

**HARDWARE IMPLEMENTATION OF A
WIRELESS IMPULSIVE SOURCE
LOCALIZATION SYSTEM**

BY

HAIDER ALI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

May, 2012

HARDWARE IMPLEMENTATION OF A
WIRELESS IMPULSIVE SOURCE
LOCALIZATION SYSTEM

by

HAIDER ALI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

In Partial Fulfillment of the Requirements
for the degree

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

KING FAHD UNIVERSITY
OF PETROLEUM & MINERALS

Dhahran, Saudi Arabia


MAY 2012

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **HAIDER ALI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

Thesis Committee



Dr. Mohammad S. Sharawi (Ad-
viser)

(Co-adviser)

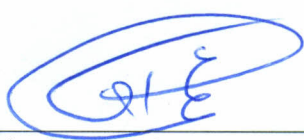


Dr. Tareq Y. Al-Naffouri (Mem-
ber)




Dr. Mohamed Mohandes (Mem-
ber)

(Member)



Dr. Ali A. Al-Shaikhi
Department Chairman



Dr. Salam A. Zummo
Dean of Graduate Studies



Date



Dedicated to my brothers, Uncles and Sir Atta-Ur-Rehman and above all to my late parents who supported me throughout my academic life both morally and financially. I love you all.

ACKNOWLEDGMENTS

In the name of Allah, the Most Beneficent, the Most Merciful

All praises be to Allah (The One and The Only Creator of everything) for His limitless blessings. May Allah bestow peace and His choicest blessings on His last prophet, Hazrat Muhammad (Peace Be Upon Him), his family (May Allah be pleased with them), his companions (May Allah be pleased with them) and his followers.

I would like to express my gratitude to King Fahd University of Petroleum and Minerals (KFUPM) for providing me with the opportunity to benefit from its esteemed environment. Staying for the last two and a half year at KFUPM was the most enjoyable and learning experience in my life. I also express my deep appreciation to the faculty of KFUPM who imparted valuable knowledge, guidance and support in the pursuit of my successful graduate studies. I also want to express my appreciation towards King Abdulaziz City of Science and Technology (KACST) for supporting the research work through generous funding.

I express my deep and profound gratitude to my thesis committee chairman and adviser Dr. Mohammad S. Sharawi for his consistent guidance, support and patience that led my way through the successful completion of my research work.

He spared no single opportunity that could be helpful, both morally and technically, in achieving my goal of a successful and fruitful research work. Spending time with him provided me with valuable professional and technical experience. I am very grateful and honored for the technical help provided to me by my thesis committee member Dr. Tareq Y. Al-Naffouri in understanding the mathematical concepts and background of my thesis work. Many of his suggestions really helped us in the practical implementation of my research work. I also express my profound gratitude to Dr. Mohamed Mohandes for being part of my thesis committee.

I also express my appreciation and gratitude to Mr. Ahmad Abdul Quadeer for helping me understand a portion of the mathematical background of my thesis work and his support during the implementation of my thesis work. I am extremely grateful to Mr. Amin-ud-Din for letting me benefit from his experience in this area of research. I express my gratitude towards Mr. Rifaqat Hussain for the great help he provided during the practical experiments of my research. I also acknowledge and admire the moral support given to me by my friends who made my stay at KFUPM enjoyable and memorable. Few of them are Khwaja Muhammad, Naveed Iqbal, Ahmed Raza, Hussain Ali, and many others whom I shall not be able to name here.

Finally, I would like to express my thanks, respect and deep love towards my parents, uncles and brothers without whom continuous support I would not have achieved my goals. May Allah help us in following Islam according to Quran and

Sunna!(Ameen)

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xiv
ABSTRACT (ENGLISH)	xxii
ABSTRACT (ARABIC)	xxiv
NOMENCLATURE	xxv
CHAPTER 1. INTRODUCTION	1
1.1 Background	2
1.2 Thesis Contributions	4
CHAPTER 2. INTRODUCTION TO ACOUSTICS	7
2.1 Acoustics Basics and Terminologies	7
2.1.1 Properties of Waves	10
2.2 Indoor Acoustics	13
2.2.1 Sound Absorption and Reverberation	13
2.2.2 Effects of Room Shapes, and Sound Insulation	14
2.3 Outdoor Acoustics	15
2.3.1 Spreading Losses	16
2.3.2 Atmospheric Absorption	17
2.3.3 Ground Effects	18
2.3.4 Wind and Temperature Effects on Outdoor Sound	19

2.4	Conclusion	19
CHAPTER 3. LITERATURE REVIEW ON ACOUSTIC SOURCE		
LOCALIZATION 21		
3.1	Introduction	22
3.2	Concept of Direction of Arrival (DOA) and Time Difference of Arrival (TDOA)	22
3.2.1	DOA Technique	22
3.2.2	TDOA Technique	27
3.3	Signal Models	31
3.3.1	Single Source Free Field Model	31
3.3.2	Single Source Reverberant Model	32
3.4	Time Delay Estimation (TDE) Techniques	33
3.4.1	Generalized Cross-Correlation Techniques	33
3.4.2	Eigenvector-Based Techniques	37
3.5	Localization Schemes	40
3.5.1	Conventional Localization Schemes	40
3.5.2	Localization Schemes Utilizing CS and Orthogonal Clustering algorithms	42
3.6	Conclusion	44
CHAPTER 4. WIRELESS SENSOR NETWORK BASED SOLUTION AND ISSUES 45		
4.1	Introduction	45
4.2	Related Work	47
4.3	High Level Structure of the Application	48
4.4	TinyOS 2.x for WSN	50
4.5	Acoustic Signal Acquisition Implementation	51
4.5.1	Synchronization	52
4.5.2	Sampling	54
4.5.3	Flash Reading/Writing	56

4.5.4	Radio communication	56
4.5.5	Serial communication and MATLAB interfacing	58
4.5.6	Overall Structure of the Application	58
4.6	Results and Discussion	66
4.6.1	Error Souces and Uncertainties	66
4.6.2	An Example Experiment	76
4.7	Conclusion	79
 CHAPTER 5. THE ORTHOGONAL CLUSTERING ALGO-		
RITHM AND EXPERIMENTAL SETUP		80
5.1	The Orthogonal Clustering (OC) Algorithm	81
5.1.1	Problem Development	81
5.1.2	The OC Algorithm	83
5.2	Experimental Scenarios and Sensors Geometries	88
5.2.1	TDOA Geometries	90
5.3	Hardware Systems	98
5.3.1	RevoLabs [®] Wireless HD Microphone System	99
5.3.2	VocoPro [®] UHF8800 Wireless Microphone System	99
5.3.3	Data Acquisition Setup	101
5.3.4	Computing Platform	105
5.4	Conclusions	107
 CHAPTER 6. INDOOR EXPERIMENTS AND RESULTS		108
6.1	Parametric Analysis of the OC Algorithm	110
6.1.1	Effect of Ψ on the results	111
6.1.2	The index of CIR used to calculate the TDOA	118
6.1.3	Effect of Probability of Impulses (p)	119
6.1.4	Signal Variance (σ_{imp})	120
6.1.5	Noise Variance (N_0)	121
6.1.6	Conclusion of This Section	123
6.2	Results using RevoLabs Wireless Microphone System	123

6.2.1	Pyramid Geometry Experiments and Results	124
6.2.2	Circular Geometry Experiments and Results	131
6.2.3	Rhombus Geometry Experiments and Results	138
6.2.4	Observations for the Experiments using RevoLabs System	144
6.3	Results using VocoPro Wireless Microphone System	148
6.3.1	Pyramid Geometry Experiments and Results	149
6.3.2	Circular Geometry Experiments and Results	154
6.3.3	Rhombus Geometry Experiments and Results	162
6.3.4	Observations for the Experiments using VocoPro System .	168
6.4	Conclusion	172
CHAPTER 7. OUTDOOR EXPERIMENTS AND RESULTS		174
7.1	Experimental Scenarios	174
7.2	Results using RevoLabs System	176
7.2.1	Pyramid Geometry Experiments and Results	177
7.2.2	Circular Geometry Experiments and Results	181
7.2.3	Rhombus Geometry Experiments and Results	185
7.2.4	Observation for RevoLabs Outdoor Experiments	188
7.3	Results using VocoPro System	190
7.3.1	Pyramid Geometry Experiments and Results	190
7.3.2	Circular Geometry Experiments and Results	194
7.3.3	Rhombus Geometry Experiments and Results	198
7.3.4	Observations for the VocoPro Outdoor Experiments	202
7.4	An Abstract Observation of All Experimental Results	204
7.5	Conclusions	205
CHAPTER 8. CONCLUSIONS AND FUTURE WORK		207
8.1	Conclusions	207
8.2	Future Work and Recommendation	208
REFERENCES		210

LIST OF TABLES

5.1	Source locations for Pyramid geometry experiments	92
5.2	Source locations for Circular geometry experiments	95
5.3	Source locations for Rhombus geometry experiments	98
6.1	Results for 8 kHz case of Triangular geometry when array was placed in the center of the hall using Ψ_0 . All the measurements are in meters	112
6.2	Results for the case of Ψ_1	113
6.3	Results for the case of Ψ_2	115
6.4	Results for the case of Ψ_3	115
6.5	Results for the case of Ψ_4	117
6.6	Index choice for TDOA calculation	118
6.7	Effect of change in p on the results of the algorithm	120
6.8	Effect of change in σ_{imp} on the results of the algorithm	121
6.9	Effect of change in N_0 on the results of the algorithm	122
6.10	Pyramid indoor results using RevoLabs at 8 kHz for the case when the array was placed in the center of the hall. Fast Machine was used.	126
6.11	Pyramid indoor results using RevoLabs at 8 kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.	128
6.12	Pyramid indoor results using RevoLabs at 4 kHz for the case when the array was placed in the center of the hall. Slow Machine was used.	129

6.13	Pyramid indoor results using RevoLabs at 4 kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.	129
6.14	Circular indoor results using RevoLabs at 8 kHz for the case when the array was placed in the center of the hall. Fast Machine was used.	134
6.15	Circular indoor results using RevoLabs at 8 kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.	134
6.16	Circular indoor results using RevoLabs at 4 kHz for the case when the array was placed in the center of the hall. Slow Machine was used.	135
6.17	Circular indoor results using RevoLabs at 4 kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.	136
6.18	Rhombus indoor results using RevoLabs at 8 kHz for the case when the array was placed in the center of the hall. Fast Machine was used.	141
6.19	Rhombus indoor results using RevoLabs at 8 kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.	141
6.20	Rhombus indoor results using RevoLabs at 4 kHz for the case when the array was placed in the center of the hall. Slow Machine was used.	142
6.21	Rhombus indoor results using RevoLabs at 4 kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.	142
6.22	Pyramid indoor results using VocoPro System at 8kHz for the case when the array was placed in the center of the hall. Fast Machine was used.	151
6.23	Pyramid indoor results using VocoPro System at 8kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.	153

6.24	Pyramid indoor results using VocoPro System at 4kHz for the case when the array was placed in the center of the hall. Slow Machine was used.	154
6.25	Pyramid indoor results using VocoPro System at 4kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.	154
6.26	Circular indoor results using VocoPro System at 8kHz for the case when the array was placed in the center of the hall. Fast Machine was used.	157
6.27	Circular indoor results using VocoPro System at 8kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.	159
6.28	Circular indoor results using VocoPro System at 4kHz for the case when the array was placed in the center of the hall. Slow Machine was used.	160
6.29	Circular indoor results using VocoPro System at 4kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.	160
6.30	Rhombus indoor results using VocoPro System at 8kHz for the case when the array was placed in the center of the hall. Fast Machine was used.	163
6.31	Rhombus indoor results using VocoPro System at 8kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.	164
6.32	Rhombus indoor results using VocoPro System at 4kHz for the case when the array was placed in the center of the hall. Slow Machine was used.	166
6.33	Rhombus indoor results using VocoPro System at 4kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.	168

7.1	Pyramid Outdoor results using RevoLabs at 8kHz. Fast Machine was used.	178
7.2	Pyramid Outdoor results using RevoLabs at 4kHz. Slow Machine was used.	180
7.3	Circular Outdoor results using RevoLabs at 8kHz. Fast Machine was used.	182
7.4	Circular Outdoor results using RevoLabs at 4kHz. Slow Machine was used.	184
7.5	Rhombus Outdoor results using RevoLabs at 8kHz. Fast Machine was used.	186
7.6	Rhombus Outdoor results using RevoLabs at 4kHz. Slow Machine was used.	187
7.7	Pyramid Outdoor results using VocoPro at 8kHz. Fast Machine was used.	192
7.8	Pyramid Outdoor results using VocoPro at 4kHz. Slow Machine was used.	193
7.9	Circular Outdoor results using VocoPro at 8kHz. Fast Machine was used.	196
7.10	Circular Outdoor results using VocoPro at 4kHz. Slow Machine was used.	197
7.11	Rhombus Outdoor results using VocoPro at 8kHz. Fast Machine was used.	200
7.12	Rhombus Outdoor results using the VocoPro at 4kHz. Slow Machine was used.	201
7.13	Abstract Observations (accuracy-wise) of Indoor Results	205
7.14	Abstract Observations (accuracy-wise) of Outdoor Results	205

LIST OF FIGURES

2.1	A Basic Sine Wave	10
3.1	Sensor array in the far-field of a radiating source	24
3.2	Sensor array in the far-field of a radiating source	28
4.1	An example Wireless Sensor Network	49
4.2	Block Diagram representation of a TinyOS application	52
4.3	User-defined structure for the payload of radio packet	57
4.4	Commands and states for BaseStation and nodes respectively . .	60
4.5	The structure of sync application	62
4.6	Analyzing the arbitrariness of ReadStream and BlockWrite	69
4.7	Analyzing the effect of no. of sync message/s on synchronization, (a) 1000 mSec case, (b) 500 mSec case.	71
4.8	Analyzing the effect of no. of samples on synchronization, (a) 4096 samples case, (b) 8192 samples case	74
4.9	Nodes configuration for the experiment, (a) Front View, (b) Side View.	77
4.10	Results for the experiment conducted in Fig. 4.9	78
5.1	Problem formulation for OC	83
5.2	Flowchart of the OC algorithm showing the main steps	86
5.3	Different Scenarios Considered in the Experimentation	89

5.4	TDOA Pyramid Geometry with VocoPro System. (a) Pyramid Geometry 3D plot (b) Pyramid Geometry 2D plot (c) Pyramid Geometry, Indoor Experiment with VocoPro System	91
5.5	Selection of source locations of Pyramid geometry for experimentation	92
5.6	TDOA Circular Geometry with RevoLabs System. (a) Circular Geometry 3D plot (b) Circular Geometry 2D plot (c) Circular Geometry, Indoor Experiment with RevoLabs System	94
5.7	Selection of source locations of Circular geometry for experimentation	95
5.8	TDOA Rhombus Geometry with RevoLabs System. (a) Rhombus Geometry 3D plot (b) Rhombus Geometry 2D plot (c) Rhombus Geometry, Indoor Experiment with RevoLabs System	97
5.9	Selection of source locations of Rhombus geometry for experimentation	98
5.10	RevoLabs HD Wireless Microphone System	100
5.11	VocoPro UHF8800 Wireless Microphone System	101
5.12	MCC's DaqBoard/2000 Data Acquisition PCI Card	102
5.13	MCC's DBK202 Expansion Card	103
5.14	Signal acquired from DaqBoard/2000 with channel 0 connected to a microphone and channel 1-7 unconnected to any signal but are open.	104
5.15	MCC's USB1608FS USB Data Acquisition Board	104
5.16	USB1608FS connection to: a) RevoLabs system (b) VocoPro system	106
6.1	Indoor experiments scenarios	109
6.2	Indoor Unloaded Gunshot at 16 kHz near the wall captured using RevoLabs System	113
6.3	Indoor Unloaded Gunshot at 16 kHz using RevoLabs System when Source was in the middle of the hall and near the sensor	114
6.4	Outdoor Unloaded Gunshot at 16 kHz using RevoLabs System when Source was Far from the Sensor	116

6.5	Outdoor Unloaded Gunshot at 16 kHz using RevoLabs System when Source was near to the Sensor	117
6.6	Indoor Pyramid Geometry Setup Using RevoLabs System, (a) Indoor Pyramid Geometry in the Center of the Hall using RevoLabs System, (b) Indoor Pyramid Geometry in a Corner of the Hall using RevoLabs System.	125
6.7	Pyramid Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case	127
6.8	Pyramid Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.	130
6.9	Indoor Circular Geometry Setup Using RevoLabs System, (a) Indoor Circular Geometry in the Center of the Hall using RevoLabs System, (b) Indoor Circular Geometry in a Corner of the Hall using RevoLabs System.	132
6.10	Circular Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case	133

6.11	Circular Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case	137
6.12	Indoor Rhombus Geometry Experimental Setup Using RevoLabs System, (a) Indoor Rhombus Geometry in the Center of the Hall using RevoLabs System, (b) Indoor Rhombus Geometry in a Corner of the Hall using RevoLabs System.	138
6.13	Rhombus Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.	140
6.14	Rhombus Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.	143
6.15	A sample unloaded gunshot acquired using VocoPro system at 16 kHz, amplified 100 times in MATLAB	149
6.16	Indoor Pyramid Geometry Experimental Setup Using VocoPro System, (a) Indoor Pyramid Geometry in the Center of the Hall using VocoPro System, (b) Indoor Pyramid Geometry in a Corner of the Hall using VocoPro System	150

6.17	Pyramid Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d)Height Estimate for the more-reverberant case.	152
6.18	Pyramid Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.	155
6.19	Indoor Circular Geometry Experimental Setup Using VocoPro System, (a) Indoor Circular Geometry in the Center of the Hall using VocoPro System, (b) Indoor Circular Geometry in a Corner of the Hall using VocoPro System.	156
6.20	Circular Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case	158
6.21	Circular Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.	161

6.22	Indoor Rhombus Geometry Experimental Setup Using VocoPro System, (a) Indoor Rhombus Geometry in the Center of the Hall using VocoPro System, (b) Indoor Rhombus Geometry in a Corner of the Hall using VocoPro System.	162
6.23	Rhombus Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case	165
6.24	Rhombus Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case,, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case	167
7.1	Satellite image of the outdoor experimental environment	175
7.2	Outdoor experimental scenarios	176
7.3	Pyramid geometry using RevoLabs system outdoor	177
7.4	Pyramid Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	179
7.5	Pyramid Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	180
7.6	Circular geometry using RevoLabs system outdoor	181

7.7	Circular Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	183
7.8	Circular Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	184
7.9	Rhombus Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	185
7.10	Rhombus Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z -coordinate of the estimated source locations.	187
7.11	Pyramid geometry using VocoPro system outdoor	191
7.12	Pyramid Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	192
7.13	Pyramid Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	194
7.14	Circular geometry using VocoPro system outdoor	195
7.15	Circular Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	196

7.16	Circular Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	198
7.17	Rhombus geometry using VocoPro system outdoor	199
7.18	Rhombus Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	200
7.19	Rhombus Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.	202

THESIS ABSTRACT

NAME: Haider Ali
TITLE OF STUDY: Hardware Implementation of a Wireless Impulsive Source
Localization System
MAJOR FIELD: Electrical Engineering
DATE OF DEGREE: May 2012

Acoustic source localization systems are becoming an essential part of any modern security system and their accuracy and performance play an important role in the overall structure of such systems. Any acoustic source localization system should be robust enough to work in any real situation like in an indoor or an outdoor environment.

This thesis work focuses on the implementation of an impulsive acoustic source localization system using a new algorithm known as Orthogonal Clustering (OC) by utilizing different wireless signal acquisition hardware and devices. The implementation is carried out using both this new algorithm (i.e. OC) and the conventional Cross Correlation (CC) for comparison purposes. Three different sensor geometries were considered for implementation to observe the effect of the sensor

geometry on the performance of the localization system. Three different wireless hardware systems were used to implement the localization system to verify, compare and analyze the performance of the new algorithm as well as the CC method.

A practical and realistic approach is taken for the implementation of the localization system by implementing it in three-dimensions (3D). To check the consistency and accuracy of the new algorithm as well as of the hardware system the implementation was carried out both in an indoor and outdoor environments. To further verify the consistency of the system in the indoor environment, experiments were carried out both in a less-reverberant environment like the center of a hall and a more-reverberant environment like in a corner of the hall. To analyze the system for computational complexity versus the performance tradeoff the system was implemented for signal acquisition at different sampling rates.

From experiments it was observed that the Pyramid geometry of the sensors was the best among all the geometries accuracy wise. In indoor less-reverberant environment the performance of the OC algorithm was better than the CC algorithm. However, in outdoor environment the CC algorithm produces more accurate results than the OC. Furthermore, the runtime of the CC algorithm (less than a second) is much less than the OC algorithm (more than 10 seconds). The RevoLabs system produces more accurate and consistent results than the VocoPro system.

Keywords: *Acoustic Source Localization, Indoor Source Localization, Outdoor Source Localization, Impulsive Acoustics, Wireless Sensor Networks, Commercial off the Shelf Devices*

خلاصة

الاسم: حيدر على

عنوان الرسالة: بناء نظام لاسلكي لتحديد مكان مصادر الاصوات الانفجارية.

التخصص: الهندسة الكهربائية

تاريخ التخرج: مايو ٢٠١٢

إن إمكانية تحديد مصادر الصوت أصبحت جزء أساسي منه أجهزة الحماية الحديثة وباتت فاعلية ودقة هذه الإمكانية جزء أساسي من هذه الأجهزة. أجهزة تحديد أماكن الصوت يجب أن تكون قادرة على العمل في أي بيئة حقيقية مثل داخل المباني أو خارجها.

هذه الرسالة تركز على بناء نظام لتحديد أماكن الأصوات الانفجارية باستخدام خوارزمية جديدة تدعى المجموعة المتعامدة (Orthogonal Clustering) وتطبيقها على عدة أجهزة ومعدات لمعالجة وتخزين الإشارات. التطبيق يقارب بين صحة ودقة الخوارزمية الجديدة والخوارزمية المعروفة بفحص التماثل (Cross Correlation). عدة أشكال هندسية تم اعتمادها لتوزيع المجسات وأثرها على دقة تحديد مكان الصوت. ثلاثة أجهزة لاسلكية تم استخدامها لتطبيق النظام عملياً وفحص فعاليتها باستخدام الخوارزميات السابق ذكرها.

لقد تم بناء النظام بشكل عملي لتحديد مكان الصوت بأبعاده الثلاث (3D) للتحقق من صحة النظام ونتائجه تم فحصه في داخل المبنى وخارجه. بالإضافة إلى ذلك تم فحصه في عدة أماكن داخل المبنى مثل بعيداً عن الجدران وقريباً منها للتحقق من أثر امتداد الصوت على صحة تحديد مكانه. تم فحص أثر شدة التردد في تخزين الإشارة على صحة ودقة تحديد مكان الصوت أيضاً.

من التجارب تم ملاحظة أدق شكل هندسي في تحديد مكان الصوت. هذا الشكل هو الهرم الثلاثي الأبعاد (Pyramid)، في داخل المبنى. كانت أدق النتائج باستخدام خوارزمية المجموعة المتعامدة. أما في خارج المبنى، فقد كانت أدق النتائج باستخدام فحص التماثل بالنسبة لسرعة حساب مكان الصوت، فقد كانت المدة باستخدام خوارزمية فحص التماثل أسرع بكثير من المجموعة المتعامدة بعشر مرات على الأقل. وكان نظام RevoLab اللاسلكي أدق من نظام VocoPro.

Nomenclature

Abbreviations

AMMSE	:	Approximate Minimum Mean Square Error
AOA	:	Angle of Arrival
CC	:	Cross Correlation
CIR	:	Channel Impulse Response
COTS	:	Commercial off the Shelf
CRLB	:	Cramer-Rao Lower Bound
CS	:	Compressive Sensing
DAQ	:	Data Acquisition
DF	:	Directivity Factor
DFT	:	Discrete Fourier Transform
DI	:	Directivity Index
DOA	:	Direction of Arrival
DTFT	:	Discrete Time Fourier Transform
EDT	:	Early Decay Time
EM	:	Electromagnetic
FPGA	:	Field Programmable Gate Array
FTSP	:	Flooding Time Synchronization Protocol
GCC	:	Generalized Cross Correlation
IASL	:	Impulsive Acoustic Source Localization
IDTFT	:	Inverse Discrete Time Fourier Transform

LS	: Least Squares
MAP	: Maximum A Posteriori
MCU	: Microcontroller Unit
MMSE	: Minimum Mean Square Error
MUSIC	: Multiple Signal Classification
OC	: Orthogonal Clustering
PHAT	: Phase Transform
RBS	: Reference Broadcast Synchronization
RF	: Radio Frequency
RIR	: Room Impulse Response
RSS	: Received Signal Strength
SCOT	: Smoothed Coherent Transform
SIL	: Sound Intensity Level
SIMO	: Single Input Multiple Output
SNR	: Signal to Noise Ratio
SPL	: Sound Pressure Level
TDDF	: Time Delay Direction Finding algorithm
TDE	: Time Delay Estimation
TDOA	: Time Difference of Arrival
TPSN	: Timing-sync Protocol for Sensor Networks
UWB	: Ultra Wide-Band
WSN	: Wireless Sensor Network

Notations

\mathbf{x}	:	Channel Impulse Response
$\epsilon_{CC.2D}$:	2D error for CC algorithm
$\epsilon_{OC.2D}$:	2D error for OC algorithm
$\epsilon_{CC.3D}$:	3D error for CC algorithm
$\epsilon_{OC.3D}$:	3D error for OC algorithm
Z	:	Acoustic Impedance
I	:	Acoustic Intensity or Identity Matrix depending on the context
α_n	:	Attenuation factor for path n
ϕ	:	Azimuth angle of the source at origin
Z_0	:	Characteristic Impedance of an acoustic medium
\mathbf{n}	:	Complex White Gaussian Noise
Φ	:	Design matrix for the Least Squares method
Ψ	:	Dictionary or Sensing matrix
$\vec{\mathbf{k}}$:	Direction of Arrival Vector
θ	:	Elevation angle of the source at origin
$\hat{\mathbf{x}}$:	Estimated channel impulse response
x	:	Estimated source location
$\mathbb{E}[x y]$:	Expected value of x given y
f	:	Frequency
$\psi_{y_1 y_2}$:	Generalized Cross Spectrum
P	:	Maximum number of clusters

a_n	:	Location of microphone n
$\hat{\mathbf{x}}_{MMSE}$:	MMSE estimate of the CIR
$u(k)$:	Noise signal
\mathbf{v}	:	Noise vector
b	:	Observation matrix for the Least Squares problem
p	:	Probability of impulses in the CIR
k	:	Propagation Constant
Γ	:	Reflection Coefficient
RT_{60}	:	Reverberation Time
x	:	Source location
X_{CC}	:	Source location estimated using CC algorithm
X_{OC}	:	Source location estimated using OC algorithm
$s(k)$:	Source signal
\mathcal{S}	:	Sparsity
c	:	Speed of Sound
N	:	The length of the channel impulse response
M	:	The number of rows in the Dictionary Matrix
\mathbf{y}	:	The observation vector for OC
τ_{ij}	:	Time delay between sensor i and j
TDE_{CC}	:	Time Delay Estimated using CC algorithm
TDE_{OC}	:	Time Delay Estimated using OC algorithm
p_T	:	Total acoustic pressure at a point

- TL : Transmission Loss
- σ_x : Variance of the CIR
- σ_{imp} : Variance of the impulsive signal
- σ_n : Variance of the noise
- λ : Wavelength

CHAPTER 1

INTRODUCTION

The need of acoustic source localization is growing day by day as an integral part of any modern security system. In particular, localization of sources of impulsive nature in hazardous environments like war, natural catastrophes among others requires the need of robust systems specially designed for this kind of situations. Impulsive acoustic source localization can also be helpful in many commercial applications like shopping malls, conference halls etc. where the location of a burst or a gun shot is to be determined. This work will focus on methods to localize an impulsive acoustic source both in indoor and outdoor environments in three dimensions. A comparison among these methods and their real time implementation and performance will be analyzed and discussed.

This chapter serves as an introduction to the topic of this thesis work. Section 1.1 will discuss the background and a little description about this work while Section 1.2 states the objectives of this thesis work.

1.1 Background

Source localization is needed by many applications in engineering and science disciplines [1][2]. Applications ranging from localizing a cell phone user to localizing a sniper fire involves finding the position of the source emanating a signal that can be of electromagnetic or acoustic nature. Increased security issues demand more sophisticated, reliable and robust source localization systems.

Time delay based source localization techniques utilize the time delay that occurs to the signal when it reaches to different sensors at different times. The so called Time Difference of Arrival (TDOA) is the difference in the arrival times of the signal at each sensor. Some algorithms use these TDOA measurements and incorporate them in some mathematical models to directly estimate the source location. These are called TDOA algorithms [3]. Other algorithms take a different approach, they utilize the TDOA information to calculate the Angle of Arrival (AOA) (or Direction of Arrival, DOA) of the signal at each sensor and then utilizing this AOA information they can localize the source. Such algorithms are called AOA algorithms.

Acoustic source localization is becoming an essential technology and its applications are attracting researchers. Localizing the position of the source of gun fire or blast in a war has been studied since World War II [4]. Acoustic source localization systems can be used in malls for security issues or even for daily use, for example, directing the microphone or camera to a certain user in a conference hall or a talk show. Localizing a source that generates an impulsive signal is

equally important and needs the attention of researchers. Impulsive signal source localization systems can be used in military or similar applications.

Although there is plenty of literature available on acoustic source localization, the new technology and advancement in mathematical modeling needs to be incorporated with the existing techniques to take their full advantage to reduce the hardware complexity and energy consumption and enhance the performance of existing systems.

In this work we will use the modern Compressed Sensing (CS) approach to localize impulsive acoustic sources and evaluate its performance. The focus will be on three dimensional source localization using a new algorithm called Orthogonal Clustering (OC) [5] which is a variant of CS. This new algorithm will allow impulsive acoustic source localization by sampling the sensors at reduced rates lower than the Nyquist sampling rate. The work done in [6] already has proved the applicability and feasibility of the algorithm in a two dimensional acoustic source localization environment. Moreover, the algorithm works best (according to the author of [6] in a dense reverberant environment which is the requirement of several practical systems that operate indoors.

In addition, three hardware systems will be used to implement a wireless impulsive acoustic source localization system in 3D. The effect of variations of several parameters on the system performance will be evaluated, such as the indoor and outdoor scenarios, the effect of the number of microphones and the directivity of microphones and sampling rate on the accuracy of the localization system.

Moreover, a performance comparison between a conventional time delay estimation (TDE) technique such as cross correlation (CC) and the new OC based estimation technique will be conducted and the performance of both techniques in the hardware systems implemented will be evaluated for localizing an impulsive acoustic source in 3D.

1.2 Thesis Contributions

The major contributions of this thesis are given below:

1. A Commercial off the Shelf (COTS) Wireless Sensor Network (WSN) realizing a wireless localization system of acoustic impulsive sources was configured and tested. Different issues and error sources were identified and verified through a number of experiments to show that the COTS WSN platforms are not suitable for acoustic source localization. Several suggestions were given with examples from the literature to overcome the shortcomings of the existing WSN hardware and software to make them workable for acoustic source localization.
2. A detailed study on the effects of the orthogonal clustering (OC) algorithm parameters on the accuracy of the time delay estimates (TDE) obtained from it in a reverberant environment was conducted. Such a study is the first to appear for this newly developed algorithm.
3. Two different wireless microphone systems were used for acoustic signal

acquisition and integrated with the hybrid algorithm based on the OC TDE method and time difference of arrival (TDOA) 3D localization method for impulsive acoustic source localization. It was observed that the hardware system has a significant contribution in the accuracy of the system.

4. An extensive experimentation and performance study was carried out both in indoor and outdoor environments to analyze and compare the performance of the 3D impulsive acoustic localization system using two different TDE methods one based on OC and the other was based on the well-known cross correlation (CC) method. In the indoor environment, two locations were examined; at the center of the room and at the corner of the room to represent low and high reverberant environments, respectively. This is the first extensive experimentation study to appear comparing OC with other methods for 3D localization.
5. The effects of the microphone geometry, sampling rates and acquisition systems were investigated in details on the accuracy of impulsive acoustic localization system in 3D. Three different microphone array geometries were used to see the effect of geometry structure on the accuracy and performance of both of the OC and CC TDE methods. It was observed that the Pyramid geometry, due to its special structure, was producing the most accurate results while the Rhombus geometry was producing the least accurate results. In addition, it was observed that in indoors the OC produces better results for 4kHz than CC, while in outdoor the CC was producing better results

than OC at reduced rates.

CHAPTER 2

INTRODUCTION TO ACOUSTICS

Acoustics is a complete science by itself. Acoustic waves generally behave differently in indoor and outdoor environment and thus a comprehensive study is required in each case. This chapter is devoted to provide basic understanding of acoustics waves and their properties and characteristics. Section 2.1 gives a brief introduction to acoustic waves and discuss various terminologies that are used in acoustics. Indoor acoustics are discussed in Section 2.2 and Section 2.3 provides basics about outdoor acoustics. Finally we conclude the chapter in Section 2.4

2.1 Acoustics Basics and Terminologies

Sound is a wave and a wave is a disturbance that propagates through a medium. There are two basic types of waves: longitudinal waves, and transverse waves. Longitudinal waves are waves in which the particle motion in the medium is in

the same direction as the wave is traveling, while transverse waves are waves in which the direction of motion of particles in the medium is normal to the direction of wave.

Acoustic waves (which are longitudinal) are generally pressure variation occurring in the medium due to some vibrating bodies. The total pressure at a point is given by

$$p_T(x, t) = p_0(x, t) + p_1(x, t) \quad (2.1)$$

where p_0 represents the ambient pressure of the fluid and p_1 represents the pressure fluctuation caused by the acoustic field. The types of sounds we encounter cause pressure fluctuations in the range of $10^{-3} - 10 \text{ Pa}$ [4].

Another terminology called acoustic intensity I is also used to describe the sound wave energy and it is given by

$$I = \frac{dP}{dA} \quad (2.2)$$

where dP is the portion of the acoustic power that interacts with the area dA of the detector oriented perpendicular to the direction of the oncoming acoustic wave. The units of acoustic intensity are watts per square meter (W/m^2).

The human ear can generally perceive sound pressures over the range from about $20\mu \text{ Pa}$ up to about 200 Pa [4]. Often acoustic intensity is measured with

respective to a reference value as a ratio called the Sound Intensity Level (SIL)

$$SIL(dB) = 10 \log \frac{I}{I_{ref}} \quad (2.3)$$

where I is the intensity of the sound wave and I_{ref} is a reference intensity. For the intensity of a sound wave in air, the reference intensity is defined to be $I_{ref} = 10^{-12} \text{ W/m}^2$. Another terminology called Sound Pressure Level (SPL) is also widely used and is defined as

$$SPL(dB) = 20 \log \frac{p}{p_{ref}} \quad (2.4)$$

where p is the acoustic pressure and p_{ref} is a reference pressure. For sound in air, the reference pressure is defined as $20 \mu \text{ Pa}$.

Sound waves can also be represented by a sinusoidal equation as shown in Fig. 2.1, e.g.

$$y(x) = A \sin(kx + \phi), \quad (2.5)$$

where A is the amplitude of wave (i. e., the particle displacement) in the y -direction and $k = 2\pi/\lambda$ represents a scaling factor called the wave number. The term ϕ is known as the phase shift because it causes a shifting of the wave profile along the x -axis (forward for a positive phase shift and backward for a negative phase shift). The v in Fig. 2.1 represents the velocity of the wave in the medium.

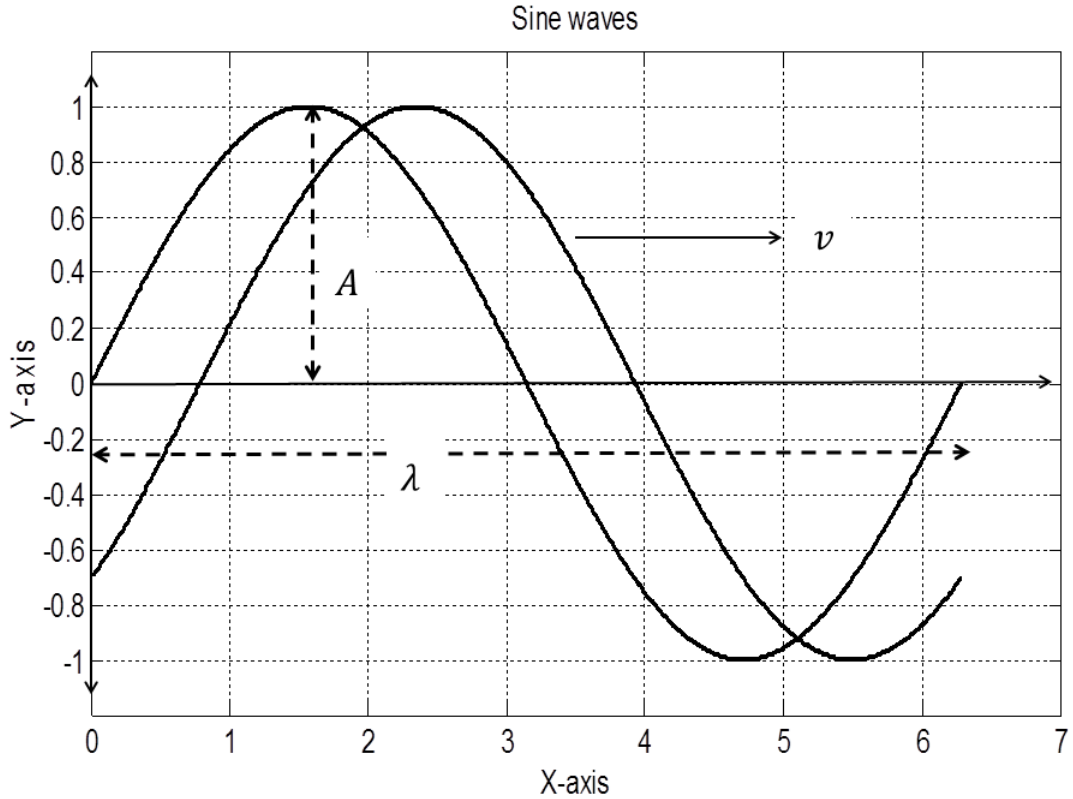


Figure 2.1: A Basic Sine Wave

2.1.1 Properties of Waves

Acoustic Impedance

Every medium has an impedance which causes waves to attenuate. For acoustic waves, the impedance Z is defined as the ratio of sound pressure to particle velocity. The unit for impedance is the Rayl, named in honor of Lord Rayleigh. $1 \text{ Rayl} = 1 \text{ Pa s/m}$. This impedance of the medium is also called the characteristic impedance and usually denoted by Z_0 . In air, the characteristic impedance near room temperature is about, 410 Rayl [4].

Acoustic impedances of media give us a measure of how much energy of a

wave is transmitted into the medium when it travels from one medium to another. When particle velocity and pressure are continuous across the interface between the two media, then the percentage of the energy that is reflected back into the medium is given by

$$\Gamma = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \quad (2.6)$$

where Z_1 and Z_2 are the impedances of the two media and Γ is called the reflection coefficient. The fraction of the energy transmitted into the second medium is given by the Transmission Coefficient: $\tau = 1 - \Gamma$ because 100% of the energy must be divided between τ and Γ .

Refraction

Refraction is a change of the direction of wave propagation as the wave passes from one medium into another across an interface. Snell's law determines the amount of energy in the reflected and transmitted (refracted) waves. All natural waves obey Snell's law. For acoustic waves the proper form of Snell's law is:

$$\frac{\sin(\theta_1)}{v_1} = \frac{\sin(\theta_2)}{v_2} \quad (2.7)$$

where v_1 is the wave velocity in medium 1 and v_2 is the wave velocity in medium 2, θ_1 is the angle that incident wave makes with the normal to boundary between two media in medium one and θ_2 is the angle which the transmitted wave makes with the normal to the same boundary in the second medium.

Reflection

Reflection occurs when a wave travels from one medium into another. If the acoustic impedance of the two media is different, then part of the wave is reflected and some part is transmitted into the medium depending on the acoustic impedances of both media.

Interference

Interference is a phenomenon that occurs when two (or more) waves add together. Consider two sinusoidal acoustic waves with slightly different frequencies and equal amplitudes that arrive at the same point in space, then using superposition principle the total pressure in the medium is

$$p_T(t) = A [\cos(\omega_1 t) + \cos(\omega_2 t)] \quad (2.8)$$

$$= 2A \cos\left(\frac{(\omega_1 - \omega_2)}{2}t\right) \cos\left(\frac{(\omega_1 + \omega_2)}{2}t\right) \quad (2.9)$$

Due to the slight difference in frequencies the two waves can be in phase, causing constructive interference and reinforcing one another. Over some period of time, the frequency difference causes the two waves to go out of phase, causing destructive interference (when $\omega_1 t$ eventually leads $\omega_2 t$ by 180°). The amplitude of the combination will rise and fall in a periodic fashion. This phenomenon is known as the beating of the two waves.

2.2 Indoor Acoustics

Walls of a room reflect sound waves which causes the phenomenon of reflection and room resonance. When multiple echoes in a room combine they produce a phenomenon known as reverberation. Indoor applications (as in our application) should take into consideration these important phenomena to accurately extract the desired information from the composite signal captured at a sensor.

2.2.1 Sound Absorption and Reverberation

Absorption is useful for reducing echo within a room. The absorption coefficient α is used to measure the amount of absorption that a material incurs. The Absorption coefficient is defined as the ratio of absorbed to incident energy. Absorption coefficients vary with frequency.

Reverberation is described by a parameter known as the reverberation time (denoted as RT_{60}) [4]. Physically, RT_{60} is the time (in seconds) that it takes for a sound source to reduce in sound pressure level (within a room) by a factor of 60 dB after that sound source has been silenced. Mathematically, RT_{60} is given by Sabin's equation:

$$RT_{60} = 0.161 \frac{V}{A} \quad (2.10)$$

where V is the room volume in cubic meters and A is the total absorption of the room's surfaces in metric Sabins. Sabin is the unit of total absorption and one Sabin is defined as the total absorption provided by a one square foot piece of material having an absorption coefficient of 1 [4]. RT_{60} can be controlled by i)

changing the room size and by ii) changing the absorption properties of the walls.

Another closely related terminology used to describe reverberation is Early Decay Time (EDT) which is actually the time taken by the sound level to drop from 0 dB to -10 dB [4] and is given by

$$EDT = \frac{60 \text{ dB}}{A(0 \rightarrow -10)\text{dB/Sec}} \quad (2.11)$$

where A represents the attenuation rate of the acoustic signal when it decreases from 0dB to -10dB sound pressure level.

2.2.2 Effects of Room Shapes, and Sound Insulation

Room shapes play an important role in the behavior of acoustic waves inside the room. Different surface structures are used to enhance sound quality inside a hall. Convex surfaces facilitate to diffuse the sound evenly throughout the audience. Concave reflective surfaces focus sound in certain areas and defocus sound from others, causing hot spots where sound is concentrated and dead spots where sound cannot be heard.

The Transmission Coefficient is the ratio of the transmitted to incident sound energy when sound waves encounter a partition or a wall. It is denoted by τ and ranges from 0 to 1. A transmission coefficient of 1 implies that all of the sound energy is transmitted through a partition and 0 means complete reflection.

The Transmission Loss is used to describe the sound insulation, measured in

dB and based on transmission coefficient

$$TL = 10 \log\left(\frac{1}{\tau}\right) \quad (2.12)$$

The transmission loss can be loosely defined as the amount of sound reduced by a partition between a sound source and a listener. The complete sound reduction of a partition between two rooms also takes into account the absorptive characteristics of the listener's room, as follows:

$$SPL_S - SPL_L = TL + 10 \log\left(\frac{A_L}{S}\right) \quad (2.13)$$

where SPL_S is the average sound pressure level in the room enclosing the sound source, SPL_L is the average sound pressure level in the adjacent listener's room, A_L is the total absorption in the listener's room, TL is the transmission loss of the partition between the two rooms, and S is the surface area of the partition between the two rooms.

2.3 Outdoor Acoustics

Most of the outdoor sound experimentations conducted in 16th and 17th century were concerned about sound speed measurement [1][2]. Besides from interests in prediction and control of noise arising from land and air transport, outdoor acoustics has continued to have extensive military applications in source acquisition, ranging and identification [7]. There are several important parameters regarding

the outdoor acoustic signals which are discussed next.

2.3.1 Spreading Losses

When waves travel, their wavefronts spread with distance. The intensity I at a distance r m from an isotropic source, which radiates equally in all direction, is given by [2]

$$I = \frac{P}{4\pi r^2} \quad (2.14)$$

where P is the power of a spherical wavefront of radius r . The relationship between sound pressure level L_p and sound power L_W may be written as

$$L_P = L_W - 20 \log(r) - 11dB \quad (2.15)$$

For an omnidirectional point sound source, (2.15) shows a reduction of $20 \log 2$ dB, i.e., 6 dB per distance doubling in all directions. For a directional source, (2.15) is modified by including the directivity index (DI).

$$L_P = L_W + DI - 20 \log(r) - 11dB \quad (2.16)$$

The DI is $10 \log(DF)$ dB where DF is the directivity factor given by the ratio of the actual intensity in a given direction to the intensity of an omnidirectional source of the same power output. The directivity factor for a point source on a perfectly reflecting plane is 2 and the directivity index is thus 3 dB.

2.3.2 Atmospheric Absorption

A proportion of sound energy is converted to heat as it travels through the air. There are heat conduction, shear viscosity and molecular relaxation losses [5]. For a plane wave, the pressure p at distance x from a position where the pressure is p_0 is given by

$$p = p_0 e^{-\alpha x/2} \quad (2.17)$$

The attenuation coefficient α can be calculated using (2.18) to (2.20)

$$\alpha = f^2 \left[\left(\frac{1.84 \times 10^{-11}}{\left(\frac{T_0}{T}\right) 1/2 \frac{p_s}{p_0}} \right) + \left(\frac{T_0}{T}\right)^{2.5} \left(\frac{0.106080 e^{-3352/T} f_{r,N}}{f^2 + f_{r,N}^2} \right) + \frac{(0.01278 e^{-2239.1/T} f_{r,O})}{f^2 + f_{r,O}^2} \cdot \frac{N_P}{m \cdot atm} \right] \quad (2.18)$$

where f is the frequency, T is the absolute temperature of the atmosphere in degrees Kelvin, $T_0 = 293.15K$ is the reference value of $T(20^\circ C)$, and $f_{r,N}$ and $f_{r,O}$ are relaxation frequencies associated with the vibration of nitrogen and oxygen molecules respectively and are given by:

$$f_{r,N} = \frac{p_s}{P_{s0}} \left(\frac{T_0}{T}\right)^{1/2} \left(9 + 280H e^{-4.17 \left[\left(\frac{T_0}{T}\right)^{1/s} - 1 \right]} \right) \quad (2.19)$$

$$f_{r,O} = \frac{p_s}{P_{s0}} \left(24.0 + 4.04 \times 10^4 H \frac{0.02 + H}{0.391 + H} \right) \quad (2.20)$$

where H is the percentage molar concentration of water vapor in the atmosphere $= \rho_{sat} r_h p_0 / p_s$, r_h is the relative humidity (%); p_s is the local atmospheric pressure and p_0 is the reference atmospheric pressure ($1atm = 1.01325 \times 10^5 Pa$); $\rho_{sat} =$

$10^{C_{sat}}$, where $C_{sat} = -6.8346(T_0/T)^{1.261} + 4.6151$. These formulae give estimates of the absorption of pure tones to an accuracy of $\pm 10\%$ for $0.05 < H < 5$, $253 < T < 323$, $p_0 < 200 \text{ kPa}$. Outdoor air absorption varies through the day and the year.

2.3.3 Ground Effects

Reflection from the ground causes interference with waves above the ground and leads to so called ground effects. A widely used model that is used to investigate the properties of outdoor acoustics makes use of a single parameter, the flow resistivity σ_e , to characterize the ground. Flow resistivity is a terminology that describes the behavior of air when it moves in and out of ground and its unit is Pa sm^{-2} . The propagation constant k and normalized impedance Z are given, in terms of σ_e , as follows

$$\frac{k}{k_1} = \left[1 + 0.0978 \left(\frac{f}{\sigma_e} \right)^{-0.700} - j0.189 \left(\frac{f}{\sigma_e} \right)^{-0.595} \right], \quad (2.21)$$

$$Z = \frac{\rho_1 c_1}{\rho c} = 1 + 0.0571 \left(\frac{f}{\sigma_e} \right)^{-0.754} - j0.087 \left(\frac{f}{\sigma_e} \right)^{-0.732} \quad (2.22)$$

where k_1 is the propagation constant within the surface layer, k is the propagation constant in the air, ρ_1, c_1 are the pressure density and sound velocity within the surface layer, f is the frequency of the sound wave and ρ, c are the pressure density and sound velocity in the air.

2.3.4 Wind and Temperature Effects on Outdoor Sound

Both wind speed and temperature affect the speed of sound in a region. The speed of sound changes with atmosphere temperature which due to the fact that gases expand and contract with changing temperature. Wind speed directly adds or subtracts from sound speed depending on the direction of wind flow and acoustic waves.

In general, the relationship between the speed of sound profile $c(z)$, temperature profile $T(z)$ and wind speed profile $u(z)$ in the direction of sound propagation \vec{z} is given by

$$c(z) = c(0) \sqrt{\frac{T(z) + 273.15}{273.15}} + u(z) \quad (2.23)$$

where T is in $^{\circ}C$ and u, c are in m/s .

2.4 Conclusion

Understanding sound waves behavior in the indoor and outdoor environment is essential for acoustic applications such as acoustic source localization. Indoor acoustic applications require an in-depth understanding of sound waves inside a building, room or hall. In this chapter, the characteristics and properties of sound waves inside a building were discussed. The effects of reflections from the walls of the room and reverberations were presented. The effects of room shapes and designs on indoor acoustic waves were also discussed.

In addition, various important outdoor properties like ground effects, spread-

ing losses, and wind and temperature effects on the acoustic signals were discussed. Atmospheric absorption was also investigated and relevant mathematical expressions were presented.

CHAPTER 3

LITERATURE REVIEW ON

ACOUSTIC SOURCE

LOCALIZATION

Localization of acoustic sources uses the same methods and algorithms which are used for radio waves, optical, ultra-wideband (UWB) or any other waves. Certain parameters of the signal like Received Signal Strength (RSS), Time of Arrival (TOA), Angle of Arrival (AOA) or Time Difference of Arrival (TDOA) are used to extract information about the source emanating the signal [3]. This chapter discusses methods of acoustic source localization. Section 3.1 introduces the chapter followed by Section 3.2 which explains the concepts of AOA and TDOA. Section 3.3 provides the mathematical modeling of signals to be processed and Section 3.4 discusses various Time Delay Estimation (TDE) techniques followed by the discussion of Compressive Sensing (CS) based localization schemes in Section 3.5.2.

The chapter is concluded in Section 3.6.

3.1 Introduction

In this chapter previous work done by several researchers in the area of acoustic source localization is presented. Different localization techniques will be described with minimal mathematical details to show the concepts. The localization techniques ultimately depend on time delay estimation (TDE) techniques. Almost all the localization techniques require calculating time delays when the sound waves travel from one sensor to another. Thus a detailed discussion on TDE techniques will also be included in this chapter.

3.2 Concept of Direction of Arrival (DOA) and Time Difference of Arrival (TDOA)

Two important localization methods that are widely used in literature are Direction of Arrival (DOA) and Time Difference of Arrival (TDOA). The details of both of the methods follow.

3.2.1 DOA Technique

Consider Fig. 3.1 in which a sensor array has been shown. The array consists of three sensors which are placed in the far-field of a radiating source. Since the array is in the far-field, the waves coming from the source can be considered as

plane waves.

The normal to the wavefront makes an angle θ with the axis along which the sensors are placed. The signals received at all sensors are delayed or advanced versions of the signal received at a reference sensor. In Fig. 3.1, the reference sensor is r_1 . The sensors are d distance apart from each sensor on its sides. Now if the signal is received at sensor 1 at t_0 time, same signal would have already reached at sensor 2 at $(t_0 - d\cos\theta)$ time [8]. Therefore, the time difference (or time delay) between the two sensors is given by

$$\tau_{21} = \frac{d \cos(\theta)}{c} \quad (3.1)$$

where c is the sound velocity in air. If τ_{21} is known and θ ranges from 0° to 180° then θ can be uniquely determined [8]. To find the angle of arrival θ we need to find the time difference τ_{21} . This time difference is also known as time-difference-of-arrival (TDOA) and the process of finding angle θ is known as Direction of Arrival (DOA) estimation (in some references it is also known as Angle of Arrival, AOA) [3]).

To formally develop a mathematical model we proceed as follows based on methods described in [9].

Suppose there are M sensors placed in the far field of a wideband source. Let $\vec{\mathbf{r}}_i$ denotes the location of the i th sensor. $\vec{\mathbf{r}}_i$ is 3-dimensional ($\vec{\mathbf{r}}_i = [x_i, y_i, z_i]$) for a 3D array or 2-dimensional ($\vec{\mathbf{r}}_i = [x_i, y_i]$) for a 2D array. The azimuth and elevation angle of the source are denoted by ϕ and θ respectively. If τ_{j1} represents

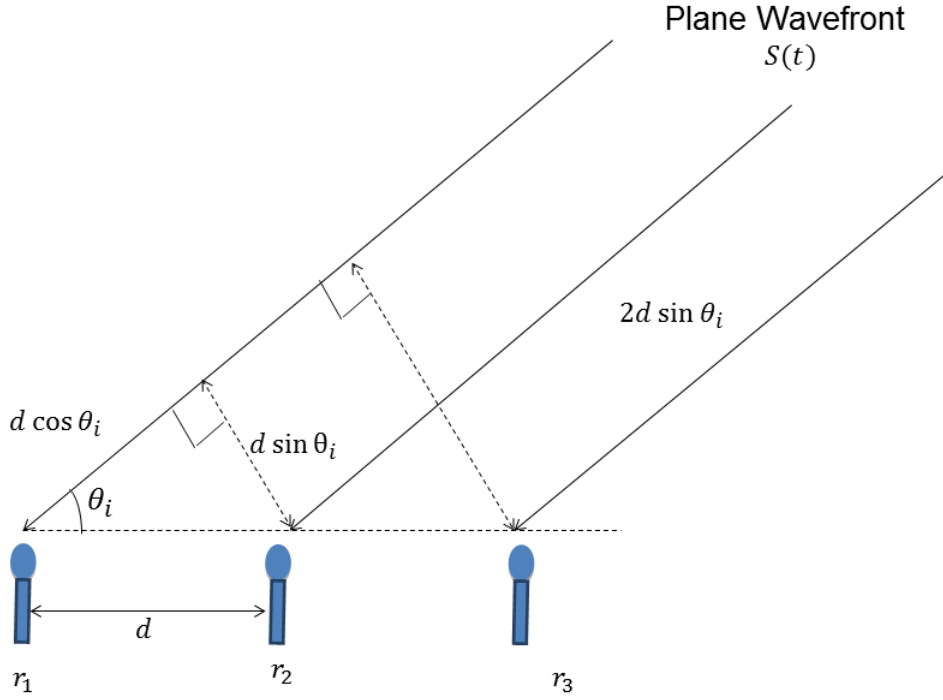


Figure 3.1: Sensor array in the far-field of a radiating source

the TDOA between the reference sensor 1 and any sensor j then we can define a vector as

$$\vec{\tau} = [\tau_{21}, \tau_{31}, \dots, \tau_{M1}]^T; \quad \tau_{j1} = \tau_j - \tau_1 \quad (3.2)$$

and accordingly we can define the DOA vector for the far field signal as

$$\vec{\mathbf{k}} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} = \begin{bmatrix} \sin(\theta) \cos(\phi) \\ \sin(\theta) \sin(\phi) \\ \cos(\theta) \end{bmatrix} \quad (3.3)$$

Now from Fig. 3.1 we see the TDOA between sensor 1 and 2 is given by (3.1) which is actually the projection of distance d between sensor 1 and 2 along the direction of the signal from the source divided by the sound speed. Generalizing

this concept we can write

$$\vec{\tau} = -\frac{\mathbf{R}\vec{\mathbf{k}}}{c}; \quad \mathbf{R} = \begin{bmatrix} \vec{r}_2 - \vec{r}_1 \\ \vdots \\ \vec{r}_M - \vec{r}_1 \end{bmatrix} \quad (3.4)$$

where \mathbf{R} is the matrix whose rows are the distance difference between the reference sensor 1 and all other sensors. Thus if we know the $\vec{\tau}$ and we already know c and \mathbf{R} we can estimate $\vec{\mathbf{k}}$ using Least Squares (LS) as the system is over-determined for $M > 3$ in 2D case and for $M > 4$ in 3D case. The LS solution for $\vec{\mathbf{k}}$ is given by [10]

$$\vec{\mathbf{k}} = \underset{\vec{\mathbf{k}}}{\text{ArgMin}} \left\{ \left(\frac{\mathbf{R}\vec{\mathbf{k}}}{c} + \hat{\vec{\tau}} \right)^T \Lambda_\tau^{-1} \left(\frac{\mathbf{R}\vec{\mathbf{k}}}{c} + \hat{\vec{\tau}} \right) \right\} \quad (3.5)$$

$$= -c (\mathbf{R}^T \Lambda_\tau^{-1} \mathbf{R}) \mathbf{R}^T \Lambda_\tau^{-1} \hat{\vec{\tau}} = -c \mathbf{B} \hat{\vec{\tau}} \quad (3.6)$$

which is a simple multiplication between c , the delay vector and and a data independent matrix \mathbf{B} . After estimating the DOA vector $\vec{\mathbf{k}}$ we can find ϕ and θ by expressing $\vec{\mathbf{k}}$ in Polar coordinates. For 2D case:

$$\hat{\phi} = \cos^{-1}(\hat{k}_x). \quad (3.7)$$

For 3D case [9]:

$$\hat{\phi} = \tan^{-1} \left(\frac{\hat{k}_y}{\hat{k}_x} \right), \quad (3.8)$$

$$\hat{\theta} = \cos^{-1}(\hat{k}_z) = \cos^{-1} \left(\left(1 - (\hat{k}_x^2 + \hat{k}_y^2) \right)^{1/2} \right) \quad (3.9)$$

For the 3D case (3.3) produces three nonlinear equations with two unknowns which is again a LS problem and its solution is given by [9]

$$(\hat{\phi}, \hat{\theta}) = \underset{\hat{\phi}, \hat{\theta}}{\text{ArgMin}} \left\{ \left(\hat{\mathbf{k}} - \vec{\mathbf{k}}(\hat{\phi}, \hat{\theta})^T \right) \Lambda_k^{-1} \left(\hat{\mathbf{k}} - \vec{\mathbf{k}}(\hat{\phi}, \hat{\theta}) \right) \right\} \quad (3.10)$$

where Λ_k^{-1} is the covariance matrix of $\hat{\mathbf{k}}$. Berdugo et al [9] propose another sub-optimal close-form estimate given by

$$\hat{\phi} = \tan^{-1} \left(\frac{\hat{k}_y}{\hat{k}_x} \right), \quad (3.11)$$

$$\hat{\theta} = \tan^{-1} \left(\frac{\left(\hat{k}_x^2 + \hat{k}_y^2 \right)^{1/2}}{\hat{k}_z} \right) \quad (3.12)$$

which they have shown to be asymptotically efficient. Moreover, they claim that following certain geometrical constraints for the sensor arrangements, the closed form in (3.11) and (3.12) achieves the Cramér-Rao Lower Bound (CRLB).

3.2.2 TDOA Technique

Fig. 3.2 displays a scenario where there are three sensors (mics) placed on the vertices of an equilateral triangle. Mic 1 is taken as the reference and is placed at the origin of the Coordinate system. The locations of microphone 1, 2, and 3 are (x_1, y_1) , (x_2, y_2) , (x_3, y_3) respectively. The unknown source location is (x, y) . The TDOA between references i and j (i.e., $t_i - t_j$) may be used to obtain the distance difference, which may be written as

$$d_{ij} = d_i - d_j = c(t_i - t_0) - c(t_j - t_0) \quad (3.13)$$

$$= c(t_i - t_j), \quad i = 1, 2, 3, j = 1, 2, 3, i \neq j \quad (3.14)$$

where t_0 is the time of flight of wave from the source to the reference sensor. $(t_i - t_j)$ corresponds to the TDOA between microphone i and j . This TDOA can be found by using TDE techniques discussed in Section 3.4. From (3.14)

$$d_i = d_{ij} + d_j \quad (3.15)$$

Applying (3.15) to microphones 1 and 2 and squaring both sides,

$$d_2^2 = (d_{21} + d_1)^2 = (x_2 - x)^2 + (y_2 - y)^2 \quad (3.16)$$

$$= x_2^2 - 2x_2x + y_2^2 - 2y_2y + d_1^2 \quad (3.17)$$

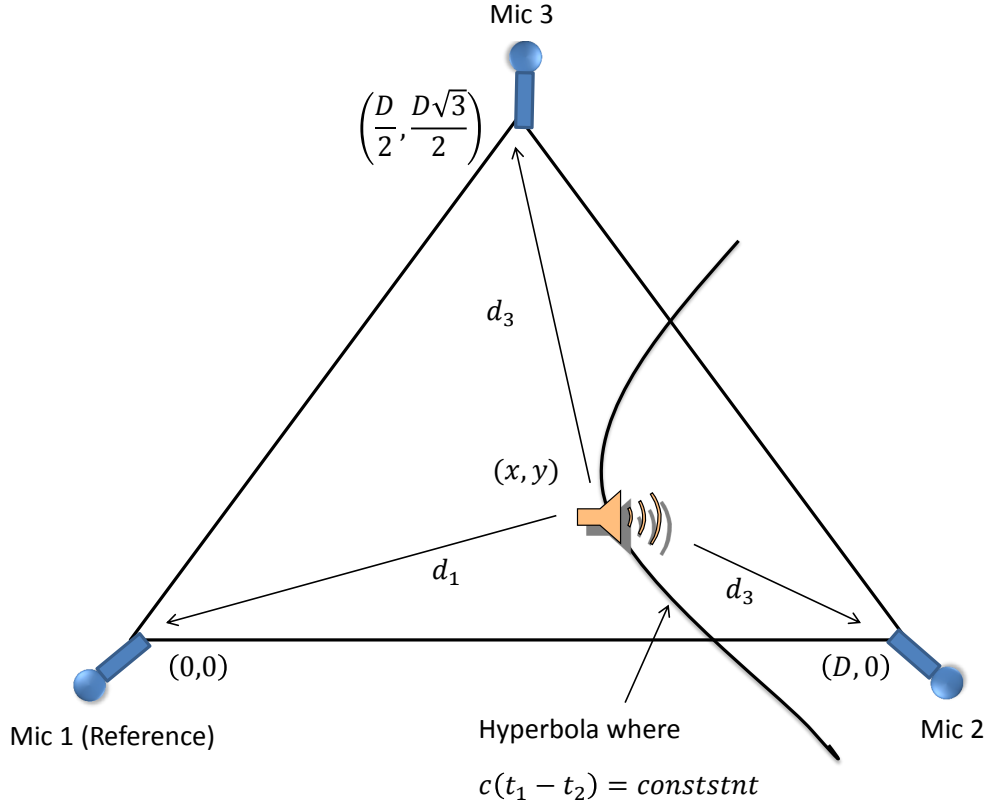


Figure 3.2: Sensor array in the far-field of a radiating source

Rearranging (3.17), we get

$$(d_{21}^2 - x_2^2 - y_2^2) + 2d_{21}d_1 = -2x_2x - 2y_2y \quad (3.18)$$

Similarly for mic 1 and 3, we can derive,

$$(d_{31}^2 - x_3^2 - y_3^2) + 2d_{31}d_1 = -2x_3x - 2y_3y \quad (3.19)$$

Equations (3.18) and (3.19) correspond to hyperbolas which represent the possible source locations corresponding to τ_{21} and τ_{31} respectively. We can solve

these nonlinear equations through a computer program e.g. MATLAB to find the actual source locations. Thus the main task is to estimate the correct TDOA and then we can estimate the source locations easily. Note that both AOA and TDOA methods require the TDOAs among the sensors, the difference is in the way of estimating source location from these TDOAs.

We can generalize this approach for N sensors for the 3D case. Suppose there is an array consisting of $N + 1$ sensors placed in a three dimensional space. Let a_n be the location of the n th sensor (a 3×1 vector, $\vec{a}_n = x\vec{a}_x + y\vec{a}_y + z\vec{a}_z$). For simplicity we assume that one of the sensors is located at origin and we use this sensor as the reference for the remaining sensors. Let \mathbf{x} denote the source location and d_n denote the distance corresponding to the TDOA between sensor n and the reference sensor, then

$$d_n = \|a_n - \mathbf{x}\| - \|\mathbf{x}\|, \quad n = 1, \dots, N \quad (3.20)$$

where $\|\cdot\|$ denotes the Euclidean norm. Equation (3.20) can be written as [11]

$$\|a_n - \mathbf{x}\|^2 = \|d_n + \mathbf{x}\|^2, \quad (3.21)$$

which upon simplifying produces the following system of equations in the unknown \mathbf{x} ,

$$d_n \|\mathbf{x}\| + a_n^T \mathbf{x} = b_n, n = 1, \dots, N \quad (3.22)$$

where

$$b_n = \frac{\|a_n\|^2 - d_n^2}{2} \quad (3.23)$$

If we define the following as

$$y = \begin{bmatrix} \|\mathbf{x}\| \\ \mathbf{x} \end{bmatrix}; \quad \phi = \begin{bmatrix} d_1 & a_1^T \\ \vdots & \vdots \\ d_N & a_N^T \end{bmatrix}; \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} \quad (3.24)$$

then (3.22) can be written as

$$\Phi y = b \quad (3.25)$$

We will be using more than four sensors to improve the TDOA uncertainty, in which case, (3.25) becomes over-determined and can be solved using the LS method. The LS solution of (3.25) is given by

$$\hat{y} = (\Phi^T \Phi)^{-1} \Phi^T b \quad (3.26)$$

Then the corresponding LS estimate of source location is given by

$$\hat{x} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \hat{y} \quad (3.27)$$

Note that the last three columns of Φ consist of the coordinates of the sensors locations. If all the sensors are in a plane, then Φ will become singular (one column will be zero) or almost singular and the solution to (3.26) will become

impossible. To avoid such a situation we arrange the sensors in such a way that the resulting array covers the three dimensions. We discuss the geometries in detail in Chapter 5.

3.3 Signal Models

Depending on the environment we may have different signal models. The possible signal models, in our case, are given below.

3.3.1 Single Source Free Field Model

Suppose there is only one source radiating sound waves in an anechoic environment (open or outdoor environment). An array of N microphones is placed in that environment. If we choose microphone 1 as the reference, the signal received at the n th microphone can be written as [8][12]

$$y_n(k) = \alpha_n s(k - t - \tau_{n1}) + v_n(k) \quad (3.28)$$

$$= \alpha_n s(k - t - \mathcal{F}_n(\tau)) + v_n(k) \quad (3.29)$$

$$= x_n(k) + v_n(k), n = 1, 2, \dots, N \quad (3.30)$$

where $\alpha_n (n = 1, 2, \dots, N)$ are the attenuation factors, $s(k)$ is the unknown source signal, t is the propagation time from source to reference mic 1, $v_n(k)$ is an additive noise signal at the n th sensor (assumed to be uncorrelated with the source signal and other mics' noise), and $\tau_{n1} = \mathcal{F}_n(\tau)$ is the TDOA between sensor 1 and

n and τ is the TDOA between sensors 1 and 2. For $n = 3, \dots, N$ the function \mathcal{F}_n depends on τ as well as on the geometry of the array.

3.3.2 Single Source Reverberant Model

Real environments are not free from objects, there are always objects causing reflections and reverberations (indoor case). Consider there is a single source in a reverberant model. This scenario can be modeled as single-input multiple-output (SIMO) system. The n th sensor signal at time k can be given by [8][12]

$$y_n(k) = g_n * s(k) + v_n(k), \quad (3.31)$$

$$= x_n(k) + v_n(k), n = 1, 2, \dots, N \quad (3.32)$$

where g_n is the channel impulse response from the source to mic n . Equation (3.31) can also be written in a matrix form as

$$\mathbf{y}_n = \mathbf{G}_n s(k) + \mathbf{v}_n(k), n = 1, 2, \dots, N \quad (3.33)$$

where

$$\mathbf{y}_n(k) = \begin{bmatrix} y_n(k) & y_n(k-1) & \dots & y_n(k-L+1) \end{bmatrix}^T, \quad (3.34)$$

$$\mathbf{G} = \begin{bmatrix} g_{n,0} & \dots & g_{n,L-1} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & g_{n,0} & \dots & g_{n,L-1} \end{bmatrix}, \quad (3.35)$$

$$s(k) = \begin{bmatrix} s(k) & s(k-1) & \dots & s(k-L+1) & \dots & s(k-2L+2) \end{bmatrix}^T, \quad (3.36)$$

$$\mathbf{v}_n(k) = \begin{bmatrix} v_n(k) & \dots & v_n(k-L+1) \end{bmatrix}^T \quad (3.37)$$

and L is the length of the longest channel impulse response of the SIMO system.

The TDOA τ is an implicit or hidden parameter in this model.

3.4 Time Delay Estimation (TDE) Techniques

As we discussed earlier, the main task that every position localization technique needs to perform is to estimate the TDOA among the sensors. There are various algorithms which are used for this purpose. Every algorithm and technique has its own benefits, drawbacks, scenarios, and applications. We will discuss the most widely used TDE techniques in the following subsections.

3.4.1 Generalized Cross-Correlation Techniques

The generalized cross-correlation (GCC) method [8][12][13] is the most widely used method for estimating time delay. GCC assumes the free field model and considers only two microphones. TDOA is found by choosing a delay that maximizes the cross-correlation function of the two mics' signals.

$$\hat{\tau}^{GCC} = \arg \max_{\tau} r_{y_1 y_2}^{GCC}(p), \quad (3.38)$$

where

$$r_{y_1 y_2}^{GCC}(p) = F^{-1} [\psi_{y_1 y_2}(f)] \quad (3.39)$$

$$= \int_{-\infty}^{\infty} \psi_{y_1 y_2}(f) e^{j2\pi f p} df = \int_{-\infty}^{\infty} \vartheta(f) \phi_{y_1 y_2}(f) e^{j2\pi f p} df \quad (3.40)$$

is the GCC function, and $F^{-1}[\cdot]$ is the inverse discrete-time Fourier transform (IDTFT),

$$\phi_{y_1 y_2}(f) = E [Y_1(f) Y_2^*(f)] \quad (3.41)$$

is the cross-spectrum with $Y_n(f) = \sum_k y_n(k) e^{-j2\pi f k}$, $n = 1, 2$, $\vartheta(f)$ is a frequency-domain weighting function, and

$$\psi_{y_1 y_2}(f) = \vartheta(f) \phi_{y_1 y_2}(f) \quad (3.42)$$

is the generalized cross-spectrum and $E[\cdot]$ represents the expectation operation.

Classical Cross-Correlation

By setting $\vartheta(f) = 1$, the GCC expression simplifies to simple cross-correlation function [8][12][13]. The free-field signal model was given in (3.28). Taking its Discrete Fourier Transform (DFT), we get

$$Y_n(f) = \alpha_n S(f) e^{-j2\pi f [t - \mathcal{F}_n(t)]} + V_n(f), n = 1, 2. \quad (3.43)$$

Substituting (3.43) into (3.42) and assuming that all noise signals are uncorrelated with each other and with source signals; we get expression for the cross-correlation function

$$\psi_{y_1 y_2}^{CC}(f) = \alpha_1 \alpha_2 e^{-j2\pi f \tau} E [|S(f)|^2] \quad (3.44)$$

which is dependent on the source signal.

Smoothed Coherent Transform

Often the microphone signal is smoothed out to reduce fluctuation effects on TDOA by using

$$\vartheta(f) = \frac{1}{\sqrt{E [|Y_1(f)|^2] E [|Y_2(f)|^2]}}, \quad (3.45)$$

which gives the so-called Smoothed COherence Transform (SCOT) [8][12][13].

Substituting (3.45) and (3.43) into (3.42), we get the expression for SCOT cross-spectrum

$$\psi_{y_1 y_2}^{SCOT}(f) = \frac{\alpha_1 \alpha_2 e^{-j2\pi f \tau} E [|S(f)|^2]}{\sqrt{E [|Y_1(f)|^2] E [|Y_2(f)|^2]}} \quad (3.46)$$

$$= \frac{e^{-j2\pi f \tau}}{\sqrt{\left(1 + \frac{1}{SNR_1(f)}\right) \cdot \left(1 + \frac{1}{SNR_2(f)}\right)}}, \quad (3.47)$$

where

$$SNR_n(f) = \frac{\alpha_n^2 E [|S(f)|^2]}{E [|V_n(f)|^2]}, n = 1, 2 \quad (3.48)$$

In case the SNRs are the same at both microphones, then

$$\psi_{y_1 y_2}^{SCOT}(f) = \left[\frac{SNR(f)}{1 + SNR(f)} \right] \cdot e^{-j2\pi f \tau} \quad (3.49)$$

Equation (3.49) means that the SCOT performance depends on the SNR and so in return, the TDOA estimate will vary with SNR. With $SNR \gg 1$,

$$\psi_{y_1 y_2}^{SCOT}(f) \approx e^{-j2\pi f \tau} \quad (3.50)$$

Thus the SCOT produces good results when noise levels are quite low.

The Phase Transform

If we set

$$\vartheta(f) = \frac{1}{|\phi_{y_1 y_2}(f)|} \quad (3.51)$$

in (3.40) we get the phase transform (PHAT) method which takes only phase of (3.40) in to consideration. The generalized cross-spectrum becomes

$$\psi_{y_1 y_2}^{PHAT}(f) = e^{-j2\pi f \tau}, \quad (3.52)$$

Substituting (3.52) in (3.40), we get the following GCC function:

$$r_{y_1 y_2}^{PHAT}(p) = \int_{-\infty}^{\infty} e^{j2\pi f(p-\tau)} df = \begin{cases} \infty, & p = \tau \\ 0, & otherwise \end{cases} \quad (3.53)$$

3.4.2 Eigenvector-Based Techniques

These techniques take advantage of the usage of multiple microphones (more than 2) [8][12]. We assume the single-source free-field model in (3.28) with N microphones. Further we assume that the array is in the far-field, all attenuation factors $\alpha_n = 1$ and noise signals are mutually independent Gaussian random processes with the same variance. We will discuss two methods within this category.

Narrowband MUSIC

Transforming (3.28) into frequency domain, we get [8][12]

$$Y_n(f) = X_n(f) + V_n(f) = S(f)e^{-j2\pi[t+\mathcal{F}_n(\tau)]f} + V_n(f) \quad (3.54)$$

where $Y_n(f)$, $X_n(f)$, $V_n(f)$, and $S(f)$ are the DTFT of $y_n(k)$, $x_n(k)$, $v_n(k)$, and $s(k)$ respectively. We define a frequency-domain vector as:

$$\mathbf{y} = \begin{bmatrix} Y_1(f) & Y_2(f) & \dots & Y_n(f) \end{bmatrix}^T \quad (3.55)$$

Substituting (3.54) into (3.55), we get

$$\mathbf{y} = \mathbf{x} + \mathbf{v} \quad (3.56)$$

$$= \varsigma(\tau)S(f)e^{-j2\pi ft} + \mathbf{v} \quad (3.57)$$

where

$$\zeta(\tau) = \begin{bmatrix} e^{-j2\pi\mathcal{F}_1(\tau)f} & e^{-j2\pi\mathcal{F}_2(\tau)f} & \dots & e^{-j2\pi\mathcal{F}_N(\tau)f} \end{bmatrix}^T, \quad (3.58)$$

and \mathbf{v} is defined similarl to \mathbf{y} . The output covariance matrix is given by

$$R_y = E(\mathbf{y}\mathbf{y}^H) = R_X + \sigma_v^2 I, \quad (3.59)$$

where

$$R_X = \sigma_S^2 \zeta(\tau) \zeta^H(\tau), \quad (3.60)$$

And $\sigma_S^2 = E[|S(f)|^2]$ and $\sigma_v^2 = E[|V_1(f)|^2] = \dots = E[|V_N(f)|^2]$ are the signal and noise variances respectively. After performing eigenvalue decomposition of R_y , we obtain

$$\mathbf{R}_Y = \mathbf{B}\mathbf{\Lambda}\mathbf{B}^H, \quad (3.61)$$

where

$$\mathbf{\Lambda} = \text{diag} \begin{bmatrix} \lambda_{Y,1} & \lambda_{Y,2} & \dots & \lambda_{Y,N} \end{bmatrix} \quad (3.62)$$

$$= \text{diag} \begin{bmatrix} \lambda_{Y,1} + \sigma_v^2 & \sigma_v^2 & \dots & \sigma_v^2 \end{bmatrix} \quad (3.63)$$

is a diagonal matrix consisting of the eigenvalues of \mathbf{R}_Y ,

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & \dots & b_N \end{bmatrix}, \quad (3.64)$$

\mathbf{b}_n is the eigenvector associated with the eigenvalue $\lambda_{Y,n}$, and $\lambda_{Y,1}$ is the only non-zero positive eigenvalue of R_Y . For $n \geq 2$

$$\mathbf{R}_Y \mathbf{b}_n = [\sigma_s^2 \varsigma(\tau) \varsigma^H(\tau) + \sigma_v^2 \mathbf{I}] \mathbf{b}_n \quad (3.65)$$

From (3.64) and (3.65), it is also found that

$$\sigma_s^2 \varsigma(\tau) \varsigma^H(\tau) \mathbf{b}_n = 0 \quad (3.66)$$

which is equivalent to

$$\varsigma^H(\tau) \mathbf{b}_n = 0 \quad (3.67)$$

or

$$\mathbf{b}_n^H \varsigma(\tau) = 0 \quad (3.68)$$

It means that the eigenvectors associated with the $N - 1$ lowest eigenvalues of \mathbf{R}_Y are orthogonal to the vector corresponding to the actual TDOA. The following cost function can be used to find TDOA τ . The function $p = \tau$ maximizes this cost function:

$$J_{MUSIC}(p) = \frac{1}{\sum_{n=2}^N |b_n^H \varsigma(p)|^2} \quad (3.69)$$

where MUSIC stands for MULTiple SIGNAL Classification.

Broadband MUSIC

Since the speech signal is non-stationary thus the narrowband MUSIC algorithm does not produce good results. One straight forward solution to this problem is to change cost function of (3.69) such that it covers all the sound frequency range. But this will degrade the delay estimation performance because the peak will not be well-defined due to a broadband signal. Another approach uses spatial correlation concepts and this is discussed in detail in [8].

3.5 Localization Schemes

3.5.1 Conventional Localization Schemes

In [9], a new algorithm has been developed for direction finding of the incoming acoustic wave. The algorithm finds the azimuth and elevation angles directly from the estimated time delays between the array elements. The algorithm offers computational simplicity as it utilizes the linear relationship between the time delay vector and the DOA vector in Cartesian coordinates. Numerical and experimental results were given to demonstrate the performance of the Time Delay Direction Finding (TDDF) algorithm. The experimental results with a 7 microphone array have shown that in an anechoic chamber the average TDDF azimuth error was about 1.5 degrees, while in a regular room the average error was about 5 degrees.

In [14], a distributed acoustic passive localization method using Wireless Sensor Network (WSN) has been proposed. Based on the time difference of arrival

(TDOA) from each participating WSN nodes in clusters, the base station calculates the azimuth angle and pitch angle, then using the geometrical information along with these TDOA estimates, the source location is found.

In [15], the authors localized the acoustic source using a WSN by utilizing the measured signal quantities like RSS, AOA and TDOA. For each of these quantities, an appropriate weighted LS criterion function was developed that was used for sound source localization. The authors claim that their work provides improvement of the localization accuracy for low SNR.

In [16], the authors propose a WSN based acoustics source localization and tracking system. The sensor board used in WSN was accompanied by a Xilinx Spartan-3L FPGA for powerful signal processing. The nodes were sparsely deployed. Each node, due to a powerful local signal processor, was able to estimate DOA locally and send it to a base station. Due to the widely distributed sensing and the novel sensor fusion technique, the method can handle multiple measurement errors prevalent in reverberant environments. Furthermore, the paper also describes the DOA estimation algorithm and the applied middleware services for coordinated sensing and communication introduces the sensor fusion algorithm and presents a detailed error analysis.

3.5.2 Localization Schemes Utilizing CS and Orthogonal Clustering algorithms

Although CS is finding applications in a wide range of areas in signal processing, it has not been widely used for acoustic source localization. The limited literature that can be found in regard to source localization using CS is mostly based on simulations. Especially, most of the researchers focused on electromagnetic (EM) source localization [17].

In [18], the author uses spatial CS for direction of arrival estimation. Utilizing the spatial scarcity of the sensor array and spatial orientation diversity, the author claims that his approach addresses challenging array signal processing problems such as left-right ambiguity and poor estimation performance at end-fire. However, the results are totally based on simulation and no practical implementation has been considered.

Exploiting spatial and signal scarcity, the authors in [19] develop a Bayesian framework for the localization problem. The authors also discuss 1-bit CS to reduce the amount of inter-sensor communications by transmitting only the intrinsic timing information. They also develop an algorithm for bearing estimation using a network of sensors. However, like most of the work in literature the CS approach towards localization problem has been considered only in simulation. It is, therefore, necessary to investigate the performance of CS theory in practical hardware based systems.

The conventional localization schemes estimate the TDOA directly from the

received signal at the sensors by cross-correlation or other TDE techniques. We can also use the channel impulse response concept to estimate the TDOA between the signals received at different sensors. But in this case the TDOA is hidden and estimation techniques are used to estimate the channel impulse response first.

The Orthogonal Clustering (OC) algorithm [5] basically estimates the Room Impulse Response (RIR) from the received signals at sensors. The TDOA can then be found by finding the time difference between the Direct Line of Sight (DLOS) component of the RIR. The details of how the algorithm works are given in chapter 5 and this method is used in the hardware implementation of the Impulsive Acoustic Source Localization (IASL) system proposed in this work. The main advantages of OC algorithm are:

1. It utilizes the a priory statistical information of the signal: sparsity (there are limited number of reflections of the signal), structure of the matrices of the mathematical models etc.
2. The algorithm does not need sampling of sensors at Nyquist rate, it can produce good results at sub-Nyquist sampling rate [5][6]
3. Subsampling reduces the power consumption of the sensors, especially in the case of the wireless mics which run on batteries.
4. Reduced computational complexity.

More details about the algorithm will be given in chapter 5.

3.6 Conclusion

Acoustic source localization is basically a three step process; i) sound data acquisition from sensors array ii) estimating time delay from captured signals and iii) finding source location from time delays using localization algorithms. This chapter introduced all of these steps. The concepts of DOA and TDOA were explained at the beginning of the chapter. Signal models for different scenarios were developed to properly implement source localization in different environments. Several Time Delay Estimation (TDE) techniques were also discussed and relevant mathematical expressions were derived. At the end of the chapter an overview of acoustic source localization literature was presented by describing the results, techniques and relevant applications found in literature.

CHAPTER 4

WIRELESS SENSOR NETWORK BASED SOLUTION AND ISSUES

4.1 Introduction

Wireless Sensor Networks (WSN) research has focused mainly on implementing applications which use sensors that read slowly varying physical phenomena such as temperature, light, pressures etc. Moreover, the devices in the network (wireless sensor nodes) themselves are low power. Accordingly the software environments (operating system for operating these devices, software drivers etc.) are also designed keeping in mind the low power capabilities of the nodes.

TinyOS is a lightweight operating system specifically designed for low-power wireless sensors [20]. TinyOS is an open source operating system and it has gone

through evaluation of several versions. Its latest version is TinyOS 2.1.1 [21].

One of the objectives of the thesis is to investigate acoustic source localization in open environments and WSN is a natural choice for this purpose because of its several properties like good range (about 300meters), being active wireless devices (the node can perform operation on samples), low power consumption compared to other technologies, fine networking among nodes and many more. However, developing a WSN for acoustic application is not an easy task and we faced several issues during the development which will be discussed shortly. In literature, there is almost no material available that discusses issues in the TinyOS for commercial off the shelf (COTS) wireless sensor nodes (WSN) platforms in regard with high processing applications [22].

In this chapter, we present an analysis of the performance of TinyOS 2.x for an application that involve high sampling of the sensors, saving of the samples in local flash memory of the nodes and after completion of the sampling process reading back these samples from flash and forwarding it to the sink node. This application involves the implementation of several components and their relevant interfaces.

We use Crossbow's IRIS [23] nodes, MIB520 programmer and sink [23] and MTS300 sensorboards [24] to implement our application. The IRIS nodes have 4Mbit flash memory [25].

4.2 Related Work

The acoustic source localization research is not new and a good literature is available in this area. Successful attempts have also been made to implement acoustic source localization using WSN. Acoustic source localization is high processing application and require higher sampling rate (at least 8 kHz) than what is provided by the available hardware and/or software environment for WSN. Therefore, the researchers use other ways such as attaching additional hardware, editing the existing software modules or writing new software modules etc. to cope with the issues that may be faced during acoustic application development.

In [26], the authors present a custom designed sensorboard that can be used with zigbee enabled nodes for multi-channel data processing. The sensorboard was designed using FPGA and it can be used with Telos-B and MICAz/MICA2 motes [27]. A microphone can be attached to the sensorboard for acoustic applications. This sensorboard relieves the motes from the burden of high processing that is needed for acoustic applications.

Gyula Simon, Miklós Maróti et al. present in [28] a WSN based counter sniper system in which they localize a sniper. They also use their custom designed sensorboards based on FPGA to carry out the processing of acoustic signals as the sensor nodes they use (UC Berkeley's Mica2 nodes [27]) are not capable of carrying out these high processing tasks using the standard TinyOS library components.

In [29], a software based approach was chosen to cope with the limitation of WSN for high processing applications. The authors wrote their own components

rather than using standard TinyOS components to implement high sampling of the microphones of Mica2. By utilizing the ADC's free running mode of ATmega128 MCU and intelligently using the flash memory of the mote the authors were able to achieve a sampling rate of 17.723 kHz. This shows that the hardware itself is capable of sampling the sensors at high rate but the software environment i.e. TinyOS is not fully utilizing it. We will explain it more in Section 4.7.

4.3 High Level Structure of the Application

Now we will present the high level structure of our application. Fig. 4.1 depicts a scenario of our application. There are three IRIS motes with MTS300 sensor-boards (MTS300 not shown in the Fig. 4.1) and a sink node which is also an IRIS mote connected to MIB520 gateway. The gateway is then connected to the PC through USB connection.

Our objective is to sample the microphones attached to every node at the same time and transfer that samples from nodes to PC. So this can be achieved in the following way.

1. All the nodes in the network should be strictly synchronized.
2. Then we send a command from sink node to all the nodes to start sampling for a certain number of samples
3. The nodes upon receiving this command start sampling and store the samples in their local flash memory.

4. Upon completion of sampling the nodes read back stored samples from flash and send them to the sink.
5. The sink forwards these received packets to the PC.

Every node in the network runs the same application except the sink node which has its own dedicated application developed for it. The details of application development and underlying structure are given in Section 4.5. But before that we would like to give a brief introduction to TinyOS 2.x.

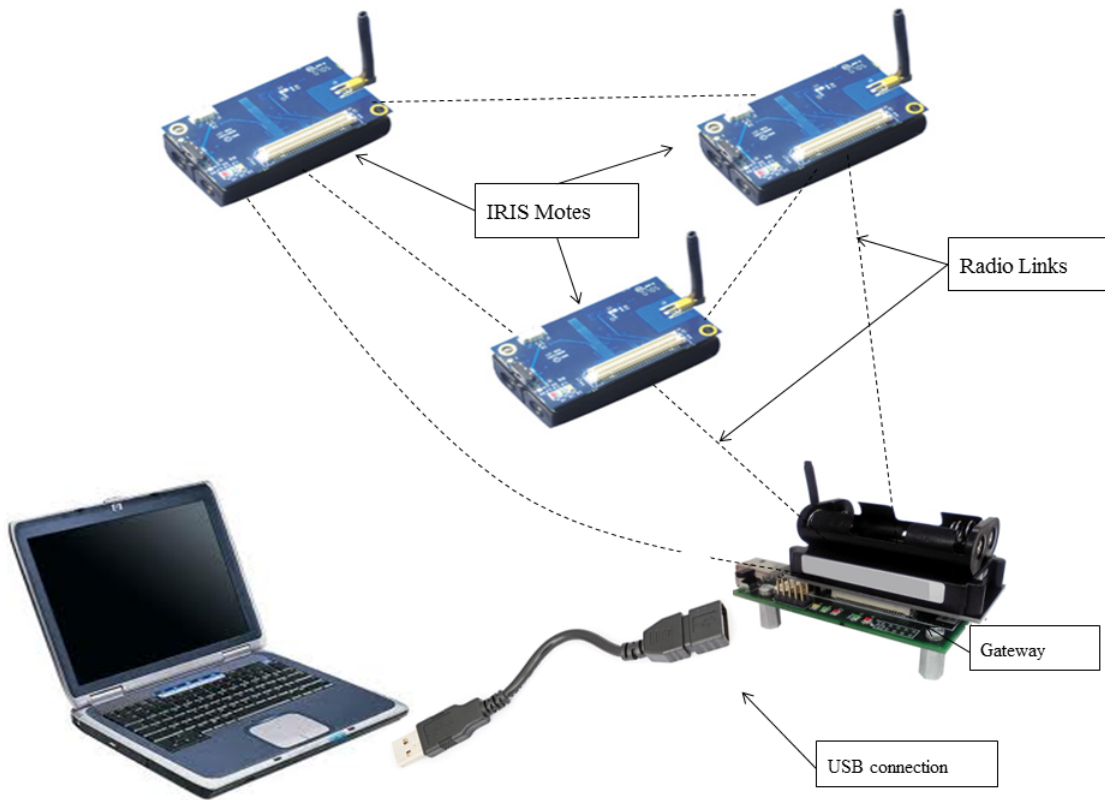


Figure 4.1: An example Wireless Sensor Network

4.4 TinyOS 2.x for WSN

“TinyOS is a lightweight operating system specifically designed for low-power wireless sensors” [20]. TinyOS is an open source operating system; a large community from all over the world contributes to its development. The website www.tinyos.net is the official website of TinyOS and provides almost all the necessary information for TinyOS like its source files, instructions for downloading TinyOS environment, tutorials, example application and much more.

TinyOS is written in nesC language (a dialect of C programming language) and features a component-based architecture, which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS uses the concepts of components and interfaces. Components are different files which perform specific tasks while interfaces are communication link among different components. Interfaces carry commands and events from one component to another.

A component has two types: i) Module and ii) Configuration. Modules implement programming logic while configurations connect components into larger abstractions. A component uses three computational concepts: i) commands, ii) events, and iii) tasks. Commands and events provide the mechanism of communication among components while tasks are used to express intra-component concurrency. A command from a component is a request to another component to use its services. An event is a signal from the provider of a service to the user about the completion of the request made by the user previously. For exam-

ple, a component may request the sampling of a sensor to corresponding provider of the service. Once the request (sampling of the sensor) has been completed the provider signals an event to the user along with the sampled data and other information.

For illustration purpose an example structure of an application called BlinkC is shown in Fig. 4.2. Fig. 4.2 is identical to a figure generated by nesdoc -a tool of TinyOS which generates documentation for an application. In nesdoc diagrams, a single box is a module and a double box is a configuration. Dashed border lines denote that a component is a generic while solid border lines indicate that a component is singleton [20]. A generic component can be instantiated multiple times in an application and every instantiation is an independent copy of the component being instantiated. Singleton components are single components and are only one instance. If multiple configuration wire to a singleton component they all will use the same single copy of that component. In Fig. 4.2, TimerMilliC is a generic component and has been instantiated three times in BlinkC application while LedsC is a singleton component.

4.5 Acoustic Signal Acquisition Implementation

As was discussed in Section 4.3, our application consists of several steps i.e. i) sampling of the microphone ii) writing to/reading from flash memory iii) transferring data from motes to the sink wirelessly and iv) forwarding data from the sink to the PC through serial communication. We will now discuss the development

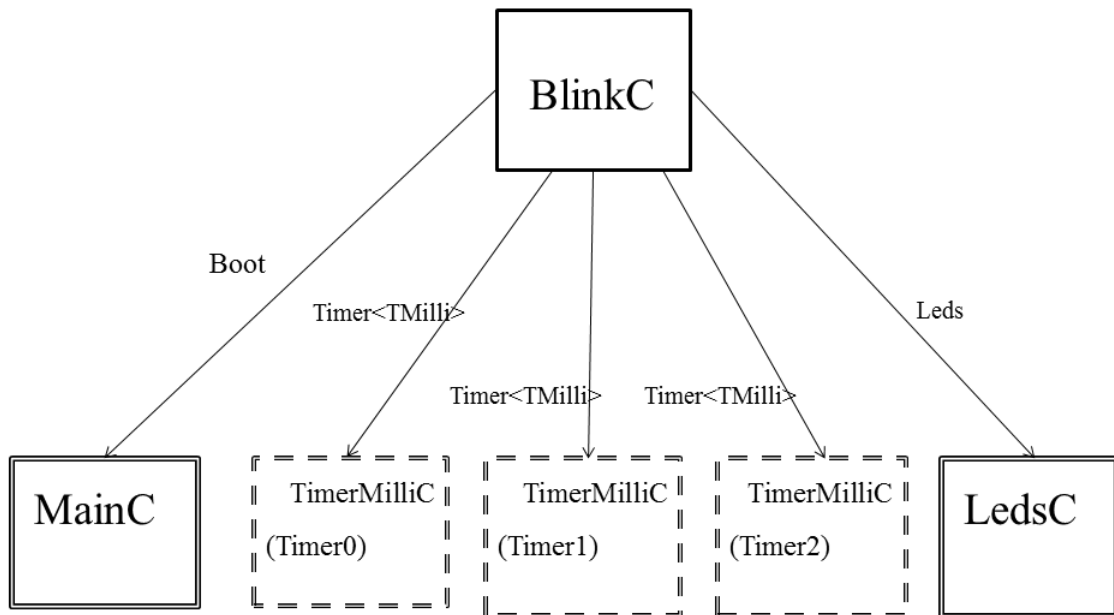


Figure 4.2: Block Diagram representation of a TinyOS application

of these steps one by one.

4.5.1 Synchronization

Our application requires the nodes to be strictly synchronized network-wide and the synchronization error should be in the order of micro seconds. There are several synchronization algorithms available for WSN but we have consulted the most popular ones. These are:

1. Reference Broadcast Synchronization (RBS) [30][31]
2. Timing-sync Protocol for Sensor Networks (TPSN) [32]
3. Flooding Time Synchronization Protocol (FTSP) [33]

In the RBS, a reference message is broadcasted. The receivers record their local time when receiving the reference broadcast and exchange the recorded times with each other. The main advantage of RBS is that it eliminates transmitter-side non-determinism. The disadvantage of the approach is that additional message exchange is necessary to communicate the local time-stamps between the nodes. The TPSN algorithm first creates a spanning tree of the network and then performs pairwise synchronization along the edges. Each node gets synchronized by exchanging two synchronization messages with its reference node one level higher in the hierarchy. The TPSN achieves two times better performance than RBS. The problem with TPSN is that it does not estimate the clock drift of nodes, which limits its accuracy.

We chose FTSP for network synchronization because of the following reasons:

1. It eliminates the shortcoming of the RBS and TPSN.
2. It achieves better performance than both of them with less synchronization error
3. Its implementation for TinyOS 2.x is already available [34].

The FTSP algorithm works by flooding several broadcast messages to the network from a beacon or root node elected by the nodes in the network. The nodes in the network receive these messages from root node, calculate the difference in the local and global (root's) time and then convert its local time to global time using the information of local clock drift and skew. FTSP uses MAC-layer time-stamping and error compensation techniques to achieve synchronization. It

is mentioned in [33] that FTSP achieves good synchronization and a maximum error of 4 microseconds can occur. However, as will be explained later in Section 4.6, this little error is achieved in an ideal scenario when there is no other processing going on the notes.

4.5.2 Sampling

The IRIS notes do not have any sensors embedded on it. Separate sensorboards are attached to it. We are using MTS300 sensorboard. This board has three sensors: microphone, temperature and a light sensor and a sounder device used to create a 4 kHz sound wave.

MTS300 carries a Panasonic WM-62A omnidirectional microphone [35]. The microphone circuitry consists of a tone detector, a multiplexer, a preamplifier, an anti-aliasing filter, a digital potentiometer and a bi-quad active filter. There are two output choices out of the circuitry: a raw microphone signal or the tone detector output. The LM567 CMOS Tone Detector IC detects a 4 kHz signal and outputs 1. If 4 kHz signal was not detected a 0 is outputted. 4 kHz tone can be generated by the sounder present on the sensorboard. We are using the raw microphone signal output for our application.

TinyOS 2.x provides a general purpose interface `ReadStream` for reading/sampling a sensor. The lower layers then provide its implementation for different sensors. For microphone `ReadStream` is provided by `MicReadStreamP` and `MicStreamC` components.

The microphone first needs to be powered on before its use. So after the completion of system initialization, we power on the microphone circuitry by using SplitControl interface. SplitControl interface for microphone is implemented by MicP component.

Unfortunately ReadStream is implemented using the single-sample mode of the ADC in combination with a microsecond alarm. It means that ADC free running mode is off and we can't achieve higher rates with this interface. During our experimentation we've found that we can attain approximately 4.9 kHz sampling rate with other processing turned off. Of course, it is not enough sampling rate for an audio signal. As the anti-aliasing filter of the microphone circuitry is a band-pass filter with cutoff frequencies 159Hz and 6.4 kHz we need to sample the mic at a rate greater than or equal to 12.8 kHz. One way of achieving higher sampling rate is to sample the microphone in ADC free-running mode [36] by writing our own hardware specific code. Since our objective was to check the performance of TinyOS 2.x standard interfaces and components for high sampling applications like ours we did not continue with our own code development and continued with the standard interfaces and components.

We cannot send data with higher sampling rate directly through radio because of the limited bandwidth of the radio channels and the overhead in communication protocols. So we need to store the sampled data into the Flash memory of the mote as it is sampled. After completing the sampling of mic we then read the data back from the mote and send it to the BaseStation through radio.

4.5.3 Flash Reading/Writing

IRIS motes have a 4-Mbit serial flash (Atmel's AT45DB041) [25] for storing data, measurements, and other user-defined information.

TinyOS 2.x implements the storage tasks through one of three abstractions: i) small objects, ii) circular/linear logs and iii) large objects [37][38]. TinyOS 2.x also provides interfaces to abstract the underlying storage services and components that provide these interfaces.

Since our application deals with large amount of data, we use the large object abstraction. The interfaces BlockRead and BlockWrite are used for this abstraction to read and write data to the flash. These interfaces are provided by the BlockStrogeC component. We use these interfaces for reading and writing to flash to check its performance.

4.5.4 Radio communication

Radio communication uses different interfaces to carry out multiple tasks. In our code we used these interfaces: i) AMSend, for sending radio messages to a single node or to the network, ii) Receive, for receiving packets from the network, iii) Packet, for accessing payload of the radio packet and iv) AMPacket, for setting packet destination address in case of the unicast communication.

The radio packet consists of a header, footer and a payload area. The header and footer are system defined and usually need not to be edited. The payload area is where the user can put the data. The data can directly be put into the payload

or for consistency and convenience structures can be used to organize the data inside the payload. We use our own structure given in Fig. 4.3 to transfer our information among the motes. There are two structures: i) `moteStruct`: which is used by the nodes in the network to transmit their packets to the sink and ii) `baseStruct`: which is used by the sink to transmit its packets to the network.

The packet length of radio message can be defined by using specific lines in the Makefile of the application. In our case the total packet length including the overhead was 65 bytes due to the fact that we added extra fields for better management of the network as will be explained in Section 4.5.6. The default packet size of TinyOS 2.x is 28 bytes

```
typedef nx_struct moteStruct {
    nx_uint16_t src_addr;
    nx_uint8_t  is_synced;
    nx_uint8_t  state;
    nx_uint32_t time_stamp;
    nx_uint32_t start_time;
    nx_uint32_t sampling_time;
    nx_uint32_t stop_time;
    nx_uint16_t reading[BUF_READ];
} moteStruct;
typedef nx_struct baseStruct {
    nx_uint16_t nodeid;
    nx_uint16_t cmd;
}baseStruct;
```

Figure 4.3: User-defined structure for the payload of radio packet

4.5.5 Serial communication and MATLAB interfacing

The next stage is to deliver the packets from the sink that it received from network to the PC over the USB link. MIB520 uses FTDI FT2232C to use USB port as virtual COM port [23]. After installing the required drivers we can access the MIB520 as serial port in the PC.

We then access the virtual COM port in MATLAB to get the binary data from the port. The data need to be parsed into packets and their corresponding fields. We wrote MATLAB programs to convert binary data into packets and extracted specific fields from these packets. As is shown in Fig. 4-3 every field of the packet contain specific data and a careful coding is required to successfully extract the required information out of the received binary data from serial port.

4.5.6 Overall Structure of the Application

After discussing individual components of our model we are now able to discuss the application itself. The network consists of a single sink node and multiple data acquisition nodes. The sink node requires its own special code to properly manage the network and transmit/receive packets to/from the network and to/from the PC. We call this application as BaseStation and from now on we will refer to the sink node, alternatively, by BaseStation. The BaseStation is our modified version of the standard BaseStation application that can be found in /tos/apps directory of the TinyOS 2.x standard distribution. The original BaseStation only works as a repeater, whatever packet it receives from the network it forwards it to the

PC and vice versa. Our modified version is an intelligent BaseStation, it does not forward every packet to PC until it was intended and also it manages the network's status by issuing certain commands to the nodes.

All the nodes besides the sink node run an application that is different from the BaseStation and this application will be called the sync. All the nodes (except the sink) have the same sync application installed onto them. We will refer to these nodes as sync, alternatively, onward.

We have designed the sync and BaseStation applications such that both of them interact with each other in a command-response manner. We define a set of commands which are known to the BaseStation and all the nodes in the network. The BaseStation sends the commands in the cmd field of its structure (baseStruct) while the nodes respond to these commands by sending their status in the state field of their structure (moteStruct). We define these commands and responses as enum constants as given in Fig. 4.4.

First we will discuss the BaseStation side story. Upon booting the BaseStation broadcasts a message with the REBOOT state in the cmd field to the network to indicate that it has booted and all the nodes need to reboot as well. After this it does nothing and waits to receive any radio packet forwarded to it. The nodes reboot and once they have booted successfully they send the state BOOTED in response to indicate that they have booted in their state field of their packet.

The command DO_NOTHING indicates that this broadcast message is a synchronization message and does not contain any command. The BaseStation uses

this state to broadcast synch messages when it receives request from nodes for synch messages transmission when they boot. The synch messages are then periodically sent to the network.

The command IS_SYNCED is used to ask nodes in the network about their synchronization status when needed. The BaseStation then waits for the nodes' response about their status. When a node becomes synchronized it sends the status NO_DATA in its state field with is_synced field set to 1 to the BaseStation. When the BaseStation receives the synchronization confirmation from all the nodes in the network it stops sending synch messages and then issues START_SAMPLE command to the nodes to start sampling of the microphone sensors attached to

```
enum{ //commands for basestation cmd
    DO_NOTHING = 1,
    IS_SYNCED =2,
    START_SAMPLE = 3,
    START_TX = 4,
    REBOOT = 5,
};
enum{ // response statuses for motes state
    NO_DATA = 10,
    DATA_START = 11,
    DATA_STOP = 12,
    SAMPLE_DONE = 13,
    REBOOTED = 14,
};
```

Figure 4.4: Commands and states for BaseStation and nodes respectively

them. After sampling, all the motes inform the BaseStation about the completion of sampling and storing data in local flash tasks which in return issues the `START_TX` command to the first node only using unicast communication.

The first node reads the data from flash which it has previously stored and transmits it to the BaseStation only. Note that all the motes communicate with BaseStation on peer-to-peer basis. Only BaseStation broadcast certain messages when needed. Also the communication among BaseStation and motes is based on best effort approach. No acknowledgement of communication is carried out to save power as the data length is quite large (32bytes). When the first node successfully completes the transmission of data it sends `DATA_STOP` status to BaseStation to indicate end-of-data. BaseStation then issues the `START_TX` command to the second node. The second node transmits data and informs BaseStation about completion. The BaseStation then sends `START_TX` commands to the third node which also transmits its data and so on. The BaseStation is programmed such that it forwards only those packets to PC which contains microphone readings, all other messages are not forwarded to PC.

Now we will discuss the sync's side story. Fig. 4.5 gives the flowchart description of the sync application that is installed on every node in the network except the sink node.

`NO_DATA` is used to exchange signaling information like synchronization status, reboot status, sample done status and several others. We define two different `message_t` variables (buffers more accurately) in the application: `AMsignal`

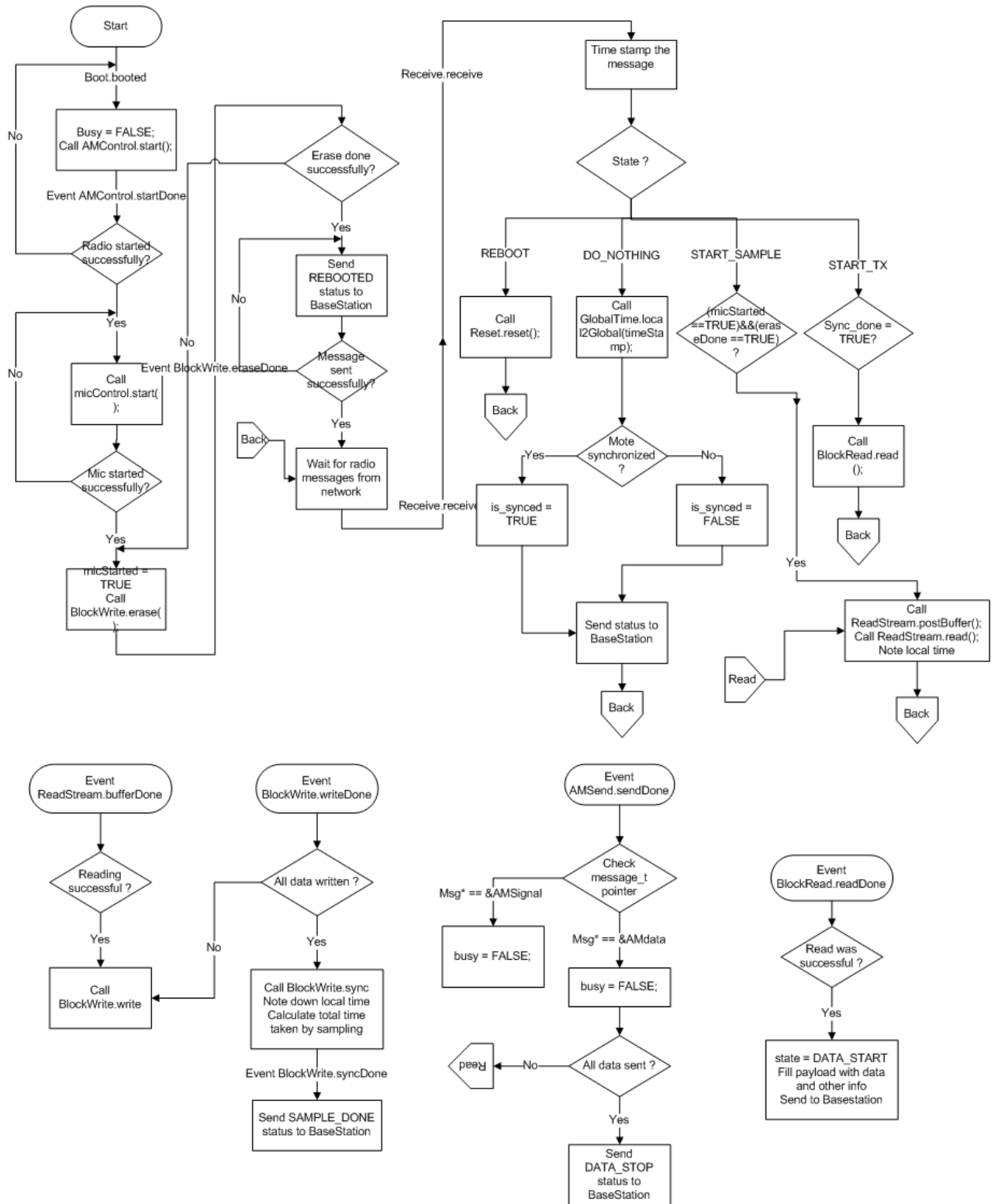


Figure 4.5: The structure of sync application

for transmitting signaling messages and AMdata for transmitting packets which contain microphone data. The data packet containing DATA_START state is transmitted using AMdata buffer. All other packets containing one of the remaining states are transmitted using AMsignal buffer. Two AM types are used: AM_BASESTRUCT (equals to 7) for receiving packets from BaseStation and AM_MOTESTRUCT (equals to 6) for sending packets to BaseStation. This strategy avoids inter-mote communication and all motes communicate with BaseStation only and the network congestion is reduced.

As will be explained in Section 4.6, TinyOS uses the concept of tasks and split-phase operation. Several tasks may be running in parallel in the system in split-phase operation sense and their completion is arbitrary depending on the application. Therefore, in the Fig. 4.5 we have represented these tasks in separate small flowcharts. There are four small flowcharts for the events of ReadStream.bufferDone (corresponding to the interface for reading sensor), BlockWrite.writeDone, BlockRead.readDone (corresponding to interfaces for writing and reading from flash memory) and AMSend.sendDone (corresponding to interface for sending radio packets to the network) besides the main application flowchart. Their interaction with the main flowchart is shown by a page-reference (like Back and Read).

The application when booted calls AMControl.start command to start the radio hardware of the node. After its successful start the application then starts the microphone hardware. After that the flash erase operation is carried out because

as in mentioned [37][38] an erase is necessary before writing to the flash memory of the node. After that the node sends the BOOTED state to the BaseStation in the state field. The node then waits for the radio packets from the BaseStation.

When a radio packet is received the node checks the payload for certain commands issued by BaseStation. The cmd may contain one of five possible commands of Fig. 4.4. If the cmd is REBOOT it means the BaseStation wants the nodes to be reset. So the node reset itself using the Reset interface.

If the cmd is DO_NOTHING it means this packet is a synchronization message. The node using FTSP algorithm converts its local clock to global time of BaseStation and updates a variable is_synced and sends the radio packet to the BaseStation. The is_synced may be TRUE or FALSE depending on the return value of GlobalTime.local2Global function. TRUE means node is synchronized to the BaseStation and FALSE means not synchronized.

As was discussed in this section previously the BaseStation sends synch messages periodically until all nodes become synchronized. If the cmd in the received message is START_SAMPLE it means that all the nodes are synchronized and the BaseStation wants all the nodes to start sampling of the microphone. So the node posts four buffers for sampling and then calls the ReadStream interface.

When the ReadStream fills the first buffer it signals the ReadStream.bufferDone event after which the application writes the sampled data to the flash using BlockWrite.write interface. After successful writing to flash the BlockWrite.writeDone event is signaled. The application then checks whether

the required number of samples have been read from mic if yes it issues Block-Write.sync command to the flash to sync the data it has written recently otherwise it again posts buffer to ReadStream interface. Sync is required for data to be stored permanently on flash otherwise the data will be lost after power off [38].

The application also notes down the time when first sample is read and the time when last sample is written to flash. This time difference is then used to calculate average sampling rate of the microphone.

After successful sampling operation, the node sends SAMPLE_DONE status to the BaseStation to indicate that it has finished sampling. When the BaseStation receive SAMPLE_DONE status from all nodes it issues START_TX command to the node1.

So if the cmd in the received message from BaseStation contains START_TX it means that the node needs to send all the mic data it has recently stored in the flash. The application keeps the records of number of messages it sends so that it sends the required number of messages. This number depends on the data. For example, in our application we have 16384 samples and we send 16 samples per message so there are total $16384/16=1024$ messages to be sent. Once all the data have been sent the node then sends a message with the state of DATA_STOP. The BaseStation can recognize this state. The BaseStation then sends START_TX command to the next node, for example, node # 2 and so on. Similarly all the nodes in the network send their mic readings one by one to the BaseStation which forwards it to the PC on serial port.

4.6 Results and Discussion

In this section we will discuss different experimentations we carried out and the analysis based on them.

4.6.1 Error Sources and Uncertainties

During the development we faced several issues. These will be discussed one by one but let's first discuss the split-phase operation of TinyOS which is the main cause of these issues (in the scenario of this specific application only).

Split-phase Operation in TinyOS 2.x

“Hardware is almost always split-phase rather than blocking” [20]. Split-phase operation means that the request for an operation is completed immediately (for example, request for reading a sensor attached to ADC) but actual completion of the operation takes sometime depending on the hardware resources. The completion of actual operation is signaled later by a separate call back.

Now the software may be required to operate either in synchronous mode or split-phase. For proper implementation the software and hardware behavior must be the same. So the solution is to either make the reading interface split-phase or make the ADC synchronous. TinyOS takes the former approach [20]. Rather than making everything synchronous through threads, operations that are split-phase in hardware are split-phase in software as well. This means that many common operations, such as sampling sensors and sending packets, are split-phase. Almost

all the interfaces we use in the applications are split-phase. As we will see in the coming sections, this split-phase operation is very unfavorable to applications which sample the sensors at high rate and require that the nodes should be strictly synchronized.

Uncertainties in Sampling

The most critical interface is ReadStream. The ReadStream interface's design is based on Read interface which works on the principle of reading only one sample at a time and signaling back this sample to the application. The Read interface does not have the capability to read ADC continuously and hence the ReadStream interface which uses it encounters a great deal of delay of arbitrary nature between any two consecutive samples. This reduces not only the sampling rate but also affect the continuity of the sampling. Since the sample interval is arbitrary, the samples from different nodes are not strictly concurrent even if their local clocks are synchronized and therefore they don't correspond to the same physical event.

This issue is very critical for our application. Our application requires that all the readings from all the nodes should correspond to the same real time happening. So using the TinyOS 2.x standard distribution ReadStream interface we cannot achieve real time concurrent sampling for all nodes.

The second problem is with achieving higher sampling rate. Using ReadStream we were able to achieve only 4.9 kHz average sampling rate. Obviously this sampling rate is not enough for audio signal processing. The literature says that we can achieve higher sampling rate by sampling ADC in free running mode but

that would mean modifying several components of TinyOS standard distribution along with their interfaces at different levels of abstraction. This proves that there is a need of enhancing TinyOS interfaces and components to accommodate high sampling rate applications.

Uncertainties in Flash Reading/Writing

Now the problems is that both the read and write interfaces (BlockRead and BlockWrite) for the flash are split-phase and so all the issues faced in sampling operation are faced in this scenario as well.

Since every ADC has a limited buffer which contains the recent reading, we need to extract data from this buffer and store it in the flash before the ADC start sampling the next reading. This means that we have to empty the buffer and write the data to flash in a time period less than or equal to the sampling interval of the microphone otherwise the ADC will overwrite the buffer with the new reading and the old reading will be lost. This requires that write interface of the flash should be able to write the data synchronously and fast enough for successful sampling. However, as we mentioned that BlockWrite interface (and all other interfaces of flash memory) are split phase and thus the writing operation is not synchronous or continuous. Instead it is arbitrary and the time interval between any two writing operations is also arbitrary. This creates problem for the ADC to successfully deliver a synchronous high rate read operation.

This situation is explained in Fig. 4.6. The sampling interval is denoted by t_s and is arbitrary. Let's assume that t_s is between time t_0 and t_1 . Similarly t_w

represents the flash writing time and let's assume this time is between t_2 and t_3 . Now for every single sampling/writing operation the total time taken t_{sw} is given by

$$t_0 + t_2 < t_{sw} < t_1 + t_3 \quad (4.1)$$

Equation (4.1) signifies the fact that for every node the interval t_{sw} is not constant and hence the sampling/writing operation takes arbitrary time for every node.

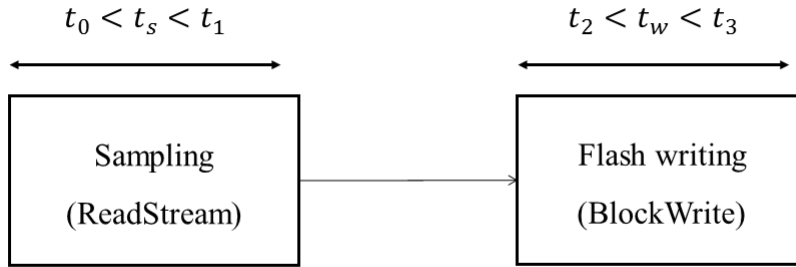


Figure 4.6: Analyzing the arbitrariness of ReadStream and BlockWrite

Synchronization Issues

We also faced problems in achieving strict synchronization among the motes. To analyze how the synchronization works and whether the selected synchronization scheme (FTSP [33]) will achieve the goal we performed certain experiments:

1. Time stamping different events in the application
2. Changing the number of synch messages sent per second to the network
3. Changing the total number of samples to be read

We used a modified version of the sync application discussed in Section 4.5. The number of samples to be read was 16384. We time-stamped the message arrival time for `START_SAMPLE` command from BaseStation to check whether all the motes receive the command at the same time. Then we start sampling and note down the local clock value when the first sampling buffer is signaled to see the timing of the first buffer completion at each node. We also note down the local time when the last sampling buffer is signaled.

Literature for FTSP [33] says that at least one synch message from beacon to the network is required for proper synchronization. However, as we will show the synchronization is also heavily dependent on the local processing on the mote. So to check the performance we played with the synch message period i.e. the number of synch messages sent to network per second from beacon.

The motes were placed next to each other. The experiment was performed to check the timing of different events so there was no audio source because we are not interested in the signal itself. Fig. 4.7 shows the output of two experiments, the first experiment was done with sync message period of 1second and the second experiment was done with sync message period of 500mSec i.e. two messages/second.

As can be seen from Fig. 4.7a and 4.7b, the average sampling frequency decreases from 3.0 kHz to 2.4 kHz when the number of synch messages are increased from 1 to 2/sec. This is because in the second case the local MCU of the mote has to do extra processing (receiving extra message and performing synchroniza-

tion tasks twice a second compared to once in the first case) which disturb the process of sampling the microphone. This issues arise from the fact that MCU is sequential by nature, the parallelism of TinyOS is only in software sense.

```

Sampling Frequency of Mote1 = 3072.917883
Sampling Frequency of Mote1 = 3059.176090
Sampling Frequency of Mote1 = 3082.023747
Mote1: START_SAMPLE cmd arrival from BS Time Stamp = 1823
Mote2: START_SAMPLE cmd arrival from BS Time Stamp = 1823
Mote3: START_SAMPLE cmd arrival from BS Time Stamp = 1823
Mote1: Time of 1st sample = 2774815
Mote2: Time of 1st sample = 2775712
Mote3: Time of 1st sample = 3663319
Time Difference in Mote 1 & 2 = -897 uSec
Time Difference in Mote 2 & 3 = -887607 uSec
Time Difference in Mote 3 & 1 = 888504 uSec
Mote1: Time of last sample = 6118176
Mote2: Time of last sample = 6137625
Mote3: Time of last sample = 7006283
Time Difference in Mote 1 & 2 = -19449 uSec
Time Difference in Mote 2 & 3 = -868658 uSec
Time Difference in Mote 3 & 1 = 888107 uSec

```

(a)

```

Sampling Frequency of Mote1 = 2457.516704
Sampling Frequency of Mote1 = 2446.188940
Sampling Frequency of Mote1 = 2443.109647
Mote1: START_SAMPLE cmd arrival from BS Time Stamp = 1318
Mote2: START_SAMPLE cmd arrival from BS Time Stamp = 1318
Mote3: START_SAMPLE cmd arrival from BS Time Stamp = 1318
Mote1: Time of 1st sample = 2320246
Mote2: Time of 1st sample = 2320244
Mote3: Time of 1st sample = 2316661
Time Difference in Mote 1 & 2 = 2 uSec
Time Difference in Mote 2 & 3 = 3583 uSec
Time Difference in Mote 3 & 1 = -3585 uSec
Mote1: Time of last sample = 5664560
Mote2: Time of last sample = 5684902
Mote3: Time of last sample = 5686335
Time Difference in Mote 1 & 2 = -20342 uSec
Time Difference in Mote 2 & 3 = -1433 uSec
Time Difference in Mote 3 & 1 = 21775 uSec
Fs = 2.4487e+003

```

(b)

Figure 4.7: Analyzing the effect of no. of sync message/s on synchronization, (a) 1000 mSec case, (b) 500 mSec case.

Looking at Fig. 4.7a and 4.7b, we observe that all motes are strictly synchronized at the time when they receive the `START_SAMPLE` command from BaseStation because the time stamp for this message reception is the same on all motes. However, the time for the first sample buffer event is not the same for all motes. This may be due to two reasons:

1. The local times of the motes are no longer synchronized because of the clock drift
2. Sample done event was not signaled at the same time because of the arbitrary nature of `ReadStream` interface on all motes.

We call this situation as uncertainty because we don't know the real reason of the problem. So we have two uncertainties here: i) synchronization uncertainty and ii) sampling uncertainty.

Fig. 4.7 clearly shows that in case (b) the time difference of sample buffer done event is less compared to case (a) both for the first sample and the last sample buffer. This verifies that the more the number of synch messages/second sent to the network the more it gets synchronized. However we cannot increase the number of synch messages more than 2/seconds because it heavily disturbs other processing going on the local MCU of the motes as is evident in Fig. 4.7 for the sampling frequency that has decreased due to increase in number of sync messages per second.

We performed another experiment playing with the total number of samples to be taken from microphone while keeping the sync message period constant (1

second). We performed the experiment for two cases: i) 4096 samples ii) 8192 samples. Fig. 4.8 shows the results of these experiments.

Here in this case as well, the motes are strictly synchronized when they receive `START_SAMPLE` command from BaseStation. The arbitrariness of sample done event remains but there are two things to be noted:

1. The average sampling frequency reduces as the numbers of samples reduce. It is 1.5 kHz for case (a) and 2.1 kHz for case (b)
2. The time difference for the last sampling buffer done event is more in case (b) than case (a) which clarifies that since in case (a) the number of samples is less and hence it takes less time and hence little clock drift is observed as compared to case (b) where there are more samples (double the no. in case (a)) to be read which take more time and hence the clock drift occurring in this time is more.

So as the time elapses, due to clock drifts, the local clocks of the motes get more and more un-synchronized.

These experiments confirm that the FTSP algorithm was not able to synchronize the network in our case and the synchronization depends on several parameters. The local clocks behavior is also arbitrary due to the inherent drift in the clock.

Uncertainties in Radio and Serial Communication

We encountered the following problems in radio and serial communication:

```

Sampling Frequency of Mote1 = 1508.338994
Sampling Frequency of Mote1 = 1505.173328
Sampling Frequency of Mote1 = 1504.135432
Mote1: START_SAMPLE cmd arrival from BS Time Stamp = 1318
Mote2: START_SAMPLE cmd arrival from BS Time Stamp = 1318
Mote3: START_SAMPLE cmd arrival from BS Time Stamp = 1318
Mote1: Time of 1st sample = 2320247
Mote2: Time of 1st sample = 2320313
Mote3: Time of 1st sample = 2316663
Time Difference in Mote 1 & 2 = -66 uSec
Time Difference in Mote 2 & 3 = 3650 uSec
Time Difference in Mote 3 & 1 = -3584 uSec
Mote1: Time of last sample = 3118601
Mote2: Time of last sample = 3121741
Mote3: Time of last sample = 3118880
Time Difference in Mote 1 & 2 = -3140 uSec
Time Difference in Mote 2 & 3 = 2861 uSec
Time Difference in Mote 3 & 1 = 279 uSec

```

(a)

```

Sampling Frequency of Mote1 = 2145.595489
Sampling Frequency of Mote1 = 2135.533437
Sampling Frequency of Mote1 = 2134.508167
Mote1: START_SAMPLE cmd arrival from BS Time Stamp = 1319
Mote2: START_SAMPLE cmd arrival from BS Time Stamp = 1319
Mote3: START_SAMPLE cmd arrival from BS Time Stamp = 1319
Mote1: Time of 1st sample = 2295104
Mote2: Time of 1st sample = