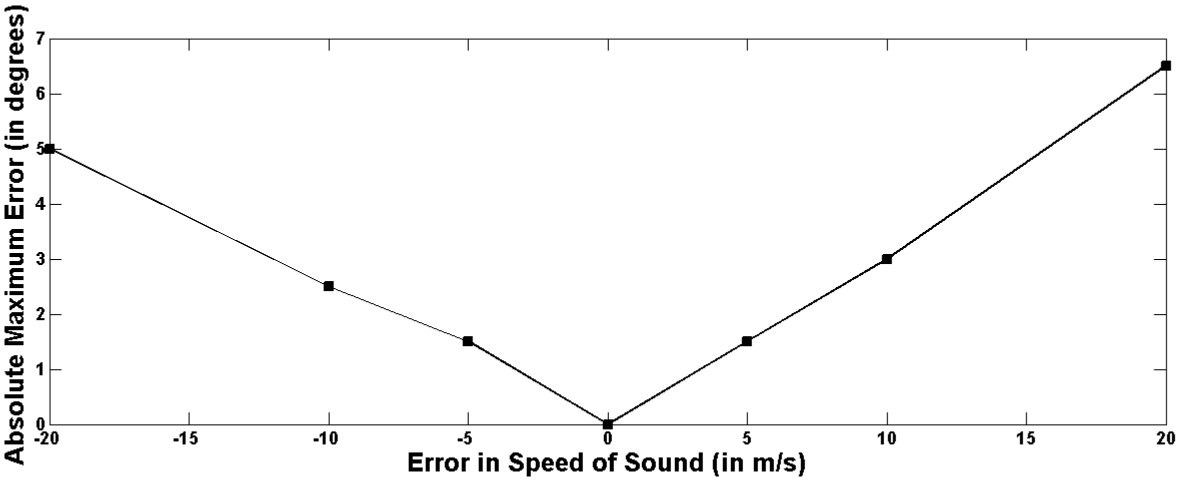


Figure 2.5: Absolute Error Vs. Error in *v*

Error in Speed of sound = Assumed speed (340 m/s) - (unknown) Actual Speed.

2.4.4 Sensitivity analysis w.r.t. Signal to Noise Ratio

Considered speech signal is given in the figure 2.6. 1024 samples are taken for simulation as this is consistent with experiment.

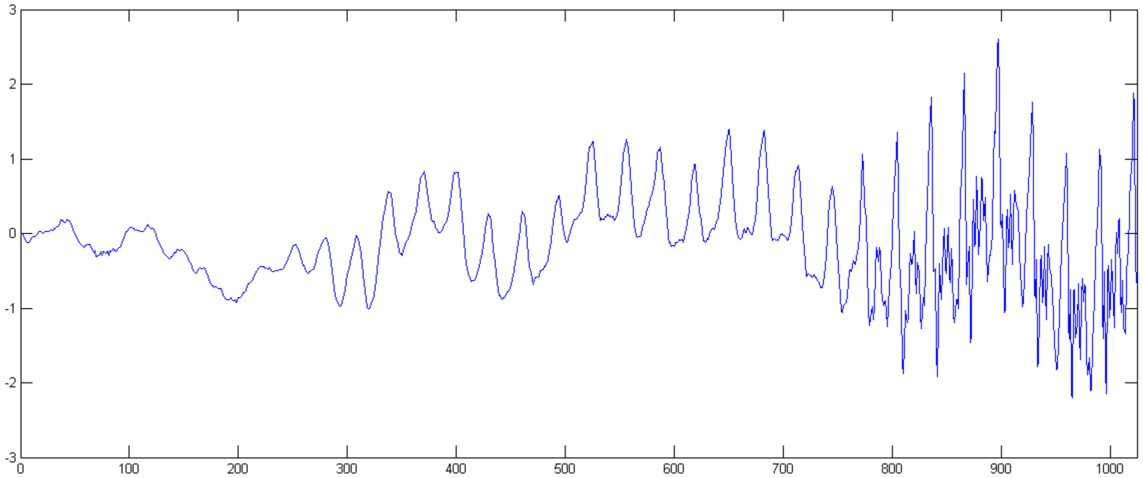


Figure 2.6: Sample speech signal from Matlab

Random noise as shown in figure 2.7 is added to this signal. Random noise is generated using MatLab function *randn*.

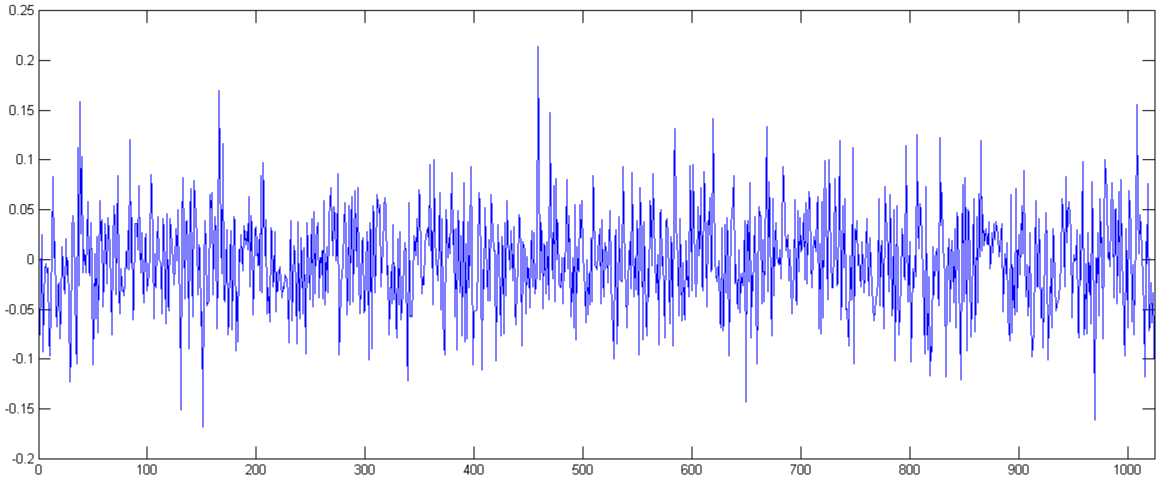


Figure 2.7: Random noise

Here, let us consider a sample speech signal. Adding noise to it with variable energies, we see where our approach ceases to give the correct result.

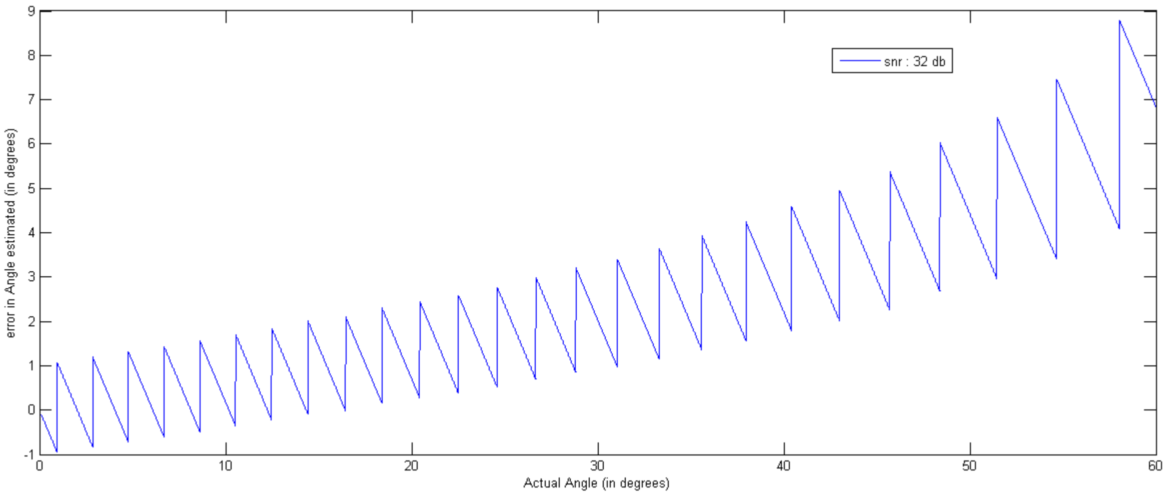


Figure 2.8: Error in estimation vs actual angle for noisy signals

Hence, we can see that the maximum error in results due to noise is around 9^0at 60^0angle.

3 Coordinates estimation

3.1 Introduction

Here, the estimation of the source is taken one step forward. From the angle estimated as discussed previously, the 2D angle of the sound source could be measured. That is, in the spherical coordinate system, ϕ is obtained. Considering the source is present in the same plane as the microphone sensors, trials are made to find the r of the source too along with ϕ. This has been simulated using the previous algorithm of Cross Correlation based technique. With the help of four microphones.

3.2 Theory

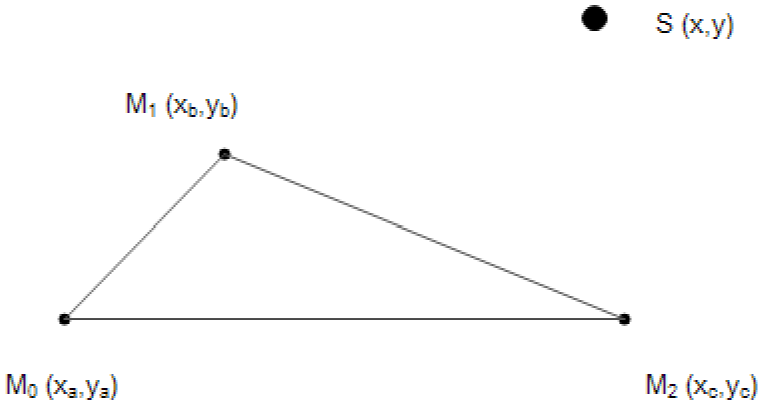


Figure 3.1: Three microphones and source

Actually, in principle, three microphones (one more than for angle as an extra parameter is measured now) are sufficient for the 2D position estimation. Let us see how this can be done and why an extra microphone is added to our setup.

Let the three microphones are located at $M_0 :(x_a, y_a)$,$M_1 :(x_b, y_b)$ and $M_2 :(x_c, y_c)$. Let the sound source is at (x, y). let us say, the signal reached M_0at t_0.And has reached $M_1$and M_2 with delays of $t_{01}$and t_{02} respectively with respect to M_0.That is , the signal has reached

$M_1$and M_2at $t_0+t_{01}$and t_0+t_{02} respectively. Let v be the speed of sound that's considered to be a constant at $340m/sec$.Then,

$$(x-x_a)^2+(y-y_a)^2=t_0^2.v^2$$

$$(x-x_b)^2+(y-y_b)^2=(t_0+t_{01})^2.v^2$$

$$(x-x_c)^2+(y-y_c)^2=(t_0+t_{02})^2.v^2$$

We don't know t_0,but we know $t_{01}$and t_{02}. Hence, we need to eliminate t_0.

$$\Rightarrow t_{01}.v=\sqrt{(x-x_b)^2+(y-y_b)^2}-\sqrt{(x-x_a)^2+(y-y_a)^2}$$

$$and\ \Rightarrow t_{02}.v=\sqrt{(x-x_c)^2+(y-y_c)^2}-\sqrt{(x-x_a)^2+(y-y_a)^2}$$

Using these two equations, solving for a relation between x and y, we get,

$$\frac{x_b^2+y_b^2}{v.t_{01}}-\frac{x_c^2+y_c^2}{v.t_{02}}-\frac{x_a^2+y_a^2}{v}(\frac{1}{t_{01}}-\frac{1}{t_{02}})-v(t_{01}-t_{02})$$

$$=x(\frac{2.x_b}{v.t_{01}}-\frac{2.x_c}{v.t_{02}}-\frac{2.x_a}{v}(\frac{1}{t_{01}}-\frac{1}{t_{02}}))+y(\frac{2.y_b}{v.t_{01}}-\frac{2.y_c}{v.t_{02}}-\frac{2.y_a}{v}(\frac{1}{t_{01}}-\frac{1}{t_{02}}))$$

But this is only one equation in two variables. Now, the angle estimation that we already derived previously is used to get the (x,y) coordinates. As the azimuthal angle $\phi$was obtained as described in section 2.2, we have,

$$\frac{y}{x}=tan(\phi)$$

This equation along with the above can be used to solve for (x,y) coordinates.

3.3 Addition of microphone

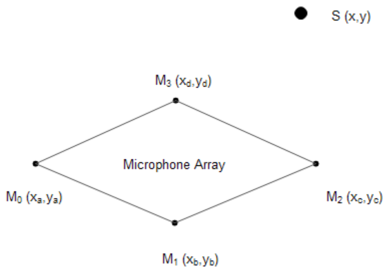


Figure 3.2: Microphones and Source Setup

As described in section 2.2.1, the additional microphone introduced in figure 3.2 is used to find out the angle ϕmore accurately. $M_1$and $M_3$are placed on the perpendicular bisector of $M_0$and M_2at equal spacing from the center so that the non linearity of angle measurement at $60^0 to 90^0$can be avoided using the measurements from other two microphones.

But this method inherently has a problem. That is : The error in angle estimation is also added up to the total measurement along with the error in measuring the individual delays. Hence, as we are using four microphones anyways, instead of going for 2D estimation, 3D position estimation can be directly carried out as follows:

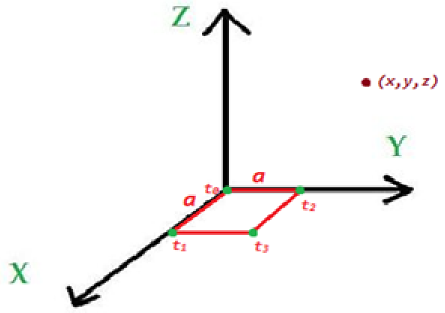


Figure 3.3: Source in 3D space

For simplicity of mathematics, let us assume the four microphones are located as $M_0(0,0,0)$,$M_1(a,0,0)$,$M_2(0,a,0)$ and $M_3(a,a,0)$. Let the sound source be located at (x,y,z). Let the signal from sound source reach the microphones M_0,M_1,M_2,M_3at times t_0,t_0+t_1,t_0+t_2,t_0+t_3 respectively. That is, the time delays of various microphones with respect to a reference microphone are observed. Then,

$$x^2 + y^2 + z^2 = v^2.t_0^2$$

$$(x-a)^2 + y^2 + z^2 = v^2.(t_0 + t_1)^2$$

$$x^2 + (y-a)^2 + z^2 = v^2.(t_0 + t_2)^2$$

$$(x-a)^2 + (y-a)^2 + z^2 = v^2.(t_0 + t_3)^2$$

Taking square root and subtracting,

$$\sqrt{x^2 + y^2 + z^2} - \sqrt{(x-a)^2 + y^2 + z^2} = -v.t_1$$

$$\sqrt{x^2 + y^2 + z^2} - \sqrt{x^2 + (y-a)^2 + z^2} = -v.t_2$$

$$\sqrt{x^2 + y^2 + z^2} - \sqrt{(x-a)^2 + (y-a)^2 + z^2} = -v.t_3$$

Taking one of the terms to RHS, squaring and on simplification, we get,

$$v^2.t_1^2 + 2.v.t_1.\sqrt{x^2 + y^2 + z^2} = -2.a.x + a^2$$

$$v^2.t_2^2 + 2.v.t_1.\sqrt{x^2 + y^2 + z^2} = -2.a.y + a^2$$

$$v^2.t_1^2 + 2.v.t_3.\sqrt{x^2 + y^2 + z^2} = -2.a.x - 2.a.y + a^2$$

From the above equations, we have,

$$\sqrt{x^2 + y^2 + z^2} = \frac{v.(t_3^2 - t_2^2 - t_1^2)}{2.(t_1 + t_2 - t_3)}$$

Hence, substituting this in above equations and solving for $x, y,\ and\, assuming\, z\, is\, positive,$

$$x = \frac{1}{2.a}[a^2 - v^2 t_1^2 - 2vt_1(\frac{v.(t_3^2 - t_2^2 - t_1^2)}{2.(t_1 + t_2 - t_3)})]$$

$$y = \frac{1}{2.a}[a^2 - v^2 t_2^2 - 2vt_2(\frac{v.(t_3^2 - t_2^2 - t_1^2)}{2.(t_1 + t_2 - t_3)})]$$

$$z = \sqrt{[\frac{v.(t_3^2 - t_2^2 - t_1^2)}{2.(t_1 + t_2 - t_3)}]^2 - x^2 - y^2}$$

3.4 Simulations

Becasue the position estimation is more sensitive to time delay error than 2D angle estimation, we decided the algorithm should have fractional sample delay measuring capacity to improve accuracy. This is an iterative algorithm [52]. Here, we first find fourier transform of correlation function by using fourier transforms of the signals. Then, its peak are calculated using parabolic interpolation.

From the tables 3.4-3.7, it can be seen that the error on line $x = y = \frac{z}{2}$is generally more than that of$x = y = z$.

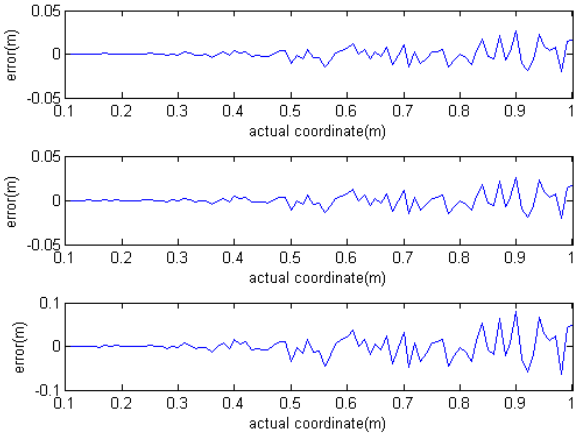
This makes sense with the extrapolation of this fact. Let us consider the line $x = y = 0$. That is, $x = y = \frac{z}{\infty}$. The error on this line will be the maximum among all lines because our microphone exists in X-Y plane and this line passes through the center. Hence, all microphones receive the signals at the same time. We have already seen from 2D angle estimation simulations that the time delay estimation is better with larger arrays with higher possible delays. Hence on this line, there will be maximum error in time delay estimation. Our mathematical derivation of coordinates is very sensitive to error in time delay estimation. Hence the result. That is why, let us consider two lines : $x = y = z$ and $x = y = \frac{z}{2}$ and simulate the error in different scenario. Also, z coordinate is a dependent variable on x and y. Hence, the errors in x and y add up in z calculation.

As previously, mtlb.mat from Matlab is used as the speech signal for simulation.Using this, the following simulations are carried out.

From table 3.2, considering the line$x = y = z$, $x, y, z \leq 1$, we have the error graph and results for all the coordinates. The results are tabulated in table 3.2.

Table 3.1: Error in Measurement on x=y=z/2 line

(a) Error graph



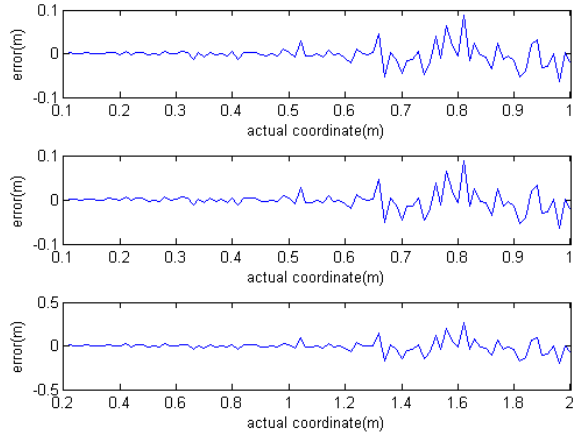
(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	2.6	90
y	2.6	90
z	7.8	90

From table 3.2, considering the line$x = y = \frac{z}{2}$, $x, y \leq 1$, we have the error graph and results for all the coordinates. The results are tabulated in table 3.2.

Table 3.2: Error in Measurement on x=y=z/2 line

(a) Error graph



(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	8.8	81
y	8.8	81
z	27.02	162

Now, let us see the sensitivity analysis on these two lines with respect to error in speed of sound, side of the square array considered considered and SNR of the signal.

3.4.1 Sensitivity analysis w.r.t. Signal to Noise Ratio

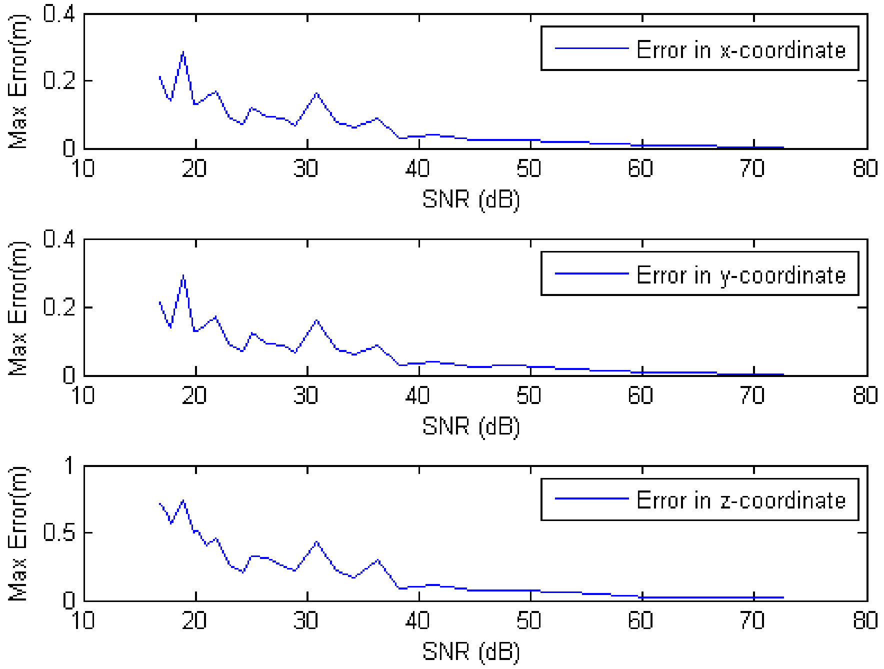


Figure 3.4: Maximum Error vs SNR on x=y=z line

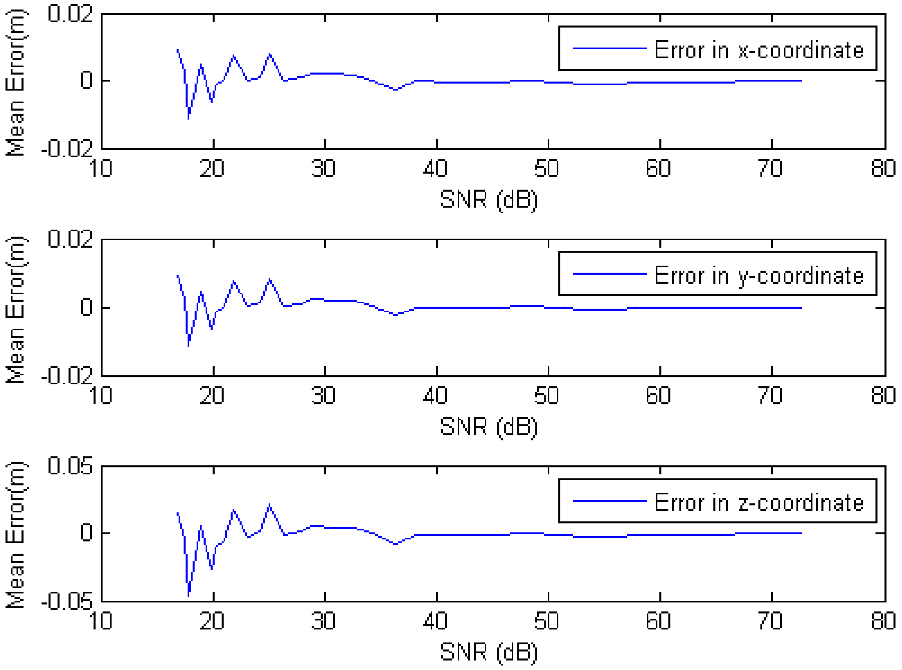


Figure 3.5: Mean Error vs SNR on x=y=z line

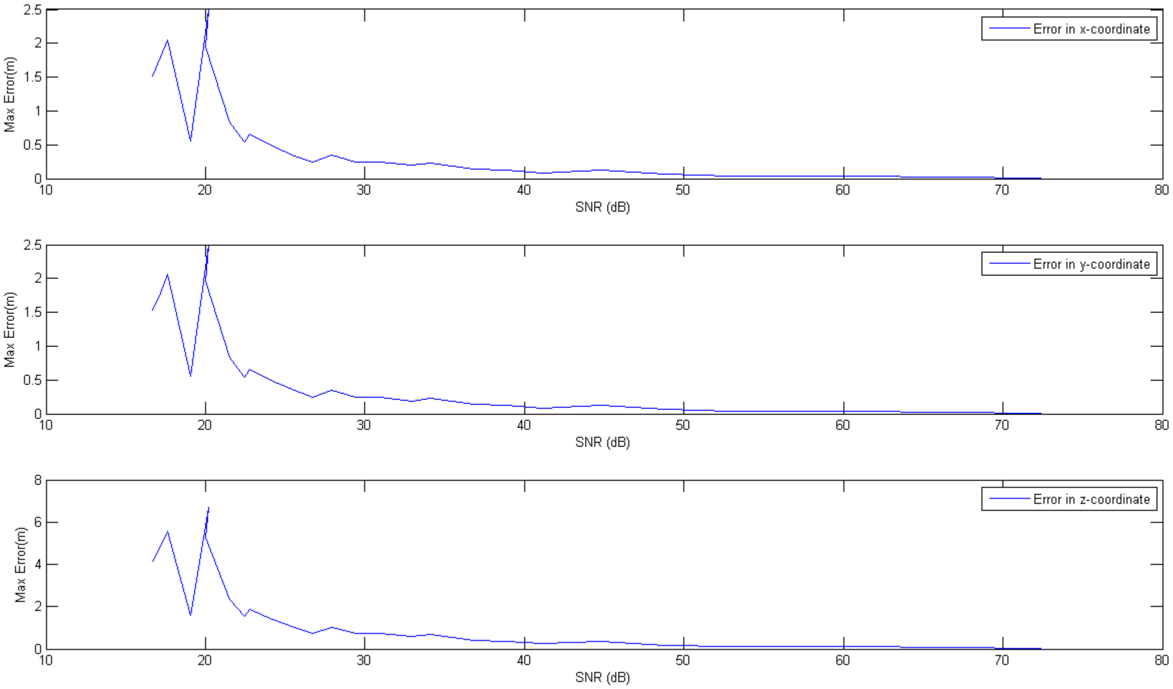


Figure 3.6: Maximum Error vs SNR on x=y=z/2 line

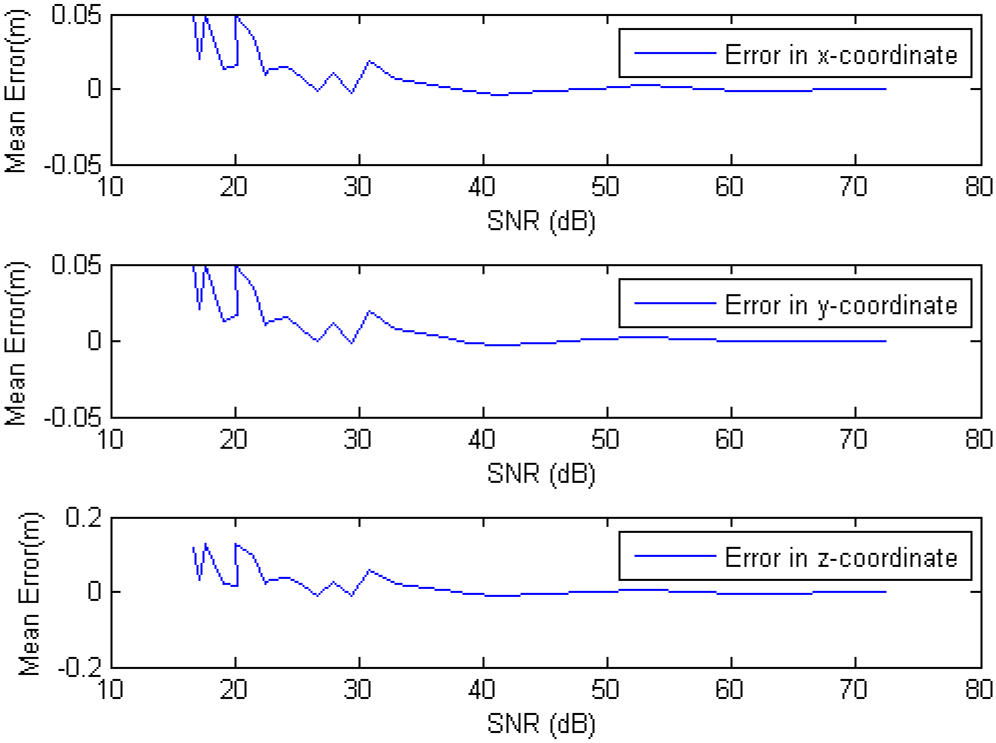


Figure 3.7: Mean Error vs SNR on x=y=z/2 line

3.4.2 Sensitivity analysis w.r.t. speed of sound

As we are considering speed of sound and distance between the microphones to be constant and already known, it is good to check the sensitivity of error with respect to these parameters.This is because, the distance between microphones can't be *exactly* the considered number. There may be a variation in the order of millimeters. Also, we know that speed of sound is a function of environmental parameters like temperature and humidity. Hence, there will be an error in considering it a constant at 340m/s. Let us see whether our approach is considerable robust for these variations or not.Let us say, we are considering the speed of sound constant at 340m/s, but the actual is at 320 m/s. That is, there is an error of 20 m/s. Let us look at the error graph then.

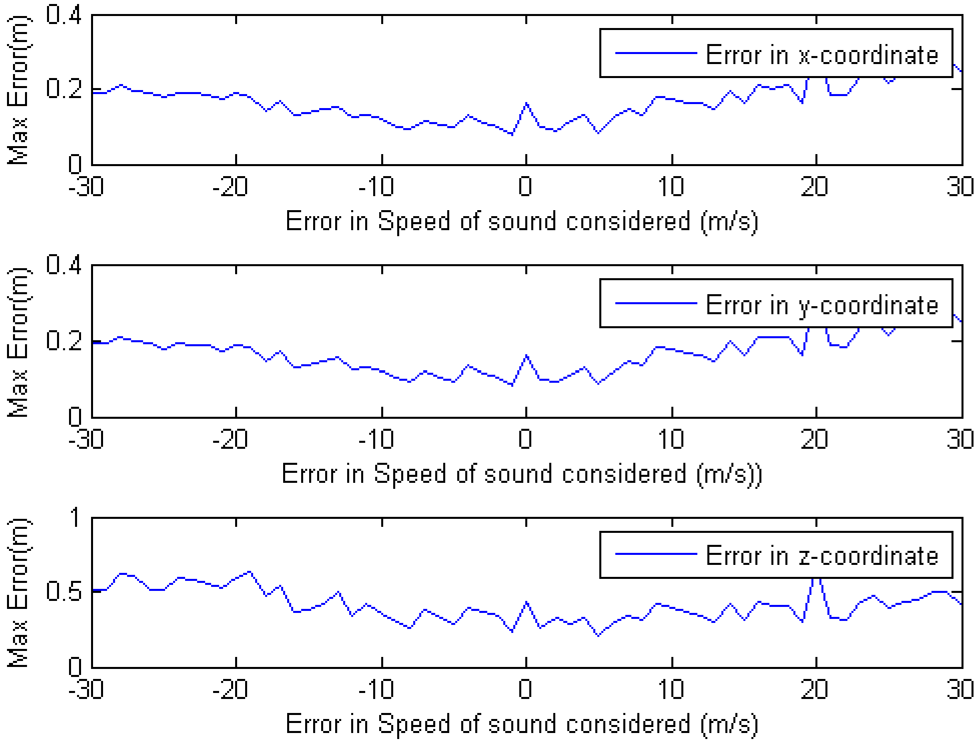


Figure 3.8: Maximum Error vs. Error in Speed of Sound

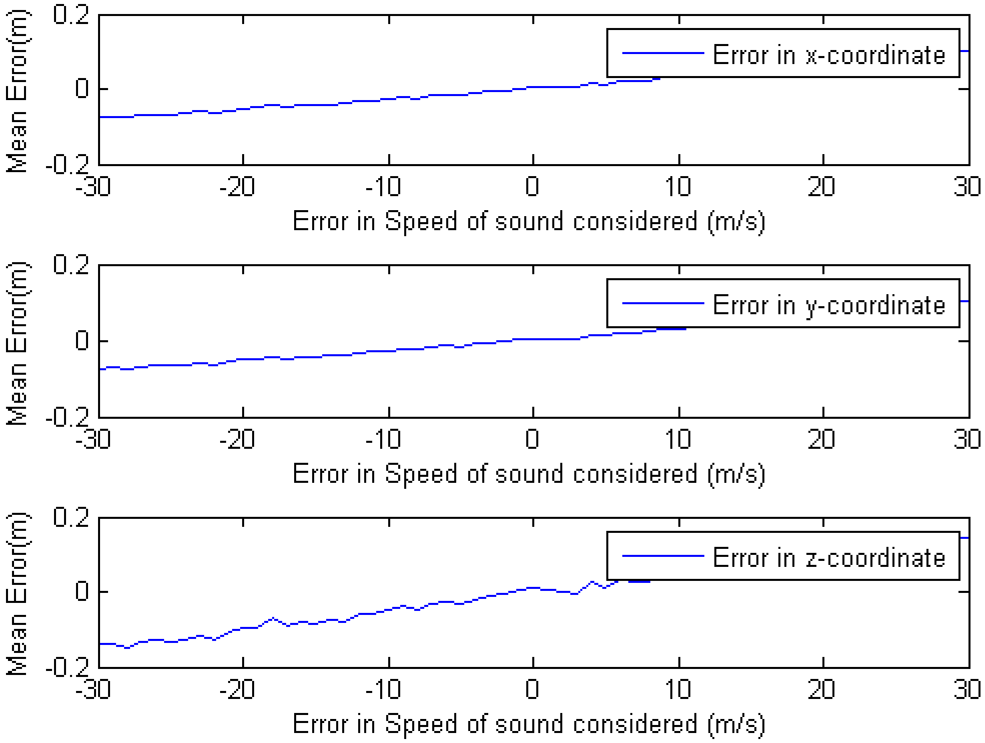


Figure 3.9: Mean Error vs Error in Speed of Sound

3.4.3 Sensitivity analysis w.r.t. array geometry

As we have already described, there may be an error in taking the distance between microphones. Hence, lets see the sensitivity of measurement to this error. As the distance is measures in cm, lets consider an error of -10 and 10 mm with respect to the actual distance.We present the graphs of error in angle, separately for positive error and negative error in distance.

Error in $d = Considered\, distance\, in\, algorithm - actual\, distance$

Let us say, we considered the square side, a as 10 cm. But it is indeed 9 cm. So there is an error of 1 cm.

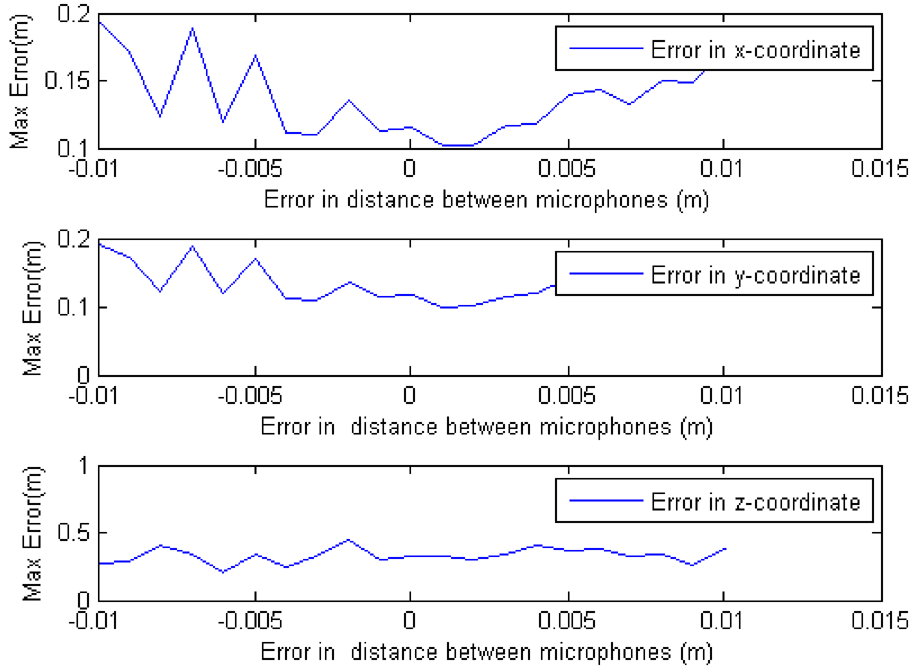


Figure 3.10: Maximum Error vs Error in distance between microphones on x=y=z line

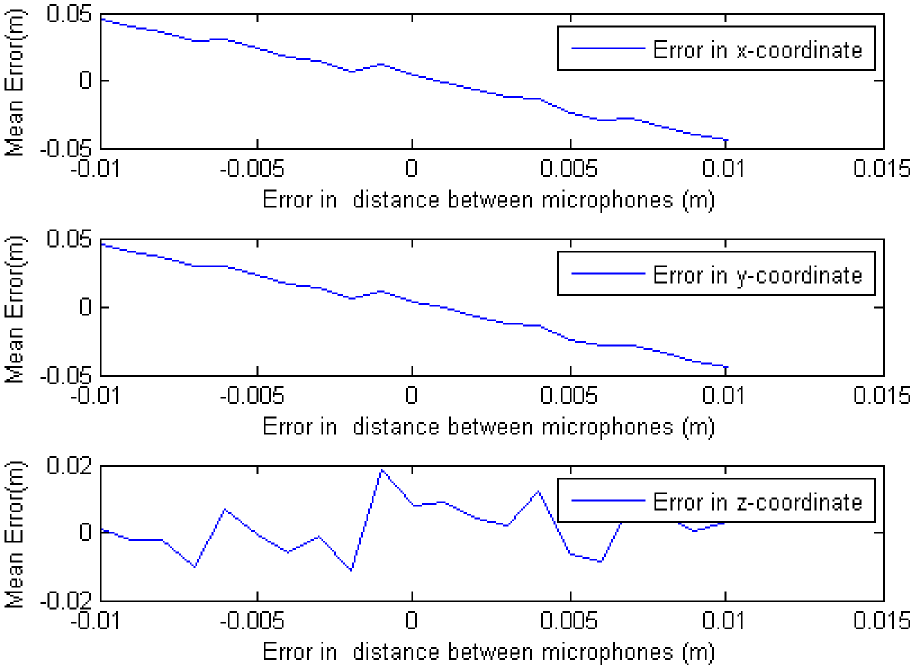


Figure 3.11: Mean Error vs Error in distance between microphones on x=y=z line

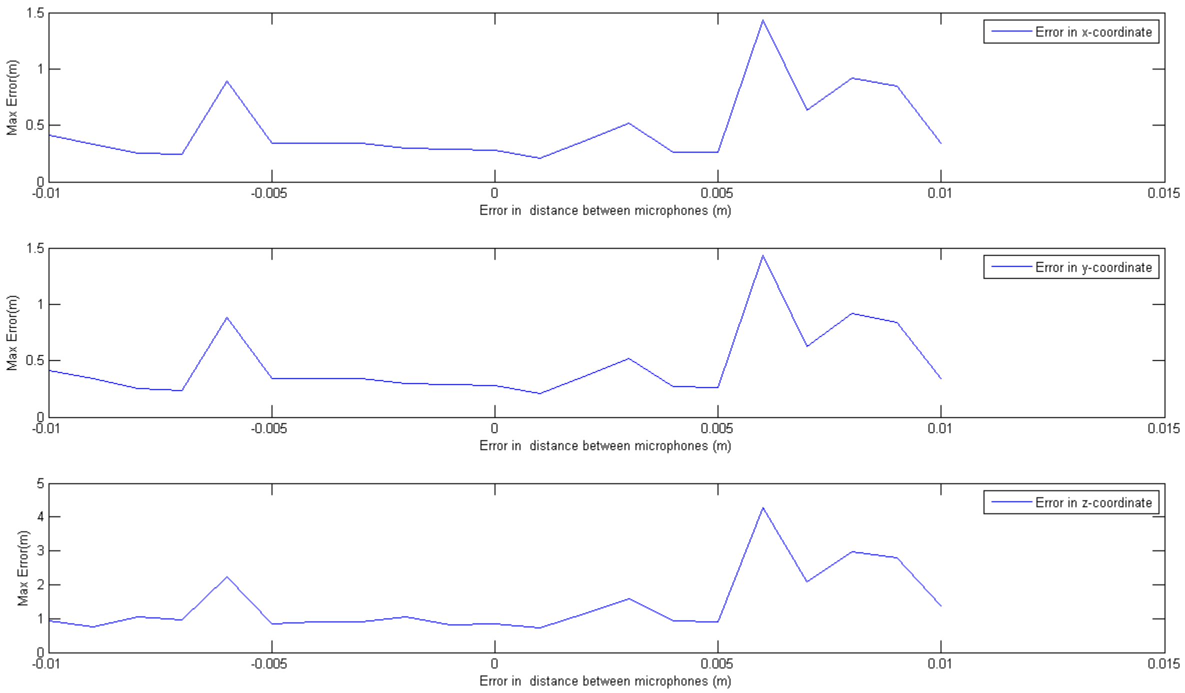


Figure 3.12: Maximum Error vs Error in distance between microphones on x=y=z/2 line

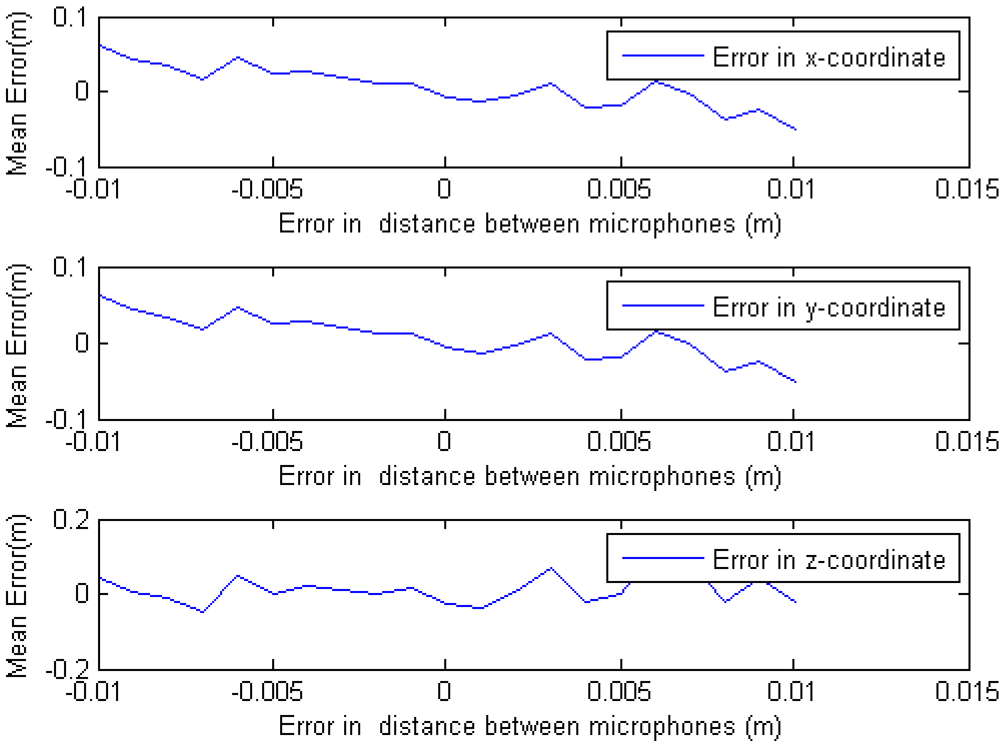


Figure 3.13: Mean Error vs Error in distance between microphones on x=y=z/2 line

3.5 Microphone array with more than four microphones

To support real-time or close to real-time processing, closed form algorithms are desired. Methods that include, and make use of, redundancy in sensor information perform better in cases with measurement error. Time difference of arrival measurements can be modeled and expressed in a least squares form. The Least Squares Estimator (LSE) can be used to minimize the squared error between the model and the measured data. With an accurate model, in the absence of noise, the square error function should be zero. The form of the error criteria derived from the source localization problem can effect the computational complexity of finding a solution.

Chan and Ho [20] suggested a least squares estimator to make use of the redundancy in the measurements. Their approach takes advantage of the relation of the source coordinates to the range to improve the estimation efficiency using quadratic correction. This is termed the Quadratic-Correction Least-Squares (QCLS) method.

Following the derivations of Chan and Ho [20], Hui and So [43], and Huang and Benesty [37] the problem can be stated as follows. Let the positions of the N +1 microphones, in Cartesian coordinates be denoted by

$$r_i = [x_i \, y_i \, z_i]^T \quad i = 0, 1, 2, 3...N$$

where $(.)^T$ denotes the transpose of a vector. The first microphone is set to be the origin, $r_0 = [0,0,0]^T$ and the source coordinates sought are $r_s = [x_s, y_s, z_s]^T$. The range between the i-th microphone and the source is given by

$$D_i = \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2 + (z_s - z_i)^2}$$

The corresponding range differences, for microphones i and j in a pair, to the source are given by

$$D_{ij} = D_i - D_j \quad i, j = 0, 1, 2....N$$

This is related, by the speed of sound v, to the time difference of arrival, τ_{ij} , measured in synchronized microphone arrays by

$$d_{ij} = v\tau_{ij}$$

It is noted [37], that there are $\frac{(N+1)N}{2}$ distinct delay measurements, τ_{ij}, excluding the i = j case and repetition arising from $\tau_{ij} = -\tau_{ji}$, but any N linearly independent v_{ij} values determine all the others in the absence of noise. For simplicity, at the cost of improved accuracy in noisy systems, the N time. The range differences measured between connected microphones, d_{i0} are modeled as the actual value and an additive noise term ε_i, assumed to be zero-mean:

$$\hat{d_{io}} = d_{i0} + \varepsilon_i \quad i = 1, 2, ...N$$

where,

$$d_{i0} = \|r_s - r_i\| - \|r_s\|$$

Hence, we have,

$$\hat{d_{i0}} = \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2 + (z_s - z_i)^2} - \sqrt{(x_s)^2 + (y_s) + (z_s)} \qquad i = 1, 2, ...N \quad \{A\}$$

The solution to this set of hyperbolic functions is nonlinear and sensitive to measurement noise. Instead, a spherical error function can be formulated.

The distances from r_0 at the origin to the remaining microphones and to the source are denoted by $R_i \textit{ and } R_s$ respectively, where

$$R_i = \|r_i\| = \sqrt{(x_i)^2 + (y_i) + (z_i)} \;\; i = 1, 2, ...N$$

$$R_s = \|r_s\| = \sqrt{(x_s)^2 + (y_s) + (z_s)} \;\; i = 1, 2, ...N$$

Squaring both sides of the equation {A} and substituting in Rs as an intermediate variable yields a set of linear, spherical signal model, equations:

$$r_i^T r_s + d_{i0} R_s = \frac{1}{2}(R_i^2 - d_{i0}^2) \quad = 1, 2....N$$

Writing this spherical least squares error function in vector form,

$$G\theta = h,$$

where,

$$G = [S|\hat{d}], \quad S = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ . & . & . \\ . & . & . \\ x_N & y_N & z_N \end{bmatrix}$$

$$\theta = \begin{bmatrix} x_s \\ y_s \\ z_s \\ R_s \end{bmatrix}, \; h = \frac{1}{2} \begin{bmatrix} R_1^2 - \hat{d}_{10}^2 \\ R_1^2 - \hat{d}_{20}^2 \\ . \\ . \\ R_1^2 - \hat{d}_{N0}^2 \end{bmatrix},$$

$$\hat{d} = [\hat{d}_{10}\,\hat{d}_{20}\ldots\ldots\hat{d}_{N0}]^T$$

Solving the corresponding least squares criterion is a linear minimization problem

$$min_\theta (G\theta - h)^T (G\theta - h)$$

subject to the quadratic constraint

$$\theta^T \Sigma \theta = 0$$

where $\Sigma = diag(1,1,1-1)$ is a diagonal and orthogonal matrix.The technique of Lagrange multipliers can be used yielding the constrained least squares estimate:

$$\hat{\theta} = (G^T G + \lambda\Sigma)^{-1} G^T h,$$

where λ is still to be found. An unconstrained spherical least squares estimator can be derived by not enforcing the quadratic constraint which is equivalent to assuming that the source coordinates and distance to the origin x_s, y_s, z_s, and R_s are mutually independent. An estimate of θ is then given by

$$\hat{\theta}_1 = G^{\urcorner} h,$$

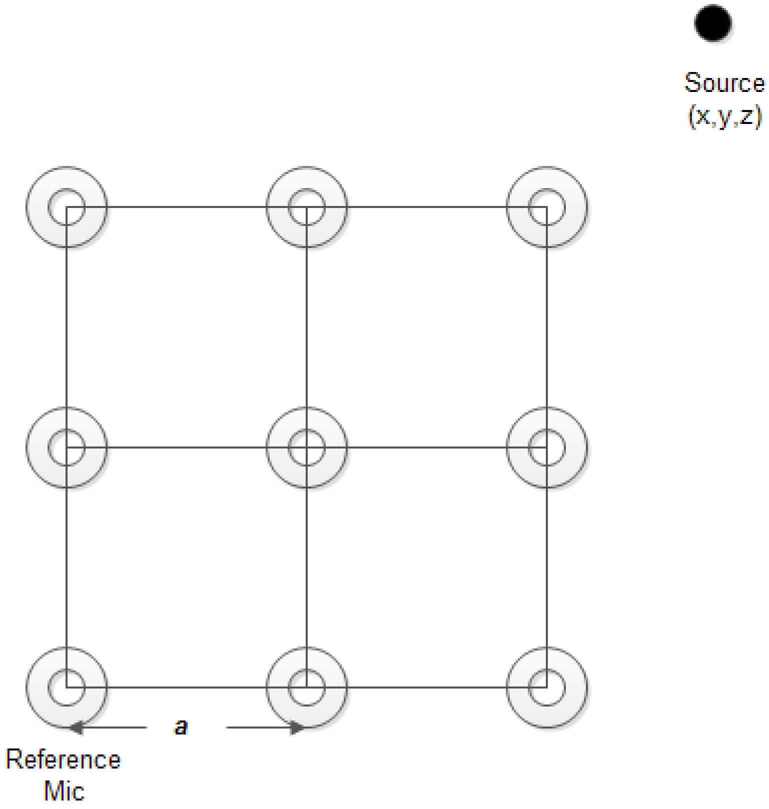
where

$$G^{\urcorner} = (G^T G)^{-1} G^T$$

is the pseudo-inverse of the matrix G. We have used this solution to find out the location in 16 microphone setup simulations.

3.5.1 Simulations with Nine microphones

Figure 3.14: Nine Microphone setup



As shown in Figure 3.14 , let us consider a Nine- microphone array in X-Y plane and a source in space. Let one of the microphones is a reference Mic. That means, the time delays at all the other receivers will be calculated with respect to that Mic and used for the coordinates estimation. Let a is the distance between adjacent microphones as shown in the figure, and it is the same between any two microphones.

Let us compare the error in this case to the previous Four microphone case with respect to decrease in SNR. As we have already seen that the error decreases with increase in side length of array, let us take a pessimistic distance of 5 cm to look at the results. So, $a = 5\,cm$.

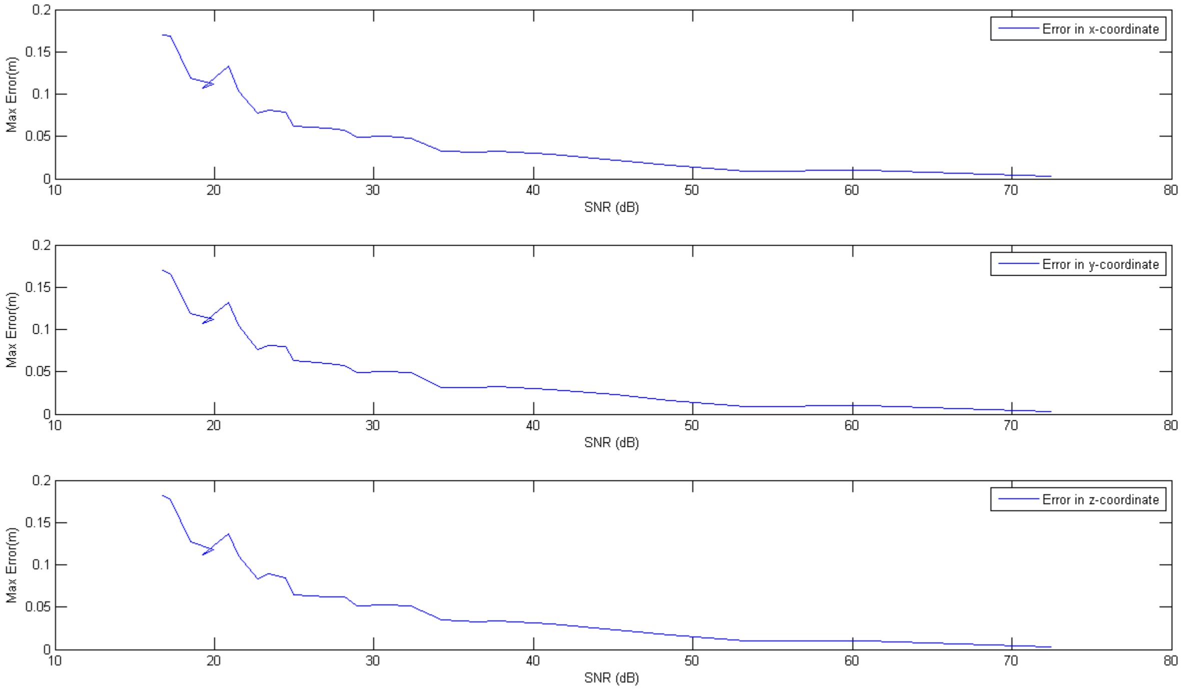


Figure 3.15: Maximum error w.r.t. SNR on x=y=z line

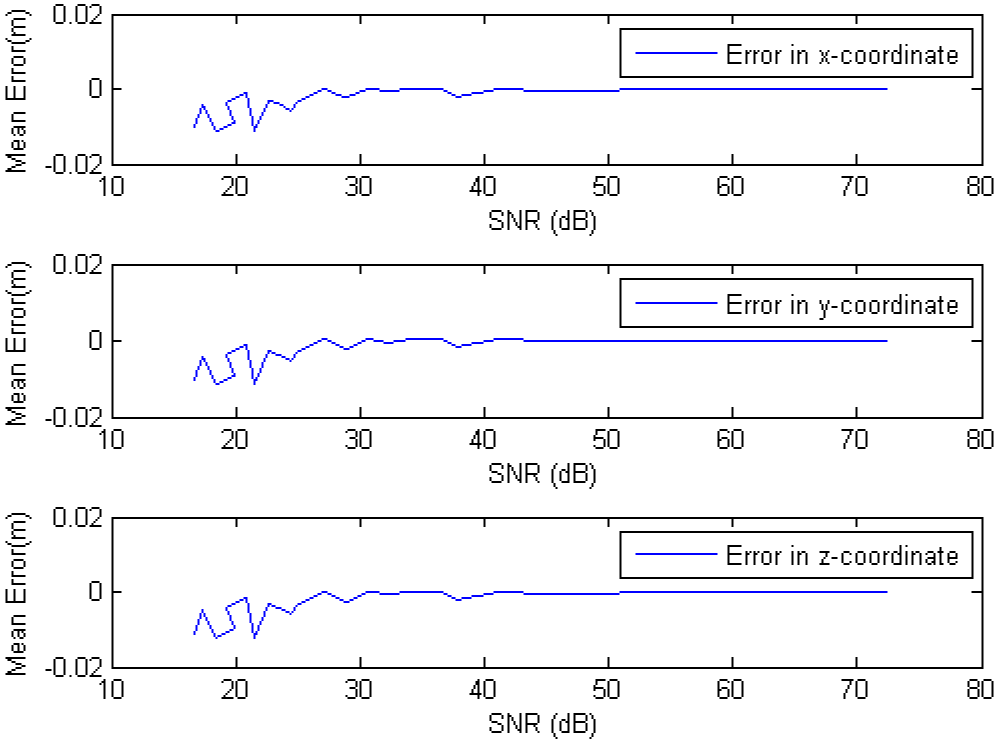


Figure 3.16: Mean error w.r.t. SNR on x=y=z line

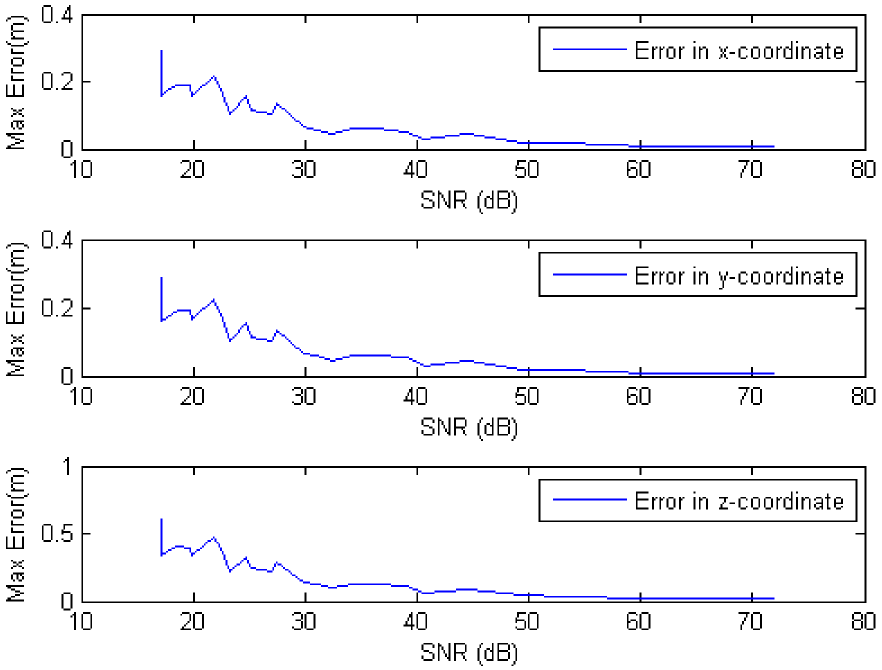


Figure 3.17: Maximum error w.r.t. SNR on x=y=z/2 line

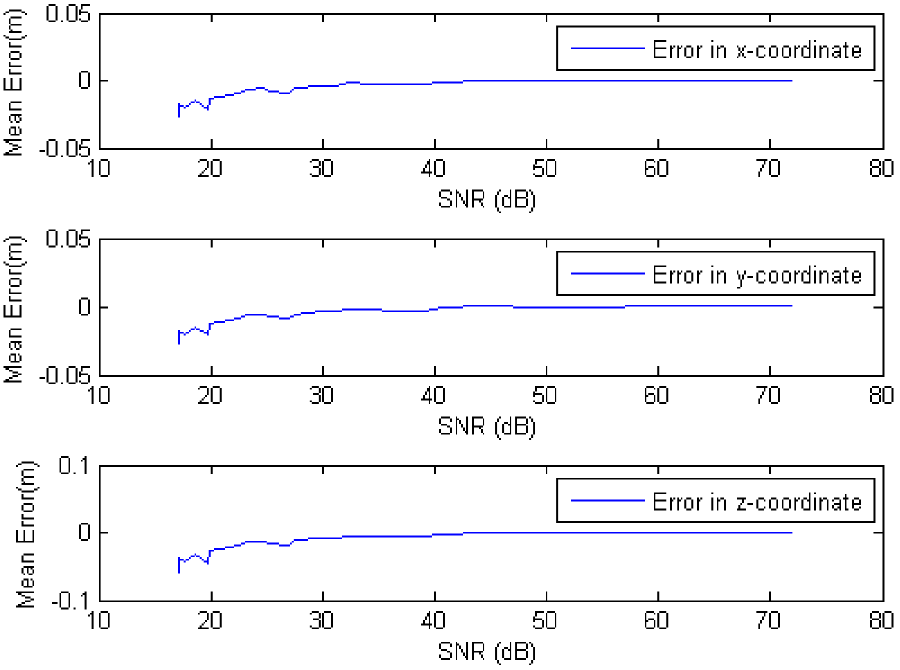


Figure 3.18: Mean error w.r.t. SNR on x=y=z/2 line

4 Experimental Setup

4.1 Selection of Microphones

We have selected ' Electret microphone ' for the experiment. Let us see a little background on electret microphone.

An Electret Condenser Microphone (ECM) consists of a very light diaphragm (moving plate) and back plate (stationary or static plate) and has a permanent charge implanted in an electret material to provide polarizing voltage. The principle of operation is that sound waves impinging on the diaphragm cause the capacitance between it and the back plate to change synchronously, this in turn induces an AC voltage on the back plate [46].

Some of the features of this microphone are [46]:

- Sensitivity - Usually expressed in V/Pa (V /10 u bar) at 1 kHz, sensitivity is the output voltage measured when a sound wave is detected by the microphone in a specified load condition. It is expressed with a specified resistive load and supply voltage since the output resistance tend s towards constant current characteristics.

$$Sensitivity(dB) = 20log(V/Pa)$$

For Electret microphone,

$$Sensitivity = -35 \pm 4db\,(0db = 1\frac{V}{Pa}, 1KHz)$$

- Directivity - Directivity is a characteristic of sensitivity according to the infliction angles of sound waves in the microphone.

  1. Omni directional - The directional property that has identical sensitivities originating from all directions.

  2. Bi-directional - The directional property that has identical frontal sensitivity and rear sensitivity values while side sensitivities are weaker. Also referred to as "cross-talking" and "noise canceling." (Noise canceling, cross-talking used in a circum-

stance of loud noise, eliminates the sound coming from a long distance and picks up the sound coming from only a short distance. Used in cellular phones and headset microphones.)

3. Unidirectional - The directional property that has the rear sensitivity relatively weaker than the frontal sensitivity.

The chosen microphones are $Omni\,directional$.

- Output Impedance - The effective output resistance is determined mainly by the value of load resistance. It can be made higher or lower by the value of load resistance with a corresponding change in sensitivity. To get the maximum output, the load should be matched to the output resistance.

  In our case, the microphones are of $low\,impedence$ ($\leq 2.2K\Omega$).

- Frequency Response - Frequency response is the microphone's sensitivity performance in the frequency range of 0 to 20 kHz. Compared with the dynamic types, ECM's tend to have an extended response both at low and high frequencies, which is also smoother.In the case of dynamic types, the response limits are defined where the sensitivity has fallen by 3 dB relative to its value at 1 kHz. It is not stated in such terms for ECM's since they have a much wider frequency response.

  ECM's work in the range of $20Hz\ to\ 20KHz$. The following is the frequency response curve [47] :

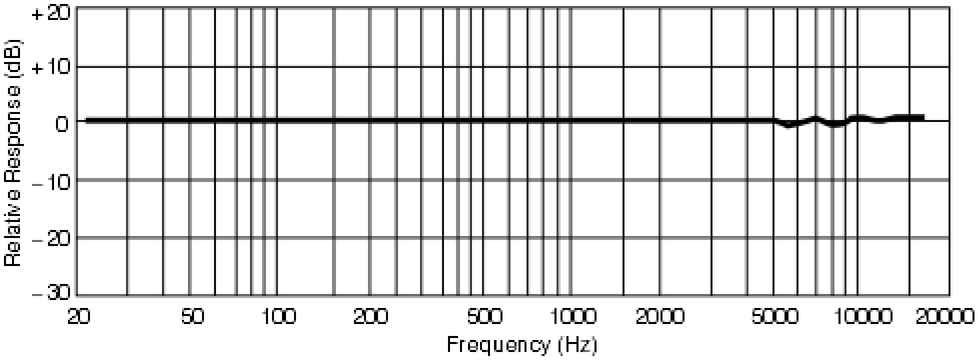


Figure 4.1: Typical Frequency Response Curve*
*figure taken from Electret Microphone data sheet

- Signal to Noise Ratio - Signal to noise ratio is the ratio of the straight frontal sensitivity of the microphone to the proper noise voltage. Proper noise is the noise in the microphone which is not produced by the action of electromagnetic induction, vibration or natural wind. It is the noise produced by the microphone itself. This ratio gives an idea of how well the signal is detected by the microphone.

$$S(dB) = 20log(V_S/V_N)$$

Where: V_S = Signal expressed in volts andV_N = Noise expressed in volts.

In this case, the EMC's have an SNR ratio $more\ than$ 62 db.

4.2 Data Acquisition

The output from the microphone is too low to be used directly for sampling. The electret microphone needs an amplifier before connecting to ADC.We have used LM324 as the op-amp IC and the circuit in figure 4.2 is the circuit used for amplification.

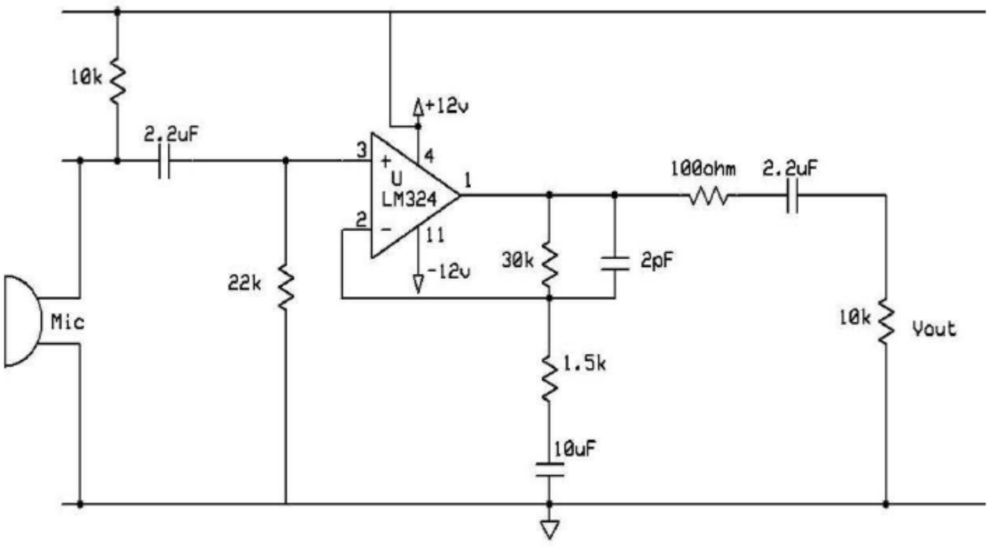


Figure 4.2: Amplifier Circuit

The output of amplification will be -0.5 V to 0.5 V peak to peak wave. We use DC coupling to make it between 0 to 1 V.

4.2.1 Virtex 5 FPGA based development board

We have used Virtex 5 FPGA based development board from Xilinx ML50*x* series for data acquisition because it has 16 channel ADC[48]. The ML50x supports both the dedicated and the auxiliary analog inputs to the Virtex-5 FPGA System Monitor block. There are 16 user-selectable auxiliary analog input channels. The Virtex-5 FPGA System Monitor function is built around a 10-bit, 200-kSPS (kilo samples per second) Analog-to-Digital Converter (ADC). When combined with a number of on-chip sensors, the ADC is used to measure FPGA physical operating parameters like on-chip power supply voltages and die temperatures.

The System Monitor is fully functional on power up, and measurement data can be accessed via the JTAG port pre-configuration. The Xilinx ChipScope™ Pro tool provides access to the System Monitor over the JTAG port. The System Monitor control logic implements some common monitoring features. For example, an automatic channel sequencer allows a user-defined selection of parameters to be automatically monitored, and user-programmable averaging is enabled to ensure robust noise-free measurements.

We have used the ADC in single ended unipolar mode rather than differential mode as our input is always positive with respect to GND.When unipolar operation is enabled, the differential analog inputs (V_P–V_N) have an input range of 0V to 1.0V. In this mode, the voltage on V_P (measured with respect to V_N) must always be positive [49]. Figure 4 shows a typical application of unipolar mode. The V_N input is typically connected to a local ground or common mode signal. The common mode signal on V_N can vary from 0V to +0.5V (measured with respect to ground). Because the input range is from 0V to 1.0V (V_P *to* V_N), the maximum signal on V_P is 1.5V.

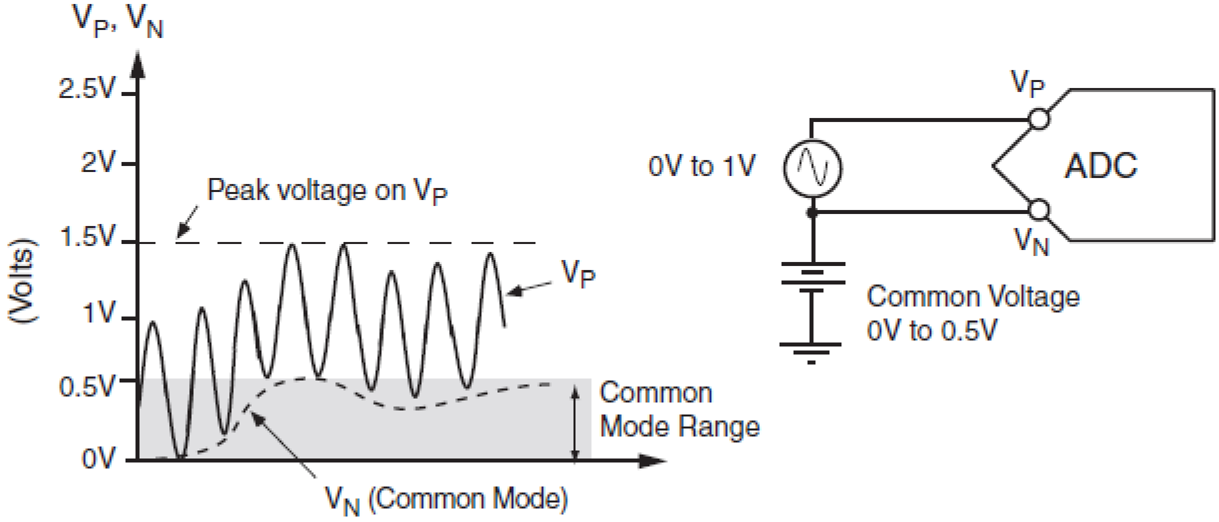


Figure 4.3: Unipolar Mode of sampling in Virtex 5*
*figure taken from Virtex-5 FPGA System Monitor User Guide from Xilinx

The ADC output coding in unipolar mode is straight binary. The designed code transitions occur at successive integer LSB values such as 1 LSB, 2 LSBs, and 3 LSBs (and so on). The LSB size in volts is equal to 1V/210 or 1V/1024 = 0.977 mV. The ideal transfer function is illustrated in Figure 5.

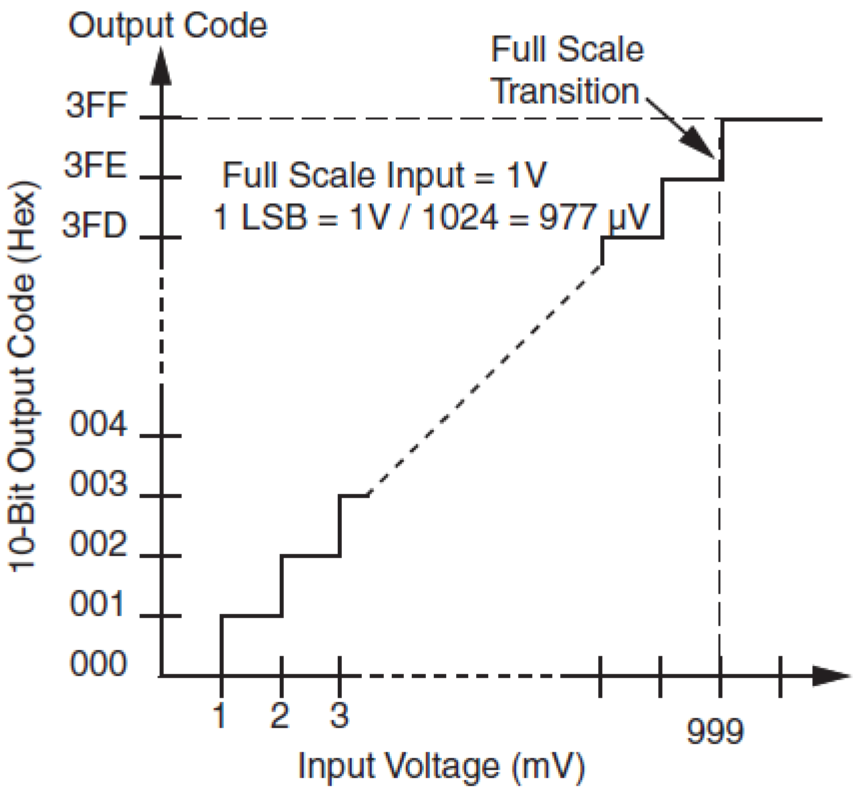


Figure 4.4: Transfer Function of ADC in Virtex 5*
*figure taken from Virtex-5 FPGA System Monitor User Guide from Xilinx

Then, the sampled signals are fed to the computer using ChipScope Pro tool.

4.2.2 Stellaris Microcontroller

The ADC available in Stellaris offers you with following Features

1. 10-Bit conversion Resolution.

2. 16 shared analog input channels.

3. Single-ended and Differential-input configurations.

4. Maximum sample rate of one million samples/second.

5. Four programmable sample conversion sequencers with corresponding conversion result FIFOs.

6. Efficient transfers using Micro Direct Memory Access Controller (μDMA).

The Stellaris Micro controller have two identical Analog-to-Digital Converter modules called ADC0 and ADC1. These two ADC s module share the same 16 analog input channels.

Each ADC module can

1. Operate Independently.
2. Execute different sample sequences.
3. Any time they can sample any of the analog input channels.
4. Can generate different interrupts and triggers &
5. Can have independent µDMA channels.

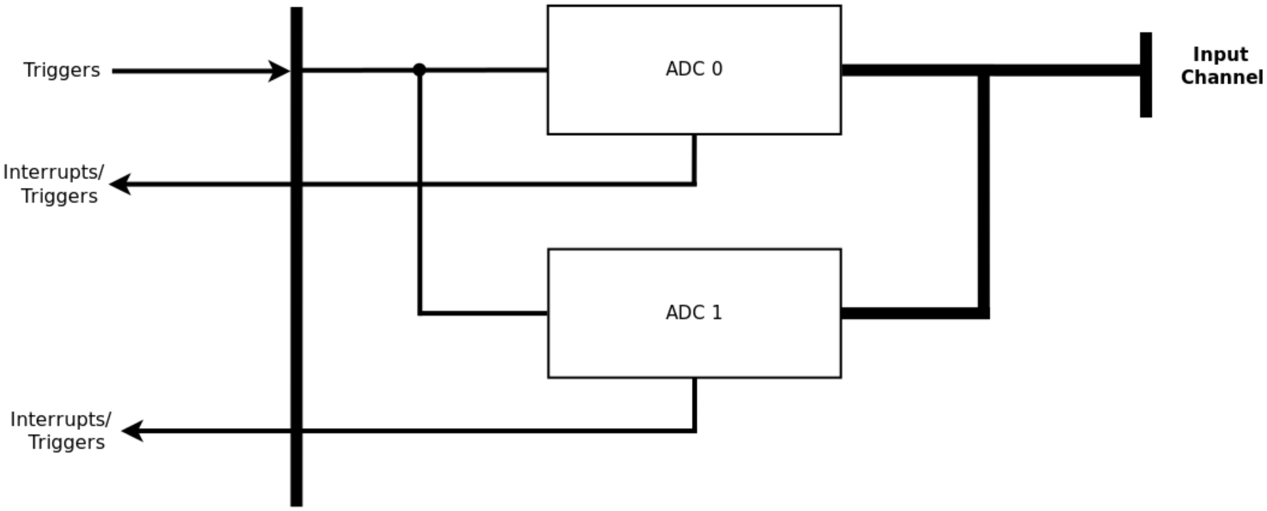


Figure 4.5: Implementation of Two ADC Blocks in Stellaris

ADC s which are available in Stellaris collects sample data by using a programmable sequence-based approach. ADC module has four sampling sequencers of depth 1,4,8 which allows you to sample different analog sources with a single-trigger event.

4.3 Sample Sequencers

We have four sample sequencers which are completely independent of each other. Each sample sequencer has its own set of configuration registers as shown below.

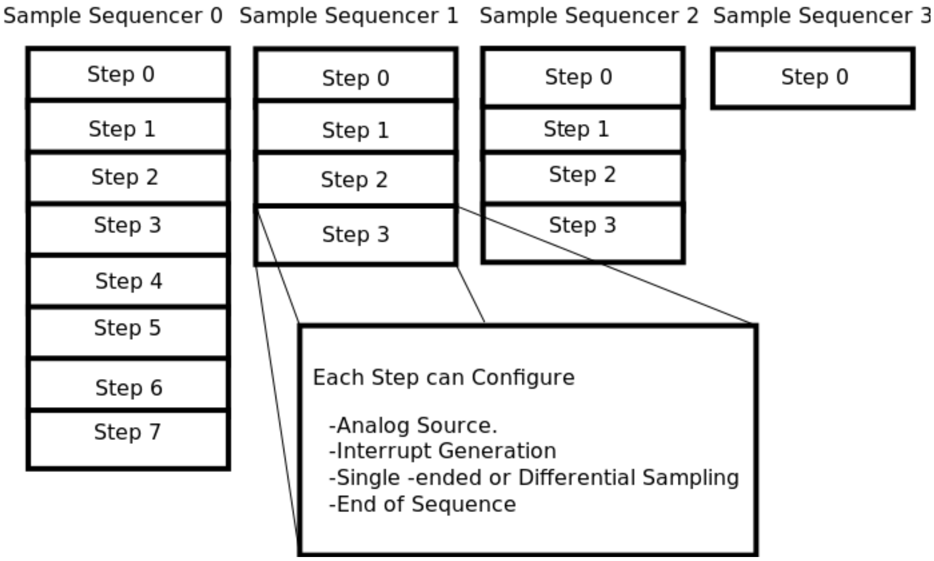


Figure 4.6: Sample Sequencer Structure in Stellaris

We can see from the above figure all the steps in the sample sequencer are configurable allowing you to select different analog sources like temperature sensor or the signal itself. Each sampler sequence has its own FIFO depth and can collect those many number of samples.

Table 4.1: Implementation of Two ADC Blocks in Stellaris

Sequencer	Number of Samples	Depth of FIFO
SS3	1	1
SS2	4	4
SS1	4	4
SS0	8	8

The above figure shows the maximum no of samples that each sequencer can capture and its corresponding FIFO depth. The number of samples an ADC can capture from each sample sequencer is also programmable by having an END bit set. It also give you flexibility to handle multiple sample sequencers which are triggered simultaneously by giving them configurable priority.

When a sample sequencer is triggered following operations are performed

- It samples the signal at the programmed sampling rates say 250K, 500K, 1M samples/sec.

- Sampling continues till it encounters the END (END bit can be set for any step in sequence) bit set, indicating end of the sequencer.
- Collect the converted result in the sampler sequencer FIFO.

Each FIFO entry is a 32-bit word, with the lower 10-bits containing the conversion result.

5 Results

5.1 On CPU (using Virtex 5 FPGA for acquisition)

5.1.1 Constants and Assumptions

8192 samples are taken per channel at a particular instant using ChipScope Pro. The data is written to a file and Matlab imports it automatically to process and give the results. So this is partial real time processing.

- Sampling Frequency is 192300 samples/sec.
- Speed of sound is assumed to be constant at 340 m/s.

The angle estimation has been checked for different microphone pair spacings and also different frequencies. The simulated and really obtained angles w.r.t. delays are presented here. We can see that, the way of obtaining the angle inherently has a disadvantage, that is, there will be steep rise in the estimation after 50 degrees w.r.t. delay. That is, the angle vs. delay curve becomes non linear. This can be noted from both the simulated and real time curves.

We can tackle this issue later by using another pair of microphones, which anyway has to be used for 3D position estimation.

5.1.2 Results

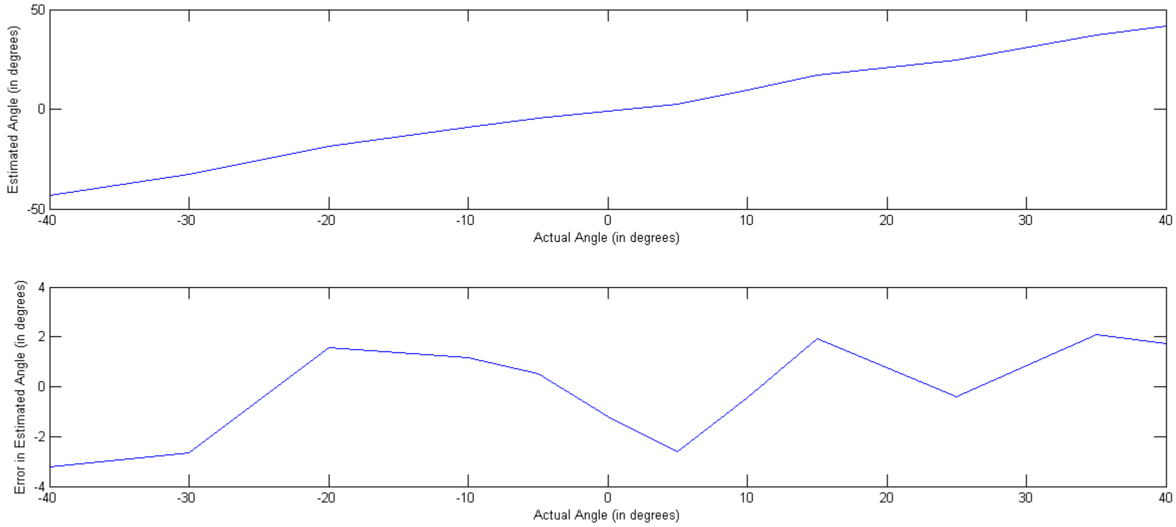


Figure 5.1: Estimated Angle Results

From figure 5.1 , it can be seen that the maximum error in estimation is around 4^0.

The real time data is obtained using 1200 Hz sine wave and moving the source slowly around the microphone pair from 0 to 90 degree. Then it is plotted along with the simulated data.

From the figure 5.2, we see the non linearity of the simulated curve and also the curve with real data. This figure doesn't account for errors in the angle estimation. It just shows that the angles more than about $60^o$are not properly estimated.

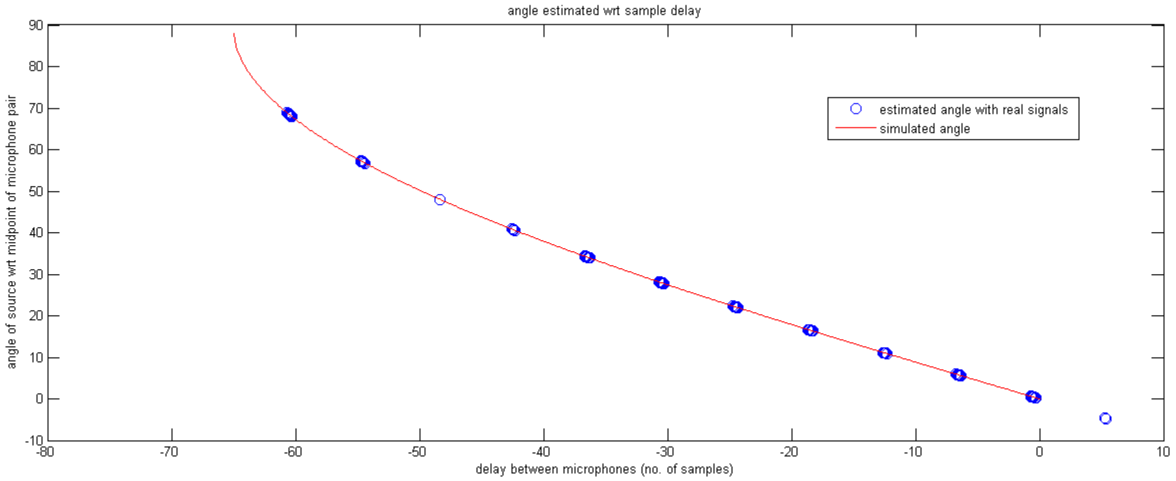


Figure 5.2: 11.5 cm 800 Hz : Angle vs Delay

5.2 Using Stellaris Microcontroller

5.2.1 Some important points

Using 1200 Hz sine wave at 55000 sampling rate for a 8.5 cm microphone array, the maximum sample delay between the microphones can be 13.75 samples (at 90^0angle). Considering linearity till about 60^0, and considering that the algorithm estimates only integer sample delays, the sample delay ranges from 0 (at 0^0angle) to 12 (at 60^0angle) in integers. So, in 0 to 90^0quadrant, we get 13 different angle estimations.

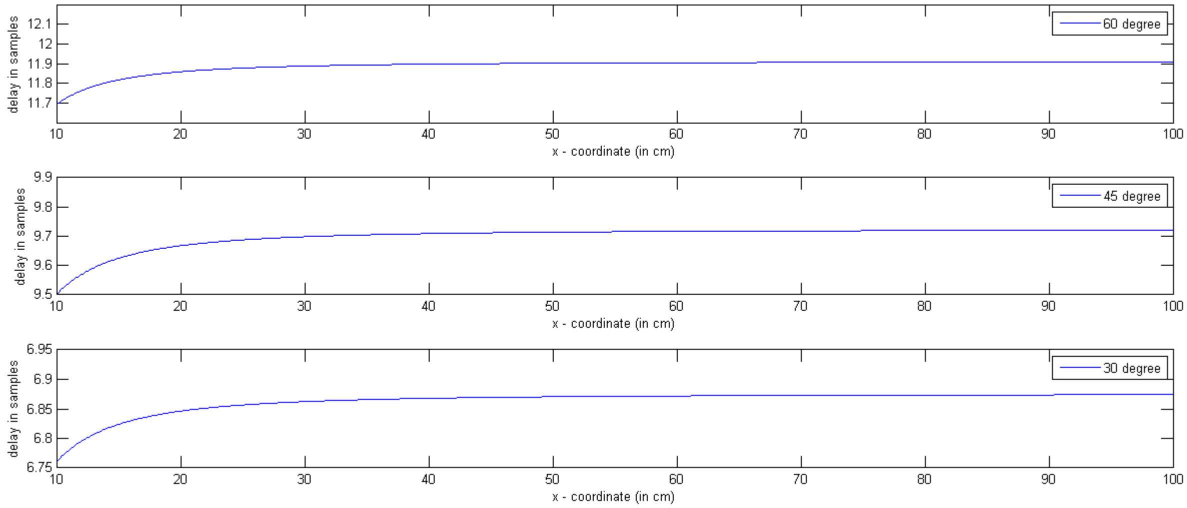


Figure 5.3: Variation in sample delay at an angle

Figure 5.3 shows the variation in sample delay at a particular angle as the source is moved away from the microphone array. It can be seen that at distances lower than 20 cm, there is a considerable variation in delay. Though the variation is only in the first decimal, the algorithm does rounding integer sample delay estimation. That is, if the actual delay is 11.4 samples, the estimated sample delay will be 11. If it is 11.6, estimation would be 12. Hence, this variation might be an issue. Thats why, the distance from the array is always maintained above 25 cm.

As the algorithm estimates integer sample delays,

5.2.2 Results

1024 samples are taken per channel at a particular instant using Stellaris ADC.

- Sampling Frequency is 55000 samples/sec.
- Speed of sound is assumed to be constant at 340 m/s.

A two microphone array ($-60^0\ to\ 60^0$ measurement is possible) is used with amplifiers, connected to Stellaris Evalbot. The whole calculations are done on the microcontroller and the result is displayed on an external LCD panel that is also controlled by the Microcontroller. Hence, this forms a memory constrained module. 1200 Hz sine wave is generated by an application in Samsung Smartphone and the phone speaker is used as the source. This is apt as the assumed point source.

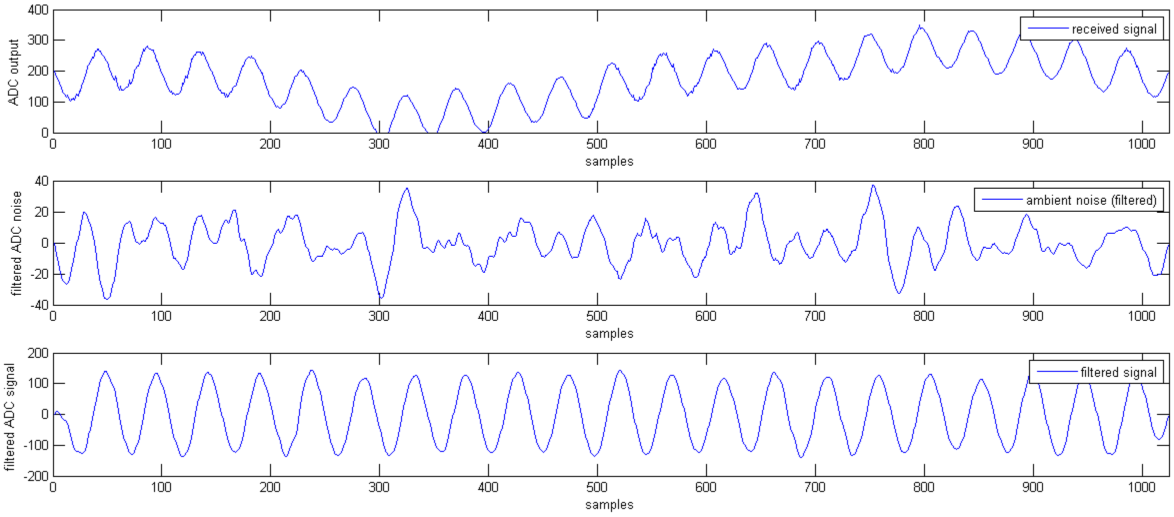


Figure 5.4: Received signals

The received signal of 1200 Hz sine wave, ambient noise and filtered signal which is used for further processing are shown in the figure. the source is about 25 cm from the center of microphone array. The SNR in the experimental setup turns out to be about 33 dB. In this condition, we have tested the angle estimation. There is approximately 5^0 error in -50^0 to 50^0 estimation. Because this is a memory constrained implementation and a band pass filter is used in the algorithm, it will take approximately 5 seconds to update the angle. This problem is not encountered using CPU. But this setup is very portable and compact.

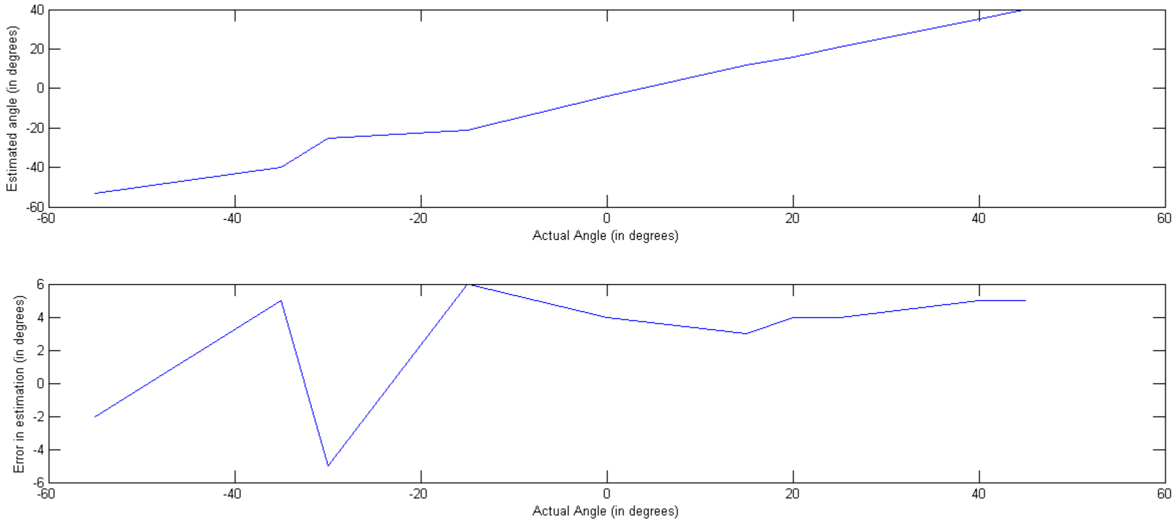


Figure 5.5: Estimated Angle Results

From figure 5.5, it can be noted that there is a maximum error of 6^0at an angle of -15^0angle.

5.3 Stellaris vs. CPU Implementations

The following table gives a comapritive understanding of the Stellaris and CPU implementations.

Table 5.1: Differences in Implementation : Stellaris vs. CPU

Stellaris 800 MHz	Intel Core2 Quad CPU 2.83 GHz
10 bit ADC input : 0 - 3 V	Virtex-5 10 bit ADC input: 0 - 1 V
Processing takes about 5 seconds	Processing takes about 50 ms
Resulting angle displayed on LCD	Resulting angle displayed on monitor
Implemented in C code	Implemented in Matlab code
Easily portable Setup	Fixed setup

The time delay in measurement by Stellaris Microcontroller is mainly because of using floating point arithmetic wherever necessary. It is noted that obtaining filtered signals from original ADC outputs is time expensive.

6 Conclusion and Future Work

In the project, practical implementation of angle estimation in real time is done in two ways: using CPU and using a microcontroller. After studying various TDE algorithms, GCC technique has been chosen to find out the time delays.

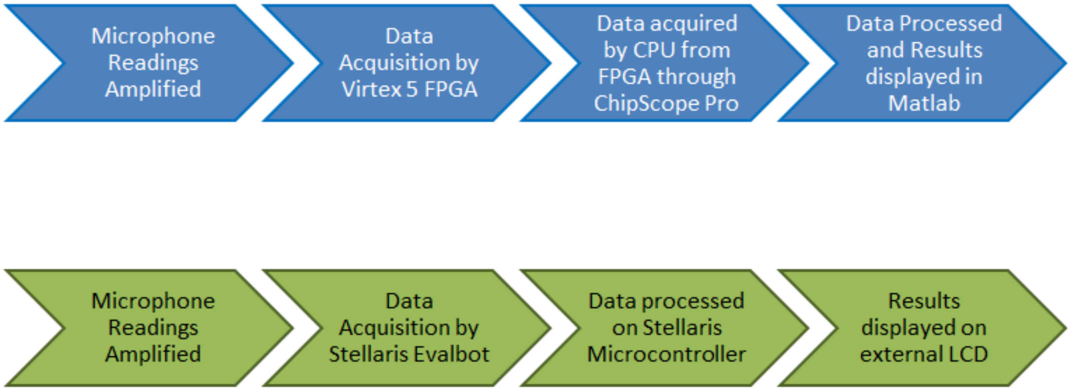

Microphone Readings Amplified
Data Acquisition by Virtex 5 FPGA
Data acquired by CPU from FPGA through ChipScope Pro
Data Processed and Results displayed in Matlab
Microphone Readings Amplified
Data Acquisition by Stellaris Evalbot
Data processed on Stellaris Microcontroller
Results displayed on external LCD

Figure 6.1: Two Implementations of Angle Estimation

Simulations have been done and results are presented for 2D angle and 3D position estimations.

As future work, practical implementation of 2D and 3D coordinate estimation can be tried out real time using this algorithm. Having multiple sound sources is another challenging task. There can be reduction in time complexity in case of using a microcontroller and make it more ’real time’.

7 Appendix

7.1 2D Angle Estimation - Sensitivity analysis

7.1.1 Sensitivity analysis w.r.t. speed of sound

Considering two microphones, we can get the angle of the source, as shown in the earlier sections. As we are considering speed of sound and distance between the microphones to be constant and already known, it is good to check the sensitivity of error with respect to these parameters.This is because, the distance between microphones can't be *exactly* the considered number. There may be a variation in the order of millimeters. Also, we know that speed of sound is a function of environmental parameters like temperature and humidity. Hence, there will be an error in considering it a constant at 340m/s. Let us see whether our approach is considerable robust for these variations or not. For this, we plot the error in measurement as the angle goes from 0 to 60^0.

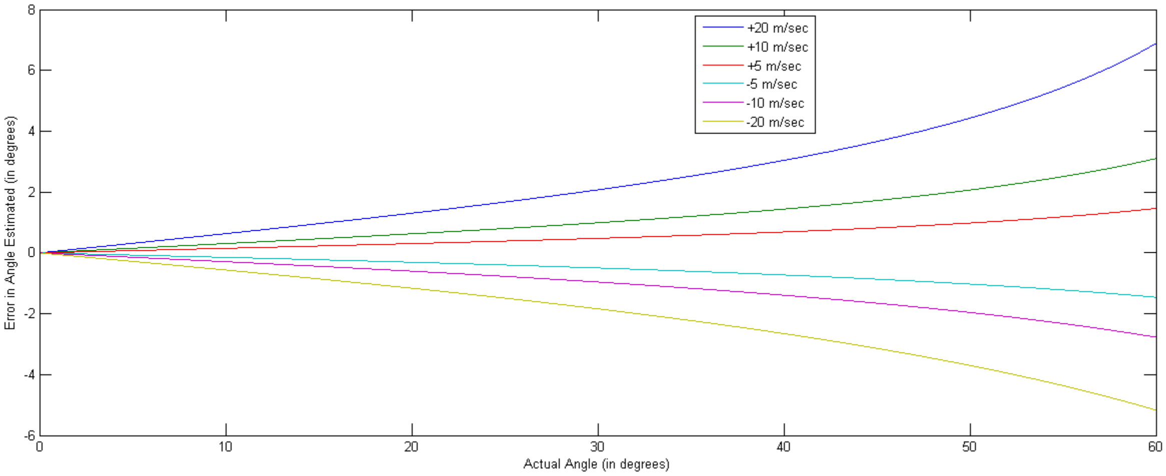


Figure 7.1: Variation in error in angle w.r.t. error in speed of sound with d=15 cm

The numbers for different colors indicate the error in speed of sound.

That is, $constant\,speed\,considered - actual\,speed$

We note from Figure 7.1 that, as the error in the angle measured increases with error in speed. Also, the error will be more for higher angles. We considered the angle sweep till 60^0only as we generally use a set of two microphones till a maximum of 60^0because of the before mentioned reasons of non linearity. The useful conclusion from this analysis is that a maximum of 6^0error occurs at around $60^0$for an error in speed consideration as high as 20 m/s. Hence, we can say that our approach is robust for variation in speed of sound.

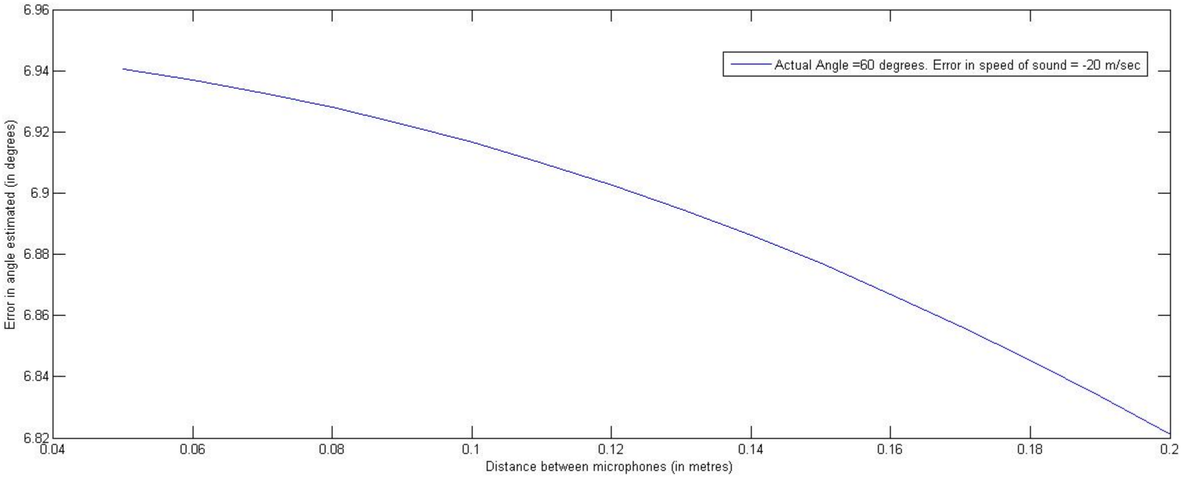


Figure 7.2: Error in estimation vs distance between microphones

The error in estimation for a 20m/s error in speed consideration, with varying geometry of microphones is presented in figure 7.5. We can see that, as the distance increases, the error on angle decreases.

7.1.2 Sensitivity analysis w.r.t. distance between microphones

The microphone used has a diameter of 9.7 mm and width of 5.2 mm. As already described, there may be an error in taking the distance between microphones. Hence, lets see the sensitivity of measurement to this error. As the distance is measures in cm, lets consider an error of -5 to 5 mm with respect to the actual distance.

We present the graphs of error in angle, separately for positive error and negative error in distance.

Error in $d = Considered distance in algorithm - actual distance$

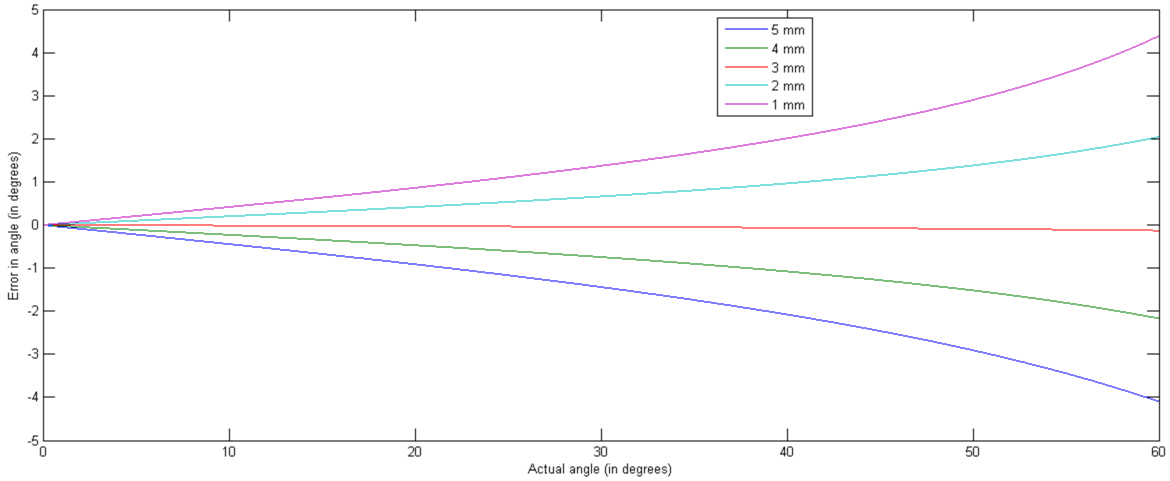


Figure 7.3: Positive error in *d* at 5 cm spacing

We can see from figure 7.4 and figure 7.3 that, if a distance more than actual is used in calculation, the error is very less compared to, if lesser value is used. If we consider a positive error of 5mm in *d*, the maximum error in angle is around 4^0at 60^0,which is fine for practical usage.

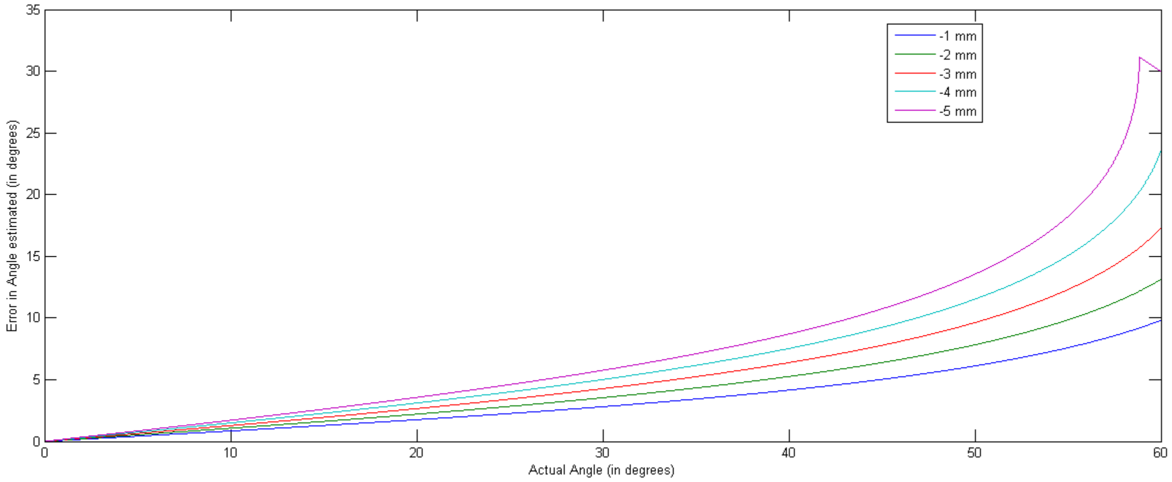


Figure 7.4: Negative error in *d* at 5 cm spacing

Hence, in the practical experimentation, the distance is used as

$(distance between centres of microphones + 5mm)$

Let us see how error varies as the actual distance increases. Here we consider the error in *d* to be maximum positive at$10mm$ and vary the actual distance from 5 cm to 25 cm. We take the angle where the maximum error is observed, that is $60^0$and look at the max. error in estimation.

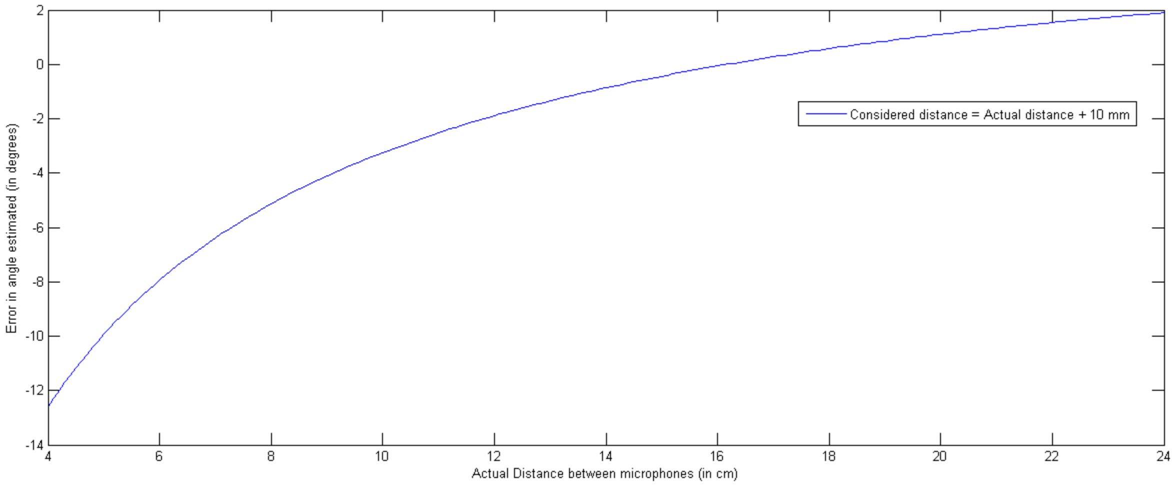


Figure 7.5: Error in estimation vs d, for an error of 10 mm in d at 60^0

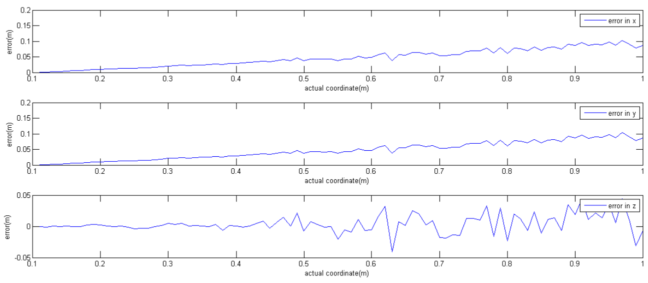
We can see from figure 7.5 that as the distance between microphones (d) increases, the result becomes more robust to the error in d.

7.2 Four microphone array

7.2.1 Sensitivity analysis w.r.t. Signal to Noise Ratio

Table 7.1: Error in Measurement on x=y=z line with SNR 44 dB

(a) Error graph

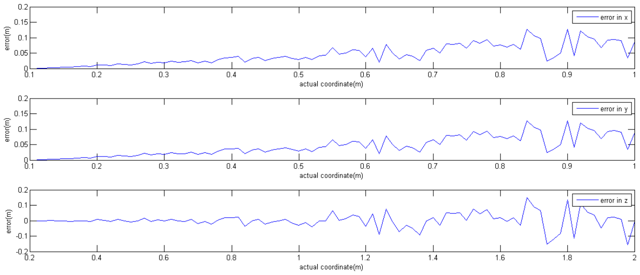


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	10.3	97
y	10.3	97
z	12.46	97

Table 7.2: Error in Measurement on x=y=z/2 line with SNR 44 dB

(a) Error graph

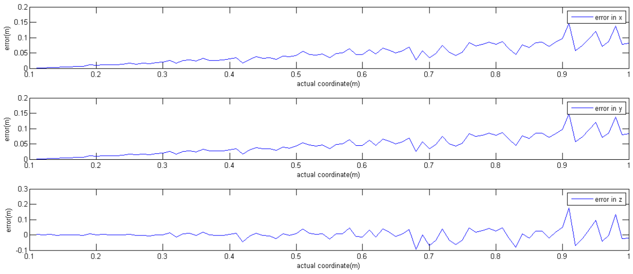


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	12.6	90
y	12.7	90
z	15.6	198

Table 7.3: Error in Measurement on x=y=z line with SNR 32 dB

(a) Error graph

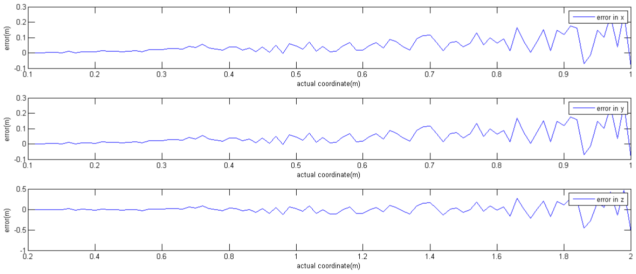


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	14.7	91
y	14.7	91
z	17.5	91

Table 7.4: Error in Measurement on x=y=z/2 line with SNR 32 dB

(a) Error graph

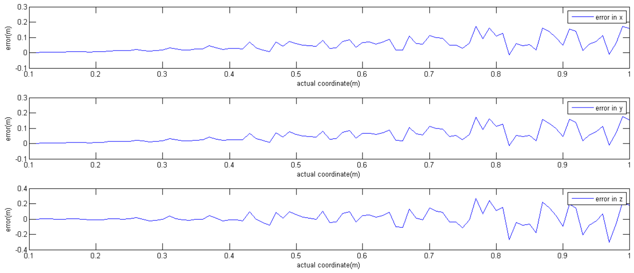


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	25.9	99
y	25.9	99
z	53.3	200

Table 7.5: Error in Measurement on x=y=z line with SNR 20 dB

(a) Error graph

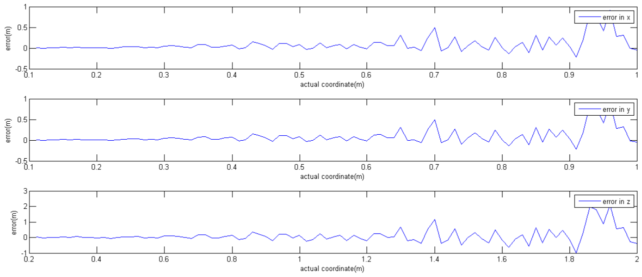


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	17.0	77
y	17.4	99
z	30.2	97

Table 7.6: Error in Measurement on x=y=z/2 line with SNR 20 dB

(a) Error graph



(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	90.8	96
y	90.3	96
z	200	200

7.2.2 Sensitivity analysis w.r.t. speed of sound

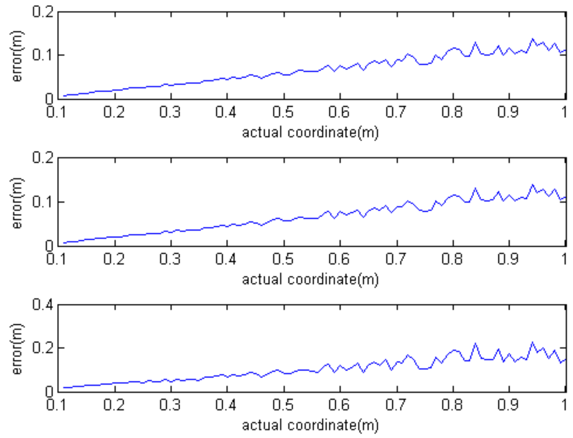
As we are considering speed of sound and distance between the microphones to be constant and already known, it is good to check the sensitivity of error with respect to these parameters.This is because, the distance between microphones can't be *exactly* the considered number. There may be a variation in the order of millimeters. Also, we know that speed of sound is a function of environmental parameters like temperature and humidity. Hence, there will be an error in considering it a constant at 340m/s. Let us see whether our approach is considerable robust for these variations or not.

Let us say, we are considering the speed of sound constant at 340m/s, but the actual is at 320 m/s. That is, there is an error of 20 m/s. Let us look at the error graph then.

From table 7.7, considering the line$x = y = z$, $x, y, z \leq 1$, we have the error graph of all the coordinates. The results are tabulated in table 7.7.

Table 7.7: Error in Measurement on x=y=z line with error in speed of sound = 20 m/s

(a) Error graph



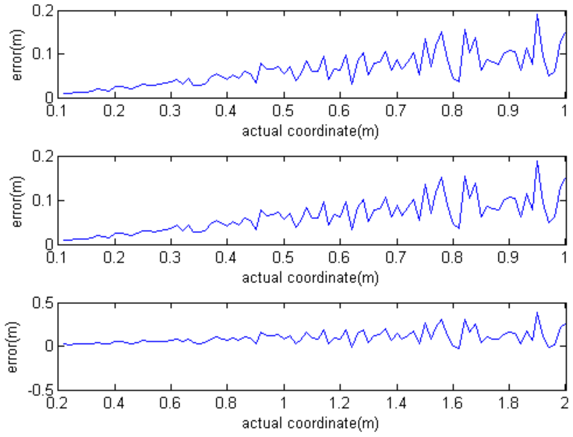
(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	13.77	94
y	13.69	94
z	22.43	94

From table 7.8, considering the line$x = y = \frac{z}{2}$, $x, y \leq 1$, we have the error graph and results of all the coordinates. The results are tabulated in table 7.8.

Table 7.8: Error in Measurement on x=y=z/2 line with error in speed of sound = 20 m/s

(a) Error graph



(b)

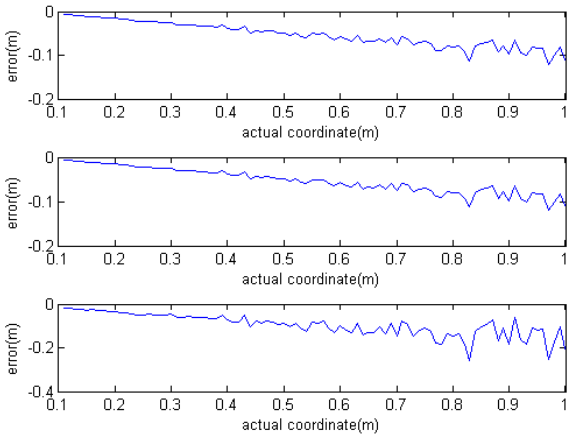
Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	18.9	95
y	18.9	95
z	38.2	190

Let us say, we are considering the speed of sound constant at 340m/s, but the actual is at 360 m/s. That is, there is an error of -20 m/s. Let us look at the error graph then.

From table 7.9, considering the line$x = y = z$, $x, y, z \leq 1$, we have the error graph and results for all the coordinates. The results are tabulated in table 7.9.

Table 7.9: Error in Measurement on x=y=z line with error in speed of sound = -20 m/s

(a) Error graph



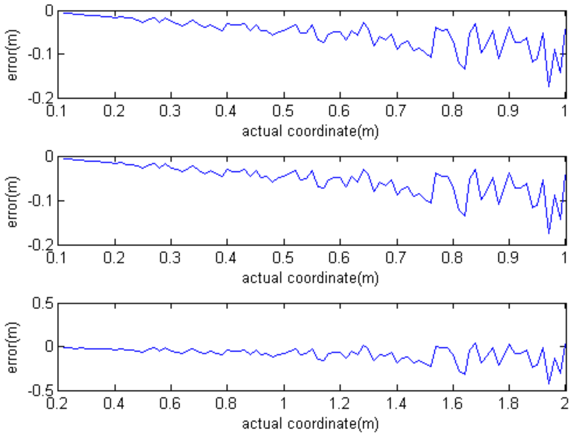
(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	12.0	97
y	11.9	97
z	25.8	83

From table 7.10 , considering the line$x = y = \frac{z}{2}$, $x, y \leq 1$, we have the error graph and results of all the coordinates. The results are tabulated in table 7.10.

Table 7.10: Error in Measurement on x=y=z/2 line with error in speed of sound = -20 m/s

(a) Error graph



(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	17.4	97
y	17.4	97
z	42.7	194

7.2.3 Sensitivity analysis w.r.t. array geometry

As we have already described, there may be an error in taking the distance between microphones. Hence, lets see the sensitivity of measurement to this error. As the distance is measures in cm, lets consider an error of -10 and 10 mm with respect to the actual distance.We present the graphs of error in angle, separately for positive error and negative error in distance.

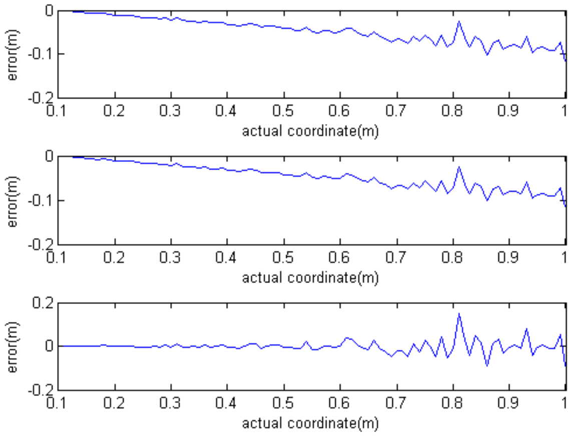
Error in $d = Considered\,distance\,in\,algorithm - actual\,distance$

Let us say, we considered the square side, a as 10 cm. But it is indeed 9 cm. So there is an error of 1 cm.

From table 7.11, considering the line$x = y = z$, $x, y, z \leq 1$, we have the error graph of all the coordinates.

Table 7.11: Error in Measurement on x=y=z line with error in square array side = 1 cm

(a) Error graph



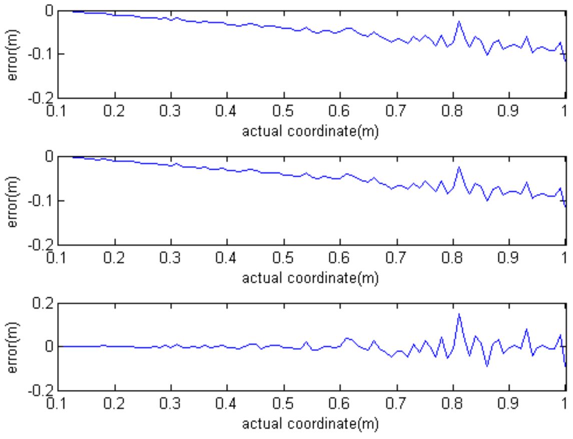
(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	12	100
y	12	100
z	14.8	81

From table 7.12, considering the line$x = y = \frac{z}{2}$, $x, y \leq 1$, we have the error graph and results for all the coordinates.

Table 7.12: Error in Measurement on x=y=z/2 line with error in square array side = 1 cm

(a) Error graph



(b)

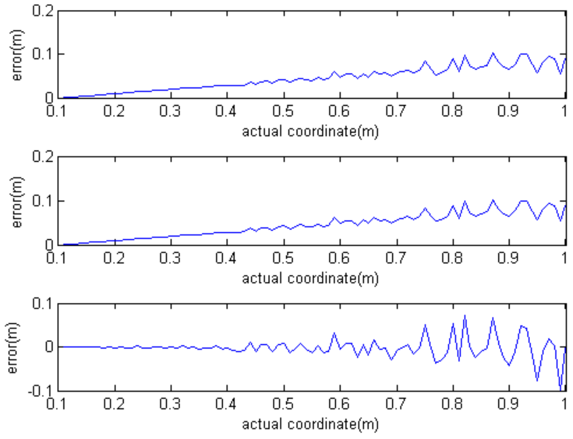
Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	21	93
y	21	93
z	45.8	186

Let us say, we considered the square side, a as 10 cm. But it is indeed 11 cm. So there is an error of -1 cm.

From table 7.13, considering the line$x = y = z$, $x,y,z \leq 1$, we have the error graph and results for all the coordinates.

Table 7.13: Error in Measurement on x=y=z line with error in square array side = -1 cm

(a) Error graph



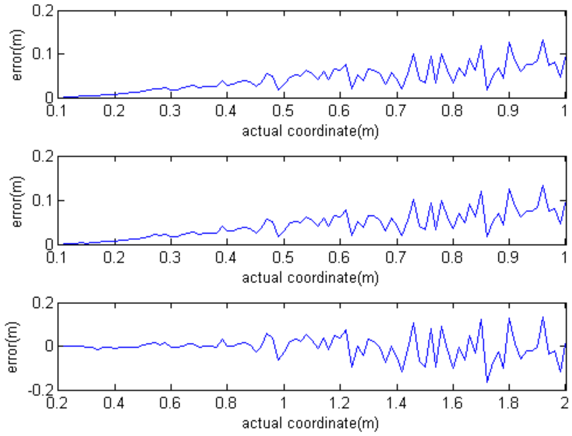
(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	10.1	87
y	10.1	87
z	9.7	99

From table 7.14, considering the line$x = y = \frac{z}{2}, x, y \leq 1$, we have the error graph and results for all the coordinates.

Table 7.14: Error in Measurement on x=y=z/2 line with error in square array side = -1 cm

(a) Error graph



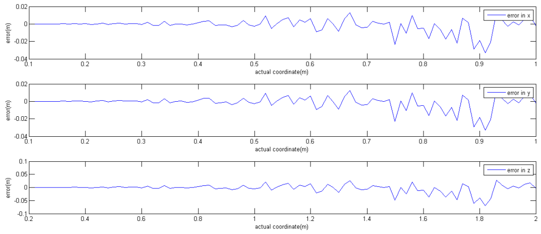
(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	13.2	96
y	13.3	96
z	16.8	172

7.3 Nine microphone array

Table 7.15: Error in Measurement on x=y=z line with SNR 44 dB

(a) Error graph

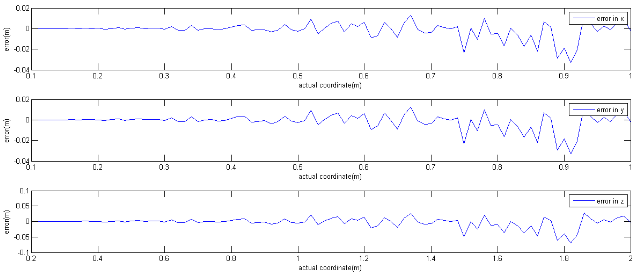


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	2.2	99
y	2.2	99
z	2.3	99

Table 7.16: Error in Measurement on x=y=z/2 line with SNR 44 dB

(a) Error graph

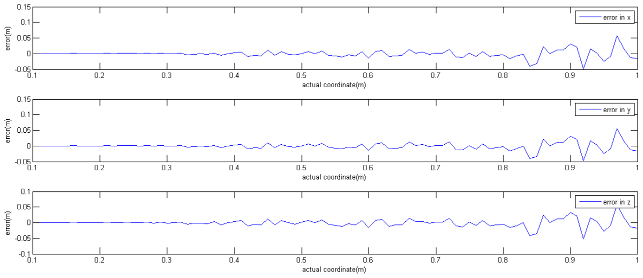


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	3.28	91
y	3.27	91
z	6.91	182

Table 7.17: Error in Measurement on x=y=z line with SNR 32 dB

(a) Error graph

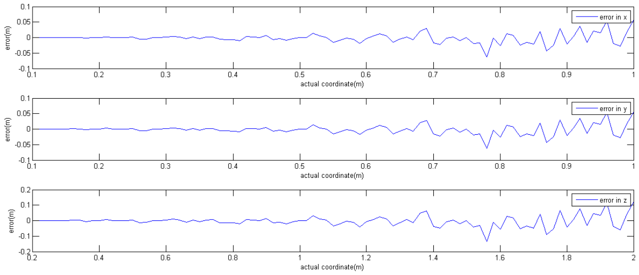


(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	5.7	97
y	5.6	97
z	5.7	97

Table 7.18: Error in Measurement on x=y=z/2 line with SNR 32 dB

(a) Error graph



(b)

Error in	Maximum error (cm)	Actual coordinate at of max error (cm)
x	6.3	78
y	6.2	78
z	13.4	156

Table 7.19: Error in Measurement on x=y=z line with SNR 20 dB

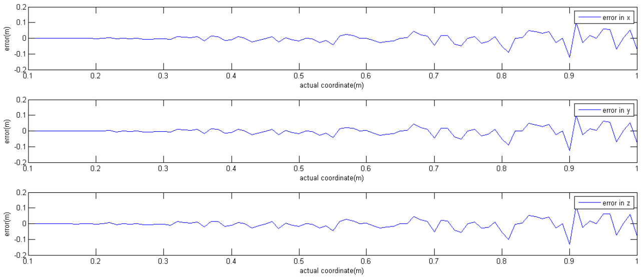
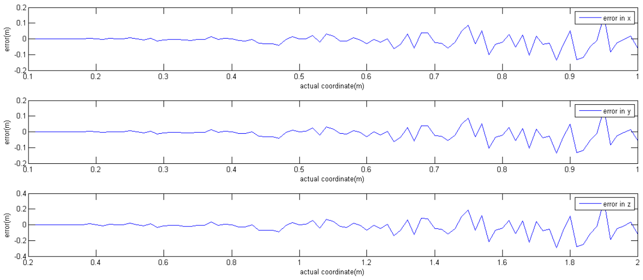
<table>
<tr><th>(a) Error graph</th><th>(b)</th></tr>
<tr><td>
</td><td>
<table>
<tr><th>Error in</th><th>Maximum error (cm)</th><th>Actual coordinate at of max error (cm)</th></tr>
<tr><td>x</td><td>12.1</td><td>90</td></tr>
<tr><td>y</td><td>12.3</td><td>90</td></tr>
<tr><td>z</td><td>13.2</td><td>90</td></tr>
</table>
</td></tr>
</table>

Table 7.20: Error in Measurement on x=y=z/2 line with SNR 20 dB

<table>
<tr><th>(a) Error graph</th><th>(b)</th></tr>
<tr><td>
</td><td>
<table>
<tr><th>Error in</th><th>Maximum error (cm)</th><th>Actual coordinate at of max error (cm)</th></tr>
<tr><td>x</td><td>14.5</td><td>95</td></tr>
<tr><td>y</td><td>14.5</td><td>95</td></tr>
<tr><td>z</td><td>30.7</td><td>190</td></tr>
</table>
</td></tr>
</table>

8 References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[28] G. C. Carter: “Coherence and time delay estimation: an applied tutorial for research, development, test, and evaluation engineers”, Piscataway, NJ: IEEE Press, 1993

[29] C. H. Knapp and C. G. Carter: “The generalized correlation method for estimation of time delay”, IEEE Trans, Acoust, Speech, Signal Processing, vol. ASSP-21, pp. 320-327, August 1976

[30] Z. CH. Liang, X. ZH. Liu and Y. T. Liu: “A modified time delay estimation algorithm based on higher order statistics for signal detection problems”, IEEE Signal Processing, vol.1, pp.255-258, Aug 2002

[31] A. K. Nandi: “Subsample time delay estimation of narrowband ultrasonic echoes”, IEEE Ultrason, Ferroelect, Freq. Contr., vol.42, pp.993-1001, November 1995

[32] CH. Zheng and TH. T. Tjeng: “A new time delay estimator based on ETDE”, IEEE Transactions On Signal Processing, vol.51, no.7, pp.1859-1869, July 2003

[33] M. Omologo and P. Svaizer: “Acoustic Event Localization Using A Crosspower-Spectrum Phase Based Technique”, IEEE Acoustics, Speech, and Signal Processing, vol.2 pp.273-276, April 1994

[34] L. CH. Chu and U. Mitra: “Analysis of MUSIC-based delay estimator for direct-sequence code-division multiple-access systems”, IEEE Transactions on Communications, vol.47, no.1, pp.133-138, Jan 1999

[35] R. J. Barsanti and M. Tummala: “Wavelet-based time delay estimates for transient signals”, IEEE Signals, Systems & Computers, vol.1, pp.1173-1177, 2003

[36] M. Jian, A. C. Kot and M. H. Er: “Performance Study of Time Delay Estimation In A Room Environment”, IEEE, Circuits and Systems, vol.5, pp.554-557, June 1998

[37] Q. Y. Zou and ZH. P. Lin: “Measurement Time Requirement For Generalized Cross-correlation Based Time-Delay Estimation”, IEEE Circuits and Systems, vol.3 pp.492-295, May 2002

[38] J. P. Ianniello: “Time delay estimation via cross-correlation in the presence of large estimation errors”, IEEE Trans. Acoust., Speech, Signal Processing, Vol. 30 No.6, pp.998-1003, December 1982

[39] S. Bedard, B. Champagne and A. Stephenne: “Effects Of Room Reverberation On Time-Delay Estimation Performance”, IEEE, Acoustics, Speech, and Signal Processing, vol.2, pp.261-264, April 1994

[40] A. J. Weiss and E. Weinstein: “Fundamental limitations in passive time delay estimation-Part ‡T: Narrow-band systems,” IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-31, pp.472-486, Apr. 1983

[41] A. K. Nandi: “On the Subsample Time Delay Estimation of Narrowband Ultrasonic Echoe”, IEEE Transactions on Ultrasonics, Ferroelectrics, And Frequency Control, Vol.42, N0.6, pp.993-1001, November, 1995

[42] E. Weinstein and A. J. Weiss: “Fundamental Limitations in Passive Time-Delay Estimation-Part ‡U: Wide-Band Systems”, IEEE Trans. Acoust., Speech, Signal Processing, vol.32,no.5, pp.1064-1077, October 1984

[43] H. Wang and P. Chu: “Voice Source Localization For Automatic Camera Pointing System In Videoconferencing”, IEEE Acoustics, Speech, and Signal Processing, vol.1 pp.187-190, April 1997

[44] G. Jacovitti and G. Scarano: “Discrete Time Techniques for Time Delay Estimation”, IEEE Transactions on Signal Processing, Vol.41, No.2, pp.525-533 February, 1993

[45] A. Jakobsson, A. L. Swindlehurst and P. Stoica: “Subspace-based estimation of time delays and Doppler shifts”, IEEE Transactions On Signal Processing, vol.46, no.9, pp.2472-2483, September 1998

[46] http://www.digikey.com

[47]http://www.panasonic.com/industrial/components/pdf/em06_wm61_a_b_dne.pdf

[48]ML505/ML506/ML507 Evaluation Platform User Guide from Xilinx.

[49]Virtex-5 FPGA System Monitor User Guide from Xilinx.

[50] Y. Huang, J. Benesty, and G. W. Elko, “Microphone Arrays for Video Camera Steering,” Acoustic Signal Processing for Telecommunications, ed. S. L. Gay and J. Benesty, Kluwer Academic Publishers, 2000.

[51] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: Real-time tracking of the human body,” Proc. on Automatic Face and Gesture Recognition, 1996, pp. 51-56.

[52] Travis Wiens and Stuart Bradley, “A comparison of Time delay estimation methods for periodic signals”, University of Auckland, 2009.