Sound Source Localization

PROJECT MEMBERS:

1. SINDHURA CHILAKAPATI
P.no: 940406-9321

2. SRI LAKSHMI JYOTHIRMAI MAMIDALA
P.no: 930929-4602

UNDER THE ESTEEMED GUIDANCE OF:

JOSEF STROM BARTUNEK
Email: josef.strombartunek@bth.se
DEPARTEMENT OF APPLIED SIGNAL PROCESSING

CONTENTS

Abstract

Sound source localization has the goal of locating a sound source with given measurements of the sound field. The concept of sound source localization used in this paper requires more than two microphones in a 2-D plane since two coupled microphones can give only one information i.e., direction of source. To estimate the location of the sound source, an array of microphones is considered. The least square method is used to estimate the practical time delays. The Steepest Descent algorithm is used to localize the sound source.

LIST OF FIGURES:

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

The main objective of the project is to localize the sound source. We assume an array of microphones for this purpose. The first step is to calculate the theoretical time delays(t_i) of each microphone with respect to the reference microphone. Then the delayed signal is produced by using the filters with delay(t_i) and Sinc functions as input [1]. Using these delayed signals, the practical delays are estimated by 'Least Square method' [2] implemented with adaptive filters. The practical delays are then compared with the theoretically calculated delays.

The function that determines the time difference between elements 'i' and '0' from sound source in (x', y') is defined. Using this function and the practically estimated delays, the position of the sound source is estimated with the help of 'Steepest Descent Algorithm'. [3]

1.2 LITERATURE REVIEW

Sound Source localization is the technique of capturing the sound and estimating the position of the source. The Sound Source Localization is done depending on the test object, the nature of the sound and the environment. In the case of environment with higher interference, any acoustics engineer faces the complex toughest challenge to figure out where the sound originates.[4]

Different methods for obtaining either source direction or source location are possible:

<u>Particle velocity or intensity vector</u> : The simplest but still a relatively new method is to measure the acoustic particle velocity using a particle velocity probe. The particle velocity is a vector and thus also contains directional information.

<u>Time difference of arrival:</u> The traditional method to obtain the source direction is using the time difference of arrival (TDOA) method. This method can be used with pressure microphones as well as with particle velocity probes. With a sensor array (for instance a microphone array) consisting of at least two probes it is possible to obtain the source direction using the cross-correlation function between each probes' signal. The cross-correlation function between two microphones is defined as

$$R_{x_1,x_2}(\tau) = \sum_{n=-\infty}^{\infty} x_1(n)\, x_2(n+\tau) \tag{1}$$

which defines the level of correlation between the outputs of two sensors x_1 and x_2.
In general, a higher level of correlation means that the argument τ is relatively close to the actual time-difference-of-arrival. For two sensors next to each other the TDOA is given by

$$\tau_{\text{true}} = \frac{d_{\text{spacing}}}{c} \tag{2}$$

where c is the speed of sound in the medium surrounding the sensors and the source.
A well-known example of TDOA is the interaural time difference. The interaural time difference is the difference in arrival time of a sound between two ears. The interaural time difference is given by

$$\Delta t = \frac{x \sin\theta}{c} \qquad (3)$$

where
Δt is the time difference in seconds
x is the distance between the two sensors (ears) in meters
θ is the angle between the baseline of the sensors (ears) and the incident sound, in degrees.

Triangulation: In trigonometry and geometry, triangulation is the process of determining the location of a point by measuring angles to it from known points at either end of a fixed baseline, rather than measuring distances to the point directly (trilateration). The point can then be fixed as the third point of a triangle with one known side and two known angles. For acoustic localization this means that if the source direction is measured at least two locations in space, it is possible to triangulate its location.

In our project, an array of microphones are considered and time difference of arrival has been chosen for the Sound Source Localization.

1.3 ADVANTAGES & DISADVANTAGES OF SOUND SOURCE LOCALISATION TECHNIQUE

Advantages:

- We can find location of sound source with accurate positions.
- Sound source localization are more accurate than other contact based methods like motion detectors [5].

Disadvantages:

- Highly sensitive to initial position due to local maxima's.
- Difficult to implement in real time system since the arrangement in real time system should meet the specifications like equal spacing between the mics, source being within the range of microphones, etc.
- Noisy environments tends to decrease the reliability since the time delay is increased with noise. [6]

1.4 LIMITATIONS

The considerations in this project are limited to certain conditions. The proposed algorithm is bounded to these conditions. They are:

- The array of microphones considered is a ‘linear array’.
- The spacing between the microphones should be equal.
- The location of the sound source cannot be far-off i.e. the sound source cannot be assumed far away from microphones.
- The plane considered is a two-dimensional plane.

1.5 THESIS ORGANISATION

The structure of the thesis is illustrated in figure 1.1. In more detail, this thesis has five chapters.

Chapter 1 provides an overview of on-going directions in this goal-driven research. In this chapter the over view of proposed technique, the advantages and the disadvantages related to our approach has been discussed.

Chapter 2 deals with the concept of Sound Source Localisation. We describe the best techniques available for locating the position of sound source.

In chapter 3, we implement the techniques used and their algorithms and thereby we develop source code. Here, the source code is explained in detail using a flow chart.

In chapter 4, we obtain step by step plots of the corresponding code developed. Also we describe about every output obtained and finally we find locate the sound source position.

In chapter 5, we finally conclude about our work and also suggest the future scope to achieve our goal more easily and in advance.

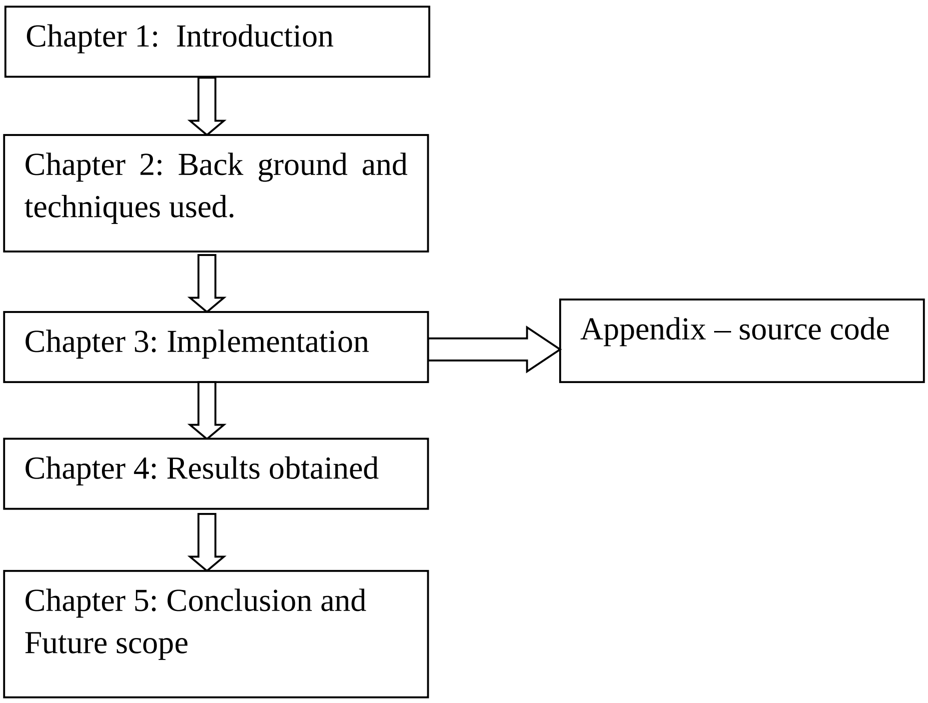


figure 1.1: representation of thesis organization.

CHAPTER 2: SOUND SOURCE LOCALISATION

2.1 BACKGROUND

- Sound Source Localization is the process of identifying the location of a detected source.

  Considering the case of enclosed rooms, there exists two types of sound waves: the direct sound from a sound source and the sound which has been reflected at the walls. According to later law of first wave front, the auditory system analyses only the direct sound, which is arriving first for sound localization, but not the reflected sound, which is arriving later. So sound localization remains possible even in an echoic environment. This echo cancellation occurs in the Dorsal Nucleus of the lateral lemniscuses. (DNLL).

  In order to determine the time periods, where the direct sound prevails and which can be used for directional evaluation, the auditory system analyses loudness changes in different critical bands and also the stability of the perceived direction. If there is a strong attack of the loudness in several critical bands and if the perceived direction is stable, this attack is in all probability caused by the direct sound of a sound source, which is entering newly or which is changing its signal characteristics. This short time period is used by the auditory system for directional and loudness analysis of this sound. [7] [8]

  When reflections arrive a little bit later, they do not enhance the loudness inside the critical bands in such a strong way, but the directional cues become unstable, because there is a mix of sound of several reflection directions. As a result no new directional analysis is triggered by the auditory system. This first detected direction from the direct sound is taken as the found sound source direction, until other strong loudness attacks, combined with stable directional information, indicate that a new directional analysis is possible.

- Now, we use the technique of using array of microphones to capture sound and estimate source position. Thus, concept of sound source localization requires more than two microphones in a 2-D plane since two coupled microphones can give only one information i.e., direction of source.

  The sound waves generated by the source reaches the first microphone earlier than the second. If the difference in the propagation delay and the sound velocity in air is known, the path difference of the sound waves can be estimated.

2.2 PROBLEM FORMULATION

Two steps are involved in the problem of sound source localization:

• The signal received by several microphones is processed to obtain information about the time-delay between pairs of microphones.

• The estimated time-delays for pairs of microphones can be used for finding the location of the sound source.

2.2.1 TIME DELAY ESTIMATION:

The delay t_i between sound source at location (x, y) and each microphone can be mathematically obtained as:

$t_i = \frac{\sqrt{(x-id)^2+(y^2)}}{c}$ (4)

Where; i is the number of the respective microphone.

d is the distance between the adjacent microphones in meters

c is the speed of sound in meters/second

The delays (dT_i) between reference microphone x_0 to each of the other microphones x_i can then be obtained:

$$\mathrm{d}T_i = t_i - t_0 \quad (5)$$

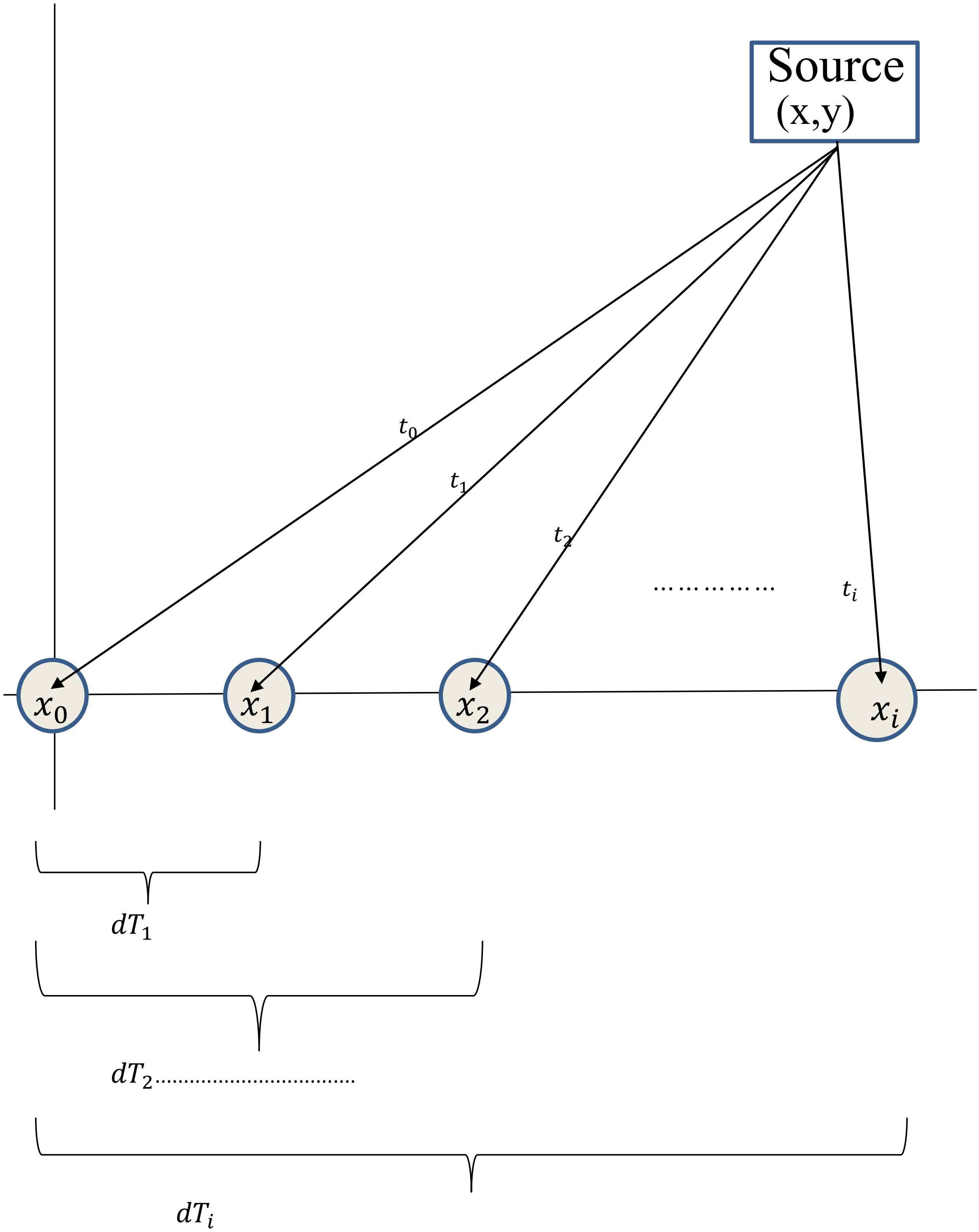
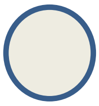


Figure 2.1: time delay estimation

where,  $\xrightarrow{\text{is}}$ Microphone

Now the algorithms used for finding the estimated position are described below:

2.2.2 LMS ALGORITHM:

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. These algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal. It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. [2]

Assume that $r_1(n)$ is the reference signal and $r_2(n)$ is the desired signal. The LMS output can be expressed by:

$$y_{12}(n) = w^T{}_{12}(n)R_{12}(n) \qquad (6)$$

Here:

T is the transpose function;

$R_{12}(n)$=$[r_1(n)\,, r_1(n-1)\ \ldots.\ r_1(n-L+1)]^T$ is the filter state consisting of the most recent samples of the reference signal;

Vector $w_{12}(n)$ denotes the vector (length L) of filter weights at any instant n.

The error signal $e_{12}(n)$can be obtained as:

$$e_{12}(n)\text{=}r_2(n) - w^T{}_{12}(n)R_{12}(n) \quad (7)$$

The weight vector is updated every sample according to:

$$w_{12}(n+1) = w_{12}(n) + \mu e_{12}(n)R^*{}_{12}(n)\ (8)$$

Finally, the LMS algorithm for a 'P' th order algorithm can be summarized as:

<u>Parameters:</u>'P' order; $\mu -$ step size

Initialisation:

$\hat{w}(0) = \text{zeroes}(p)$

Computation:

$\hat{w}(n+1) = \hat{w}(n) + \mu e^*(n)x(n)$ For n=0, 1,2,3......

In this project:

- The time delays dt_i can be estimated using LMS algorithm by making use of adaptive filter.
- The source signal is given a delay of t_i by passing the source signal through the filter. The source signal is considered to random.

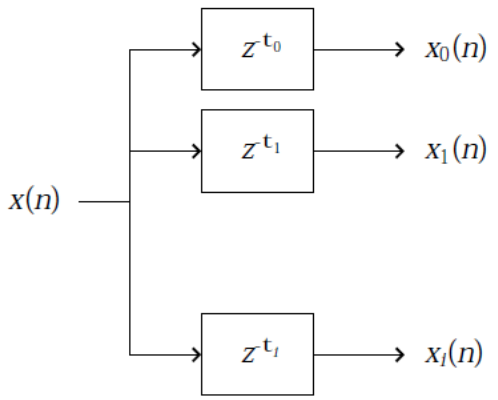


figure 2.2: filter to produce delay of t_i

- Now the signal $x_0(n)$ is taken as the reference signal and one of the remaining signals of $x_i(n)$ is taken as the desired signal.
- It cannot be predicted if the delay between the reference microphone and any other microphone is either positive or negative. So, the desired signal is delayed with half of the total length (L) of the filter.

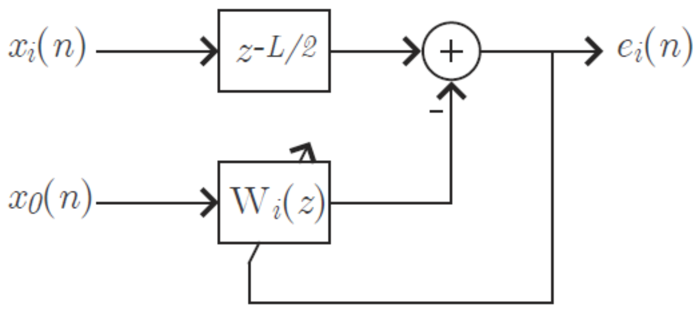


figure 2.3: filter to produce delay of L/2

- The filter considered is of limited length. So, an accurate estimation of time delays is done by analysing the slope of the phase of the impulse response of the filter.

- The phase of the filter is given by:
  $\phi_i(w) = k_i w + m_i$ (9)
- The slope can be obtained by taking the derivative of the equation (9).
  $D\tilde{T}_i = -\frac{d\phi_i(w)}{dw} = -k_i$ (10)
- The entire delay in samples can be obtained by adding the slope with the delay of the desired signal.
  $T_i = \frac{L}{2} + k_i$ (in samples) (11)
- The total delay is seconds can be obtained by dividing the equation (11) with sampling frequency.
  $t_i = \frac{1}{f_s}\left(\frac{L}{2} + k_i\right)$ in seconds (12)

2.2.3 STEEPEST DESCENT ALGORITHM

- Gradient descent algorithm is to find the local minimum of a function. It uses the concept of gradient descent.
- Here, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. [3]
- Gradient descent is also known as steepest descent, or the method of steepest descent.
- Gradient descent is based on the observation that if the multivariable function F_i(Y) is defined and differentiable in a neighbourhood of a point 'a', then decreases fastest if one goes from 'a' in the direction of the negative gradient of F_i at 'a-ΔF_i(a)'.
  It follows that, if b= a-ΔF_i(a), F_i(a) ≥ ΔF_i(b). (for γ small enough)
- With this observation in mind, one starts with a guess for a local minimum of F_i, and considers the sequence Y_o, Y_1, Y_2, Y_3..such that

$$Y_{n+1} = Y_n - \gamma n \Delta F_i(X_n),\ n \geq 0 \tag{13}$$

We have

$F_i(Y_o) \geq F_i(Y_1) \geq F_i(Y_2) \geq$ …..so hopefully the sequence Y_n converges to the desired local minimum.

- Here, in our project we use the steepest descend, to find min G(x,y). Here, G(x,y) is the difference between the obtained time delays and the function F_i(x,y).

$(x', y') = \min_{(x,y)} \sum_{i=1}^{N-1} (F_i(x,y) - D\tilde{T}_i)^2 = \min_{(x,y)}$ G(x,y) (14)

Where, F_i(x,y) is the function that determines real difference in time between element iand *0* from sound source in *(x',y')*.

$$F_i(x,y) = T_i - T_0 = \frac{\sqrt{(x-id)^2+y^2}}{c} - \frac{\sqrt{(x^2+y^2)}}{c} \quad (15)$$

- Now, the gradient vector is obtained by:

$$\begin{bmatrix} \frac{\partial G(x,y)}{\partial x} \\ \frac{\partial G(x,y)}{\partial y} \end{bmatrix} = \nabla \mathrm{G(x, y)} \quad (16)$$

CHAPTER 3: PROPOSED SOLUTION

3.1 ALGORITHM

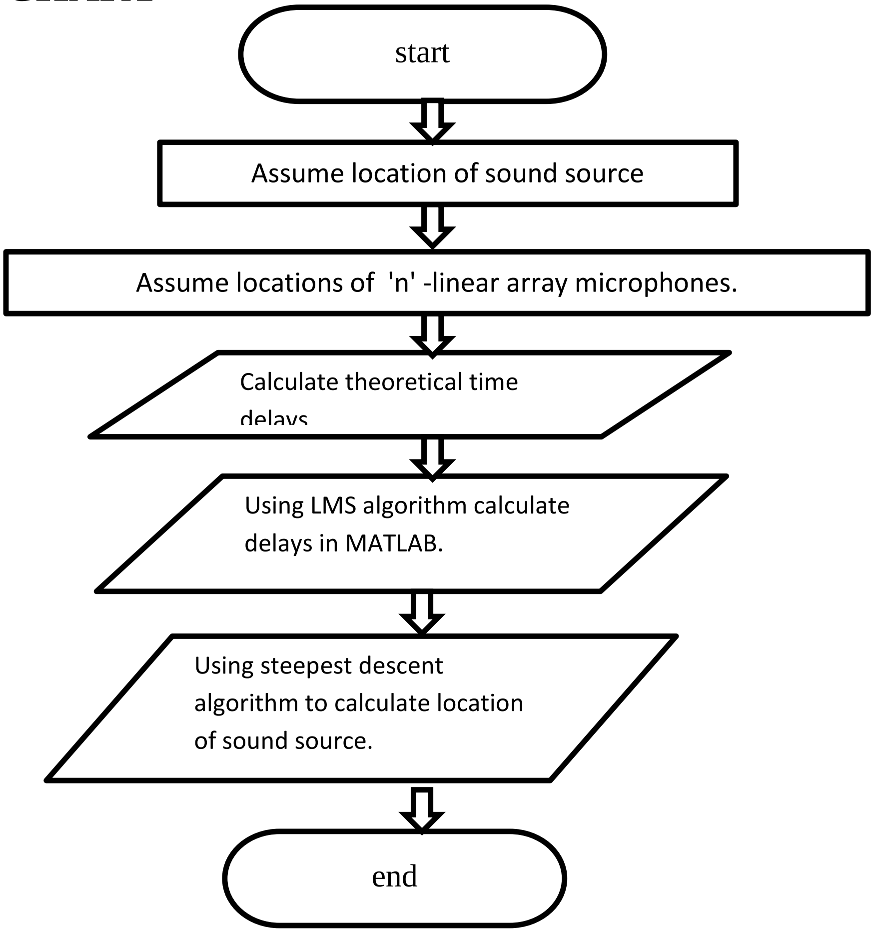
There are three main objectives in this project:

1. A robust method is chosen to estimate the time delaysfrom the filter weights.
2. $\frac{\partial G(x,y)}{\partial x}and\frac{\partial G(x,y)}{\partial y}$ are determined as in equation (16).
3. A function is written where coordinates *(x',y')* of the sound source is found as*(x',y')=minG(x,y)*as in equation (14).

Thus, the algorithm for the above steps can be designed as:

- The location of the sound source is assumed at some point in the coordinate axis.
- A linear array of microphones with equal spacing is considered on the x-axis and one microphone is set as reference.
- Theoretical delays of each microphone are calculated.
- The delays in samples are then calculated.
- Least square method is used to estimate the practical delays.
- The steepest descent algorithm is used to estimate the location of the sound source.

3.2 FLOWCHART



3.3 IMPLEMENTATION

The arrangement of the system is shown below:

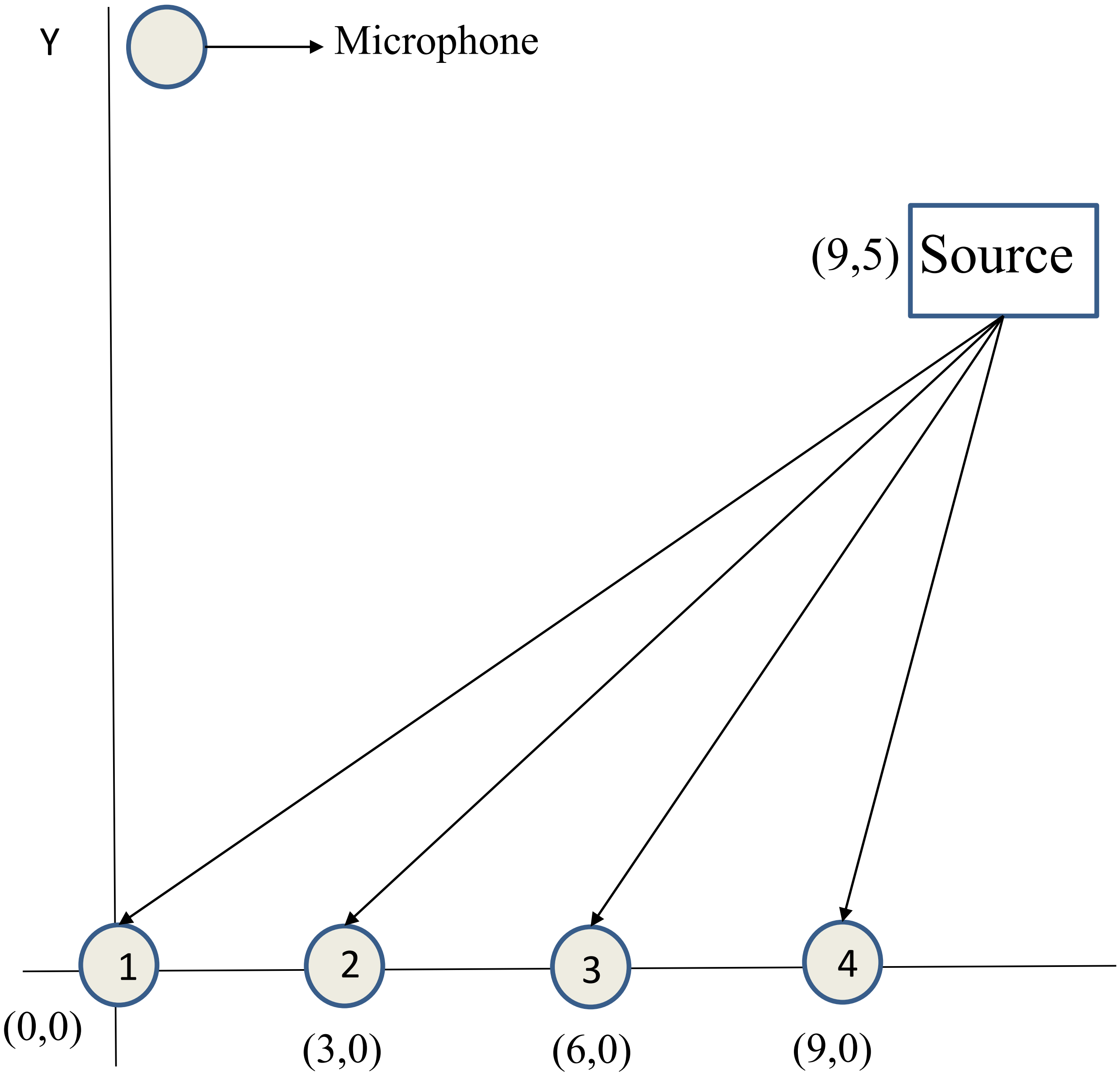


figure 3.1: coordinate system with sound source and array of microphones

3.3.1 SYSTEM DESIGN

- All the microphones are assumed to be on X-axis.
- Here, microphone 1 is considered as reference microphone.
- The location of the sound source is assumed at (9,5) meters.
- The spacing between the microphones is taken as 3 meters.

3.3.2 TIME DELAY ESTIMATION

STEP 1:
Calculating the distance between the source and each microphone:

Distance = $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ where (x_1, y_1) is he location of the source and (y_1, y_2) is the location of the corresponding microphone.

Thus, the distances can be calculated using above formula as:

Distance 1 (Microphone 1): $\sqrt{9^2 + 5^2}$ = 10.295 meters
Distance 2 (Microphone 2): $\sqrt{(9-3)^2 + (5-0)^2}$ = 7.810 meters
Distance 3 (Microphone 3): $\sqrt{(9-6)^2 + (5-0)^2}$ = 5.830 meters
Distance 4 (Microphone 4): $\sqrt{(9-9)^2 + (5-0)^2}$ = 5 meters

Calculating the time interval of each microphone:

Speed of sound in air = 343 meters/second

Microphone 1: t_1 = Distance 1/speed of sound = 0.0300 seconds

Microphone 2: t_2 = Distance 2/speed of sound = 0.0227 seconds

Microphone 3: t_3 = Distance 3/speed of sound = 0.0169 seconds

Microphone 4: t_4 = Distance 4/speed of sound = 0.0145 seconds

Here t_1 is the time interval between the source and reference microphone located at (0, 0) meters.

Here t_2, t_3, t_4 are the time intervals between the source and microphones 2,3 and 4 respectively.

Calculating the time delays of each microphone:

Microphone 2: $t_1 - t_2$ = 0.0300-0.0227 = 0.0073 seconds

Microphone 3: $t_1 - t_3$ = 0.0300-0.0169 = 0.0131 seconds

Microphone 4: $t_1 - t_4$ = 0.0300-0.0145 = 0.0155 seconds

STEP 2: (For practical time delays)

- 'Sinc' function is used in producing the delay (D_i).

  D_i= sinc(t -t_i); (17)

  where t lies between [-k,k], and k is an integer.

  and the delayed signals$x_i(n)$ described in the figure are obtained in MATLAB as:

  $x_i(n) = filter\ (D_i, 1, x(n))$ (18)

- The above obtained delayed signals are assumed to be the input desired signals for the adaptive filter as per figure 2.3

- Now, according to figure 2.3 these desired signals $x_i(n)$ are delayed by L/2 and let the obtained signals be $X_i(n)$.

  here L is the length of filter.

  $D_i{}'$= sinc(t); (19)

  where t lies between [-k/2,k/2]

  and$X_i(n)$ is obtained in MATLAB as $=> filter\ (D_i{}', 1, x_i(n)$)

- Now by implementing LMS algorithm, filter coefficients are obtained and plotted.

  filter coefficients [A_i] = lms($x_0(n)$,$X_i(n)$,μ,L); (20)

  [where μ is step size and L is length of filter]

- The phase of the filter is described as:

  $\phi_i(w) = \ k_i w + m_i$ (21)

  In the above equation, while determiningk_i and m_i we have more number of equations than variables. Thus it is said to be over determined system and we need to calculate least squares solution for it.

In MATLAB we use '\' (back slash operator) for obtaining the values of k.

- Now as per equation 1, time delay is estimated in samples and then converted to seconds by dividing with $f_s$as per equation 2. , where f_s is taken as 1200Hz.

- Finally, these obtained delays are compared with theoretical time delays.

3.3.2LOCALIZATION OF THE SOUND SOURCE:

To localize the sound source, a function $F_i(x, y)$ is defined by implementing the equation in MATLAB.
The location of the sound source (x', y') can be determined by the equation (14)

The value of $\min_{(x,y)}$ G(x,y) can be found by using the 'Steepest Descent Algorithm' which can be implemented as:

$$\begin{bmatrix} x \\ y \end{bmatrix}_{n+1} = \begin{bmatrix} x \\ y \end{bmatrix}_{n} - \mu \begin{bmatrix} \frac{\partial G(x,y)}{\partial x} \\ \frac{\partial G(x,y)}{\partial y} \end{bmatrix} \tag{22}$$

Here,

$$\begin{bmatrix} \frac{\partial G(x,y)}{\partial x} \\ \frac{\partial G(x,y)}{\partial y} \end{bmatrix} = \nabla G(x, y) \tag{23}$$

CHAPTER 4: SIMULATION RESULTS

4.1 RESULT 1

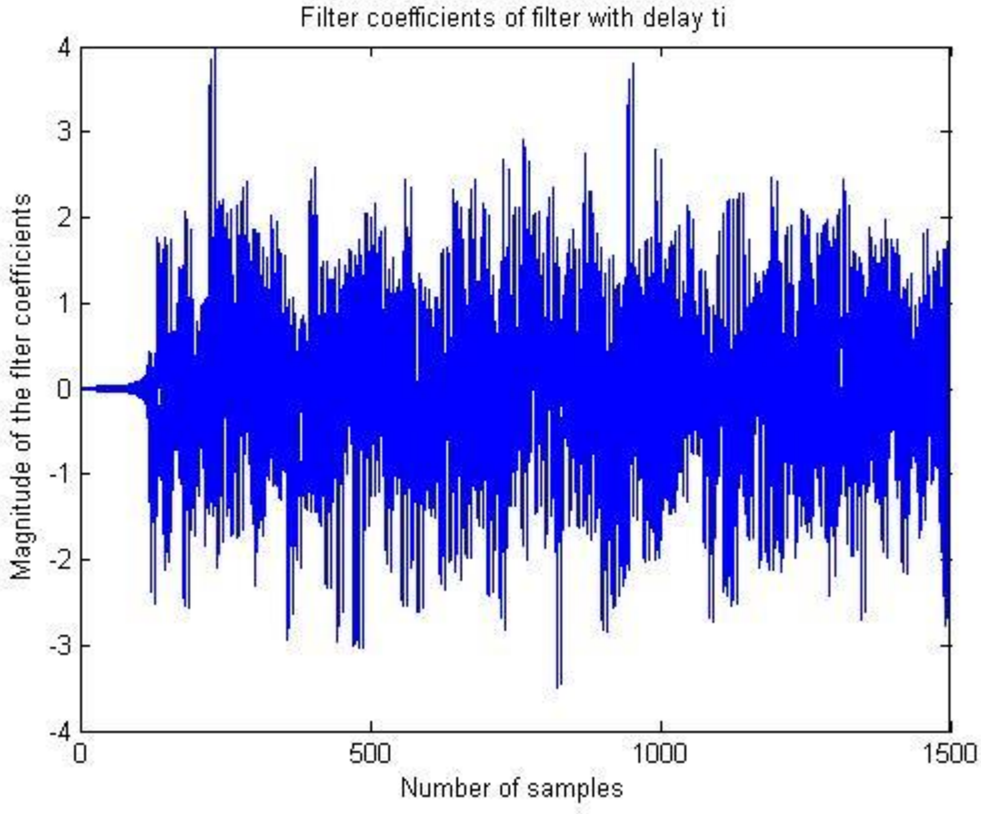


figure 4.1: filter coefficients of filter with delay t_i

Description:

- The figure 4.1 shows the magnitude of the filter coefficients of the filter with delay t_0.
- The Sinc functions are used to produce the required delay as explained in equations (17) and (18).

4.2 RESULT 2

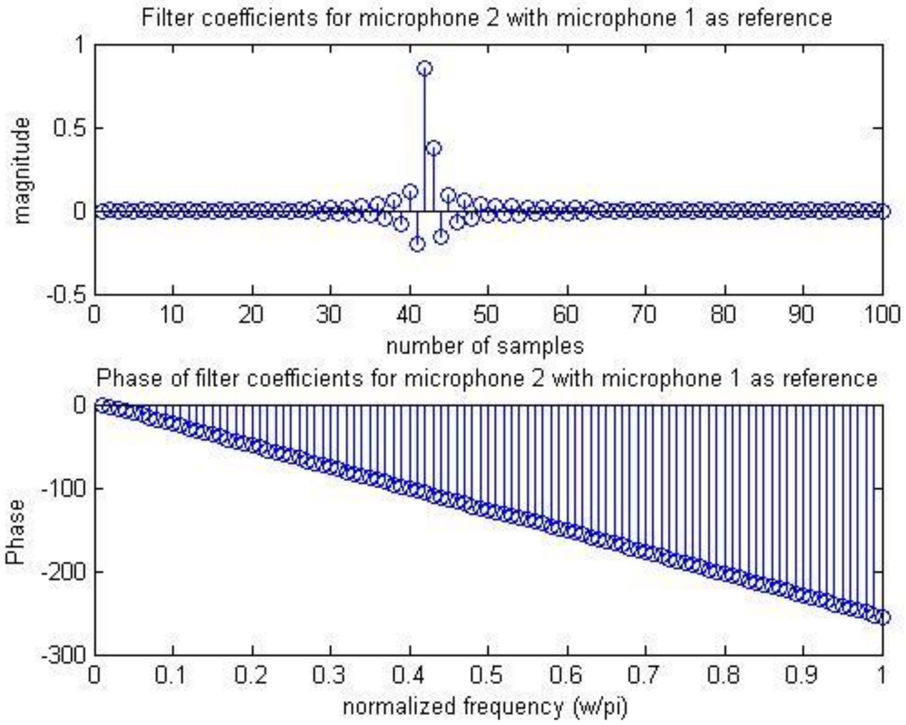


figure 4.2: filter coefficients for microphone 2.

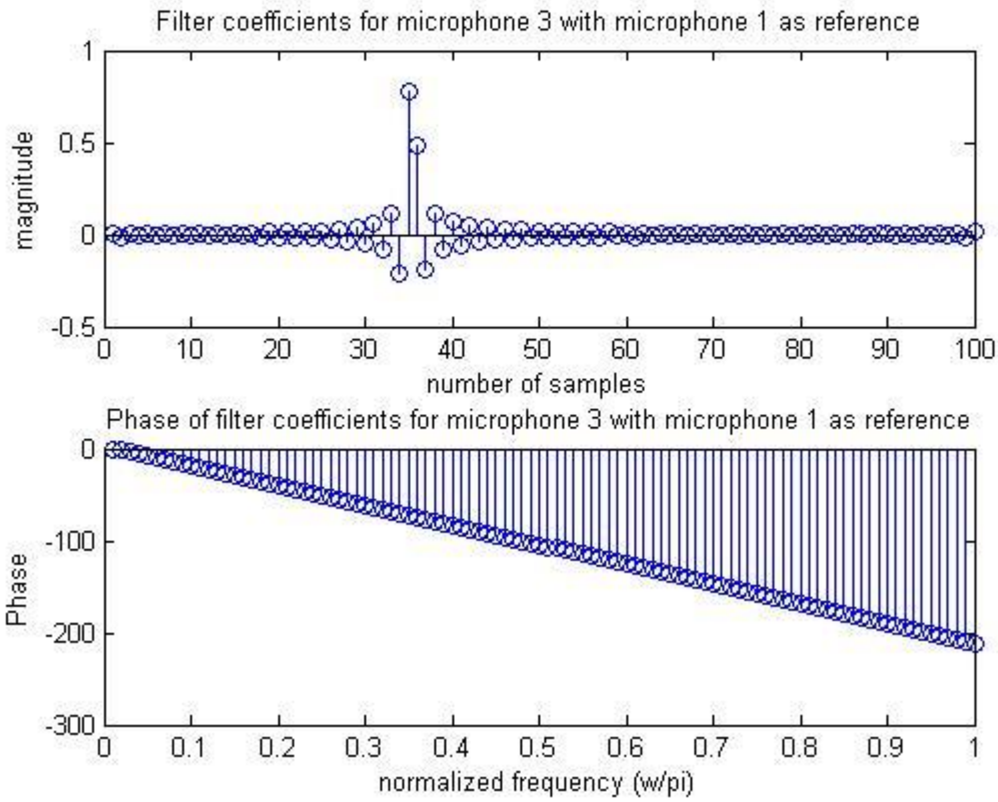


figure 4.3: filter coefficients for microphone 3.

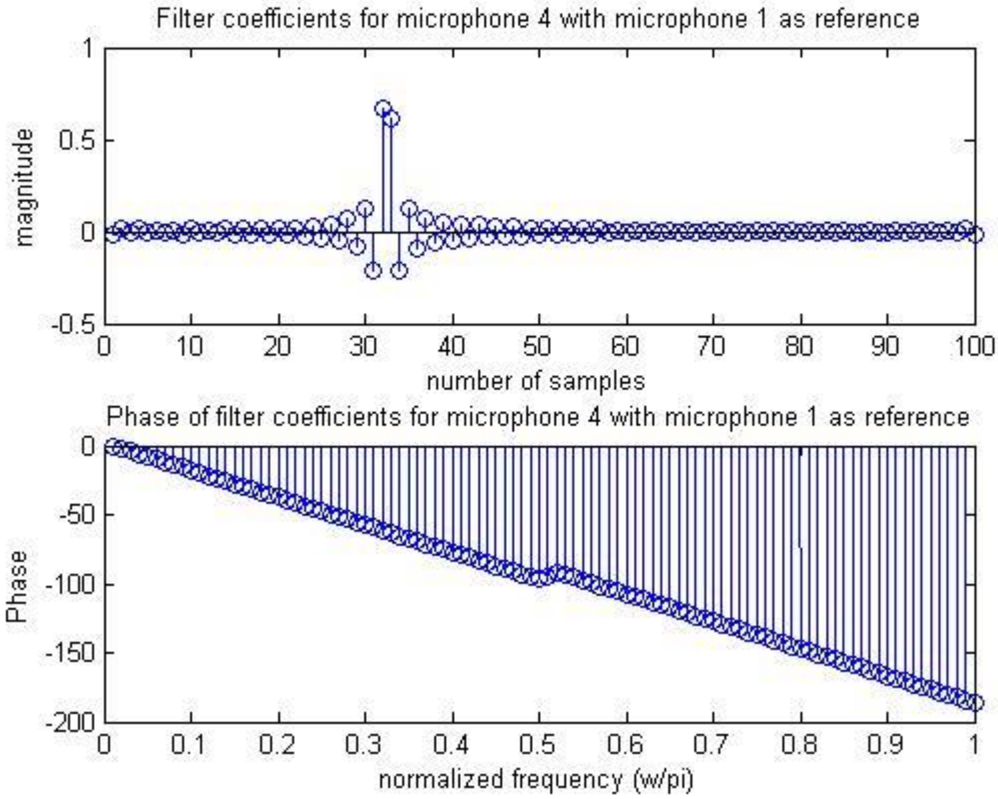


figure 4.4: filter coefficients for microphone 4.

Description:

- Figures 4.2, 4.3 and 4.4 show the filter coefficients of microphones 2, 3 and 4 placed at (3,0), (6,0) and (9,0) respectively when microphone 1 placed at (0,0) is set as reference microphone.
- It can be observed that the magnitude of the filter coefficients is obtained as a sinc function since the shape of the curve is a sinc.
- It can be observed that the phase of the filter coefficients obtained is approximately linear.

4.3 RESULT 3

- Delay estimation:

- The theoretical delays for the microphones 2,3 and 4 calculated in the section 3.3.2 are obtained as:
  Theoretical delays= 0.0073, 0.0131, 0.0155 seconds respectively
- The corresponding practical delays calculated using least square method are obtained as:
  Practical delays = 0.0080, 0.0138, 0.0172 seconds respectively.
- It can be observed that the theoretical and practical delays are approximately equal.

- Location of the sound source:

- The position of the sound source is assumed as (9,5) meters.
- By using the steepest descent algorithm the position of the sound source is obtained as:
- X-coordinate: 9.2368 meters.
- Y-coordinate: 4.2976 meters.
- It can be observed that the assumed values and the obtained values are nearly equal.
- Hence, the delays are estimated and the sound source is localized.

RESULT 4

Variation of convergence with number of iterations:

When number of microphones = 3, distance between microphones = 3 meters.

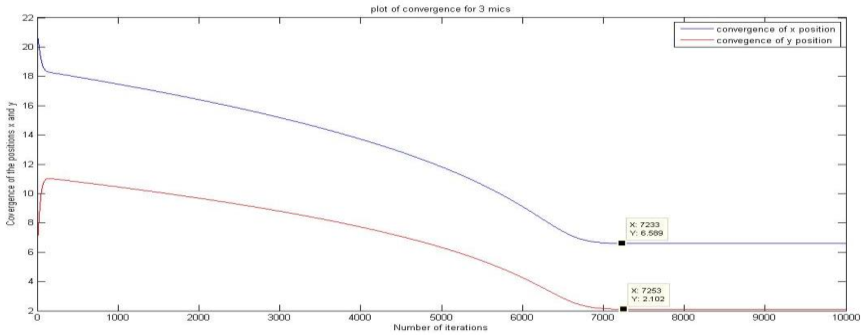


figure 4.5: variation of convergence with number of iterations when microphones=3

When number of microphones=4, distance between microphones = 3 meters.

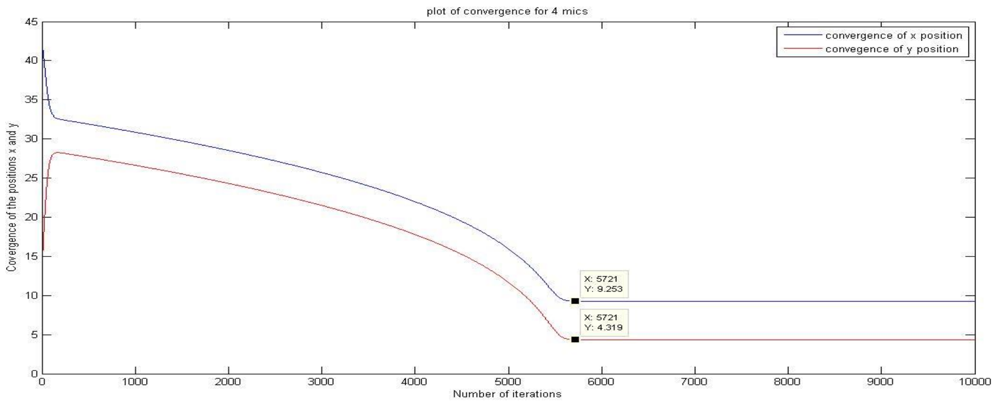


figure 4.6: variation of convergence with number of iterations when microphones=4

When the number of microphones are taken as 5:

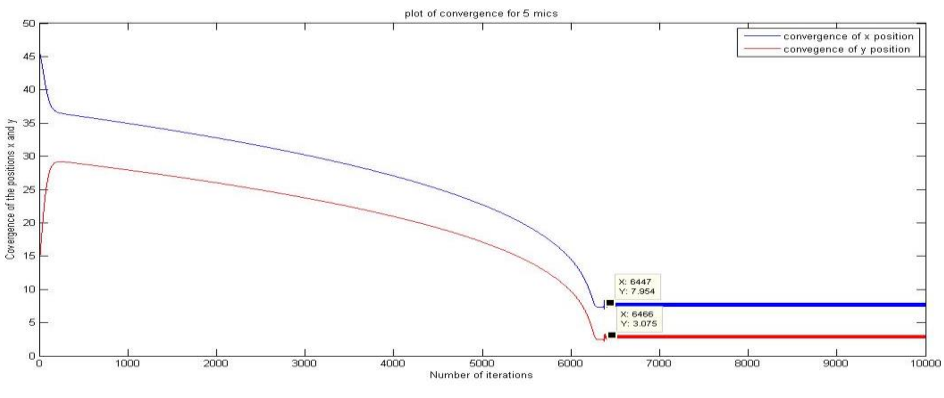


figure 4.7: variation of convergence with number of iterations when microphones=5 and microphone spacing is 2 meters

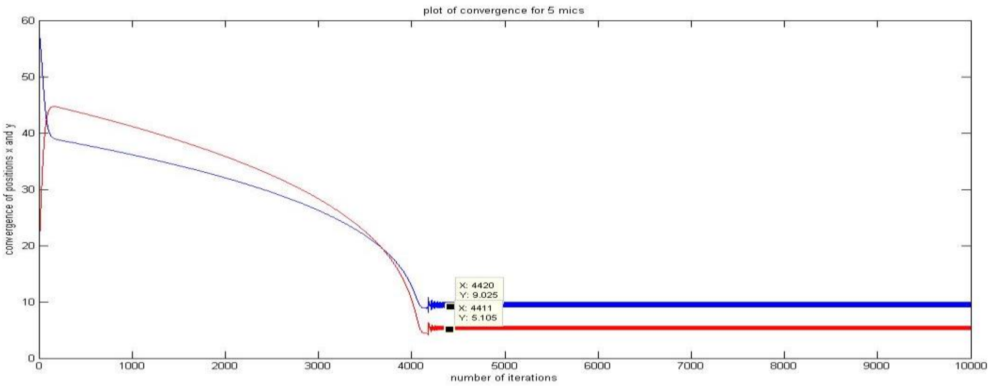


figure 4.8: variation of convergence with number of iterations when microphones=5 and microphone spacing is 2.6 meters

When the number of microphones are increased to 6and the spacing between the microphones is decreased to 2 meters and 2.2 meters respectively.

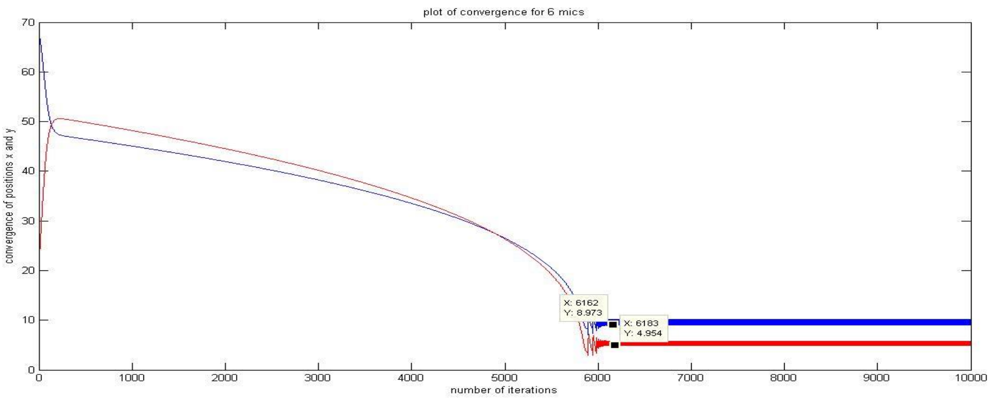


figure 4.9: variation of convergence with no.of iterations when microphones=6 and mic spacing is 2 meters.

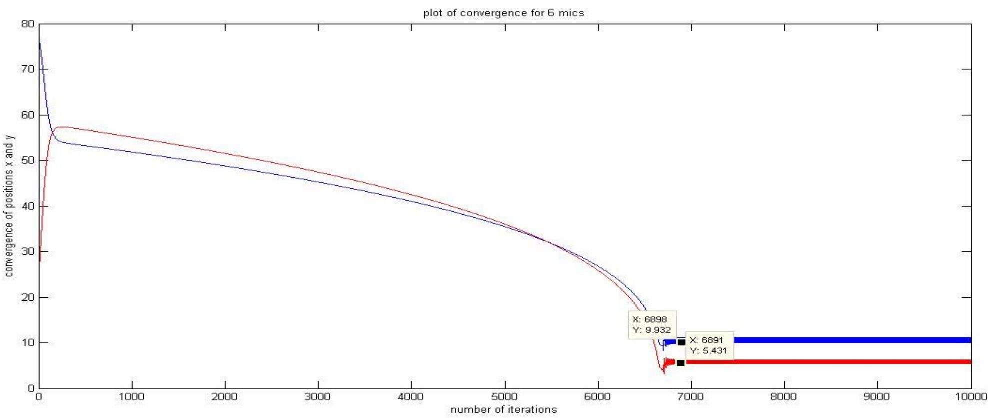


figure 4.10: variation of convergence with no.of iterations when microphones=6 and mic spacing is 2.2 meters.

CHAPTER 5: CONCLUSIONS AND FUTURE SCOPE

5.1 CONCLUSIONS:

- The filter coefficients are in the form of Sinc pulse that is shifted according to the delays.

- The estimate of the time delays of different microphones used are obtained using Least Square method.
- The obtained time delays are used to localize the sound source by implementing Steepest Descent algorithm

  From the figures 4.5 and 4.6, it can be concluded that:
- The estimation of source location is accurate for 4 microphones rather than 3 microphones since the distance between the third microphone and the source is shortest.
- The source can also be accurately localized when the number of microphones are 5 and distance between microphones is 2.6 meters.
- The source can also be accurately localized when the number of microphones are 6 and distance between microphones is 2.2 meters.
- As the number of microphones increases, the number of iterations required to localize the source decreases.
- Thus, the output also depends on the number of microphones used. As the number of microphones increases:
- The rate of convergence is high.
- The estimation of the delay is accurate.
- If the microphones are closely spaced, the number of microphones should be increased so as to obtain accuracy in the estimation as observed in figures 4.7 and 4.9 i.e. estimation is accurate for 6 mics rather than 5 microphones.

5.2 FUTURE SCOPE:

- This project can be further extended by designing an algorithm when:
  - An array of microphones is not linear.
  - Spacing between the microphones is different.
  - The plane is three dimensional.
- The technique of Sound Source Localization can be implemented to track and detect noise producing objects, designing robots, speech recognitions etc.

REFERENCES:

[1] http://www.dspguide.com/ch11/2.htm

[2] Statistical Digital Signal Processing and Modelling by Monson H. Hayes, year:1996, page no:505

[3] Statistical Digital Signal Processing and Modelling by Monson H. Hayes, year:1996, page no:499

[4] http://www.sandv.com/downloads/1006lans.pdf

[5] ftp://ftp.cc.gatech.edu/pub/groups/gvu/tr/2004/04-20.pdf

[6]https://books.google.se/books?id=iDHgboYRzmgC&pg=PA23&lpg=PA23&dq=Noisy+environments+tends+to+decrease+the+reliability&source=bl&ots=jb9KzWQneI&sig=HiS5AAsL-EQ-sE9rH77AT833pC0&hl=en&sa=X&ved=0CDoQ6AEwBWoVChMI55zL9YzbxwIVhY0sCh0ihQc1#v=onepage&q=Noisy%20environments%20tends%20to%20decrease%20the%20reliability&f=false

[7]http://www.sciencedirect.com/science/article/pii/S0960982299802279

[8]https://www.pa.msu.edu/acoustics/rooms1.pdf

[9]http://research.microsoft.com/en-us/um/people/yongrui/ps/icme03.pdf

[10] http://asmp.eurasipjournals.com/content/pdf/1687-4722-2013-27.pdf

[11]https://team.inria.fr/perception/research/geometric-sound-source-localization/

APPENDIX

```
clc;
clear all;
close all;
source = randn(1500,1);                                     %taking a
random source as source
source_X = 9;                                               %assuming
the source x location
source_Y = 5;                                               %assuming
the source y location
Mics = 4;                                                  %number of
microphones
mic_spacing = 3;                                            %distance
between microphones
c = 343;                                                    %speed of
sound
arraywidth = (Mics - 1)*mic_spacing;                       %width of array
of microphones
mic_X = linspace(0,arraywidth,Mics);                       %x locations of
the mics
mic_Y = zeros(1,Mics);                                     %y locations of
the mics
fs = 1200;                                                  %sampling
frequency
%distance of each microphone from the source
dist = sqrt(((source_X-mic_X).^2)+((source_Y-mic_Y).^2));
T = (dist/c)*fs;


for i = 1:Mics-1                                            %calculation
of delays in samples
    dt(i) = T(1)-T(i+1)
end
    delays_theo= dt/fs                                      %theoritical
delays

    Dt = round(dt*fs);                                      %samples

    % 1st filter with length 200
    t = [-100:100]; y = sinc(t-T(1));
    s0 = filter(y,1,source);
    Y1 = sinc(t-T(2));
    s1 = filter(Y1,1,source);
    Y2 = sinc(t-T(3));
    s2 = filter(Y2,1,source);
    Y3 = sinc(t-T(4));
    s3 = filter(Y3,1,source);

     figure(1)
     plot(s1);
   hold on;
   plot(s2);
   hold on;
   plot(s3);
    title('Filter coefficients of first filter with delay ti');
   xlabel('Number of samples');
   ylabel('Magnitude');
```

```
    nord = 100;
% 2nd filter with length nord/2

t1 = [-50:50]; y1 = sinc(t1);
    S0 = filter(y1,1,s0);
    S1 = filter(y1,1,s1);
    S2 = filter(y1,1,s2);
    S3 = filter(y1,1,s3);
     mu = 0.01;
  S = [S0 S1 S2 S3];
%magnitude and phase plots using lms algorithm
[A1] = lms(s0,S1,mu,nord);
A= A1(1500,:);
    figure(2)
    subplot(2,1,1)
    stem(A);
    xlabel('number of samples');
    ylabel('magnitude');
    title('Filter coefficients for microphone 2 with microphone 1 as
reference');
    %phase
    q=((pi/nord):(pi/nord):pi);
    p1=unwrap(-angle((fft(A))));
     subplot(2,1,2)
    stem(q/pi,unwrap(angle((fft(A)))));
    xlabel('normalized frequency (w/pi)');
    ylabel('Phase');
    title('Phase of filter coefficients for microphone 2 with microphone 1
as reference');

    [A2] = lms(s0,S2,mu,nord);
B= A2(1500,:);
    figure(3)
    subplot(2,1,1)
    stem(B);
    xlabel('number of samples');
    ylabel('magnitude');
    title('Filter coefficients for microphone 3 with microphone 1 as
reference');
    %phase
    q=((pi/nord):(pi/nord):pi);
    p2=unwrap(-angle((fft(B))));
   subplot(2,1,2)
    stem(q/pi,unwrap(angle((fft(B)))));
    xlabel('normalized frequency (w/pi)');
    ylabel('Phase');
    title('Phase of filter coefficients for microphone 3 with microphone 1
as reference');

     [A3] = lms(s0,S3,mu,nord);
C= A3(1500,:);
    figure(4)
    subplot(2,1,1)
    stem(C);
    xlabel('number of samples');
    ylabel('magnitude');
    title('Filter coefficients for microphone 4 with microphone 1 as
reference');
    %phase
```

```
    q=((pi/nord):(pi/nord):pi);
    p3=unwrap(-angle((fft(C))));
subplot(2,1,2)
    stem(q/pi,unwrap(angle((fft(C)))));
    xlabel('normalized frequency (w/pi)');
    ylabel('Phase');
    title('Phase of filter coefficients for microphone 4 with microphone 1
as reference');

    k1=q'\p1';
 T1=((nord+(-k1))/2)/fs;
  k2=q'\p2';
 T2=((nord+(-k2))/2)/fs;
  k3=q'\p3';
 T3=((nord+(-k3))/2)/fs;
 Tr=[T1 T2 T3]

 %%Steepest descent algorithm to localize the source
 syms x y

 for i=1:Mics-1
 F(:,i) = (sqrt((x-i*mic_spacing)^2+y^2)-sqrt(x^2+y^2));         %defining
the F function
 end

 G_fun =sum(((F+Tr*c)).^2);
                                                                 %gradiant estimate

 Gd1= matlabFunction(diff(G_fun,(x)));
 Gd2=  matlabFunction(diff(G_fun,(y)));

  N=10000;                                                       %number of
iterations

mu=.5;
 mx=1;
 my=1;
 zx=zeros(1,N).';
 zy=zeros(1,N).';
 zx(1)=mx-mu*Gd1(mx,my);
 zy(1)=my-mu*Gd2(mx,my);
 for i=2:N
     zx(i)=zx(i-1)-mu*Gd1(zx(i-1,1),zy(i-1,1));
     zy(i)=zy(i-1)-mu*Gd2(zx(i-1,1),zy(i-1,1));

 end
 display(zx(i));
 display(zy(i))

%%convergence
 figure
 plot(zx);
 hold on;
 plot(zy,'r');
xlabel('number of iterations');
 ylabel('convergence of positions x and y');
 title('plot of convergence for 4 mics');

function [A,E,y] = lms(x,d,mu,nord,a0)
```

```
%%LMS algorithm

X = convm(x,nord);
[M,N] = size(X);

if nargin < 5,   a0 = zeros(1,N);   end

a0   = a0(:).';
y= zeros(1,M);
E=zeros(1,M);
A= zeros(size(X));

y(1)= a0*X(1,:).';

E(1) = d(1) - y(1);

A(1,:) = a0 + mu*E(1)*conj(X(1,:));

if M>1

for k=2:M-nord+1;
    y(k) = A(k-1,:)*X(k,:).';
    E(k) = d(k) - y(k);
    A(k,:) = A(k-1,:) + mu*E(k)*conj(X(k,:));
    end;
end;
```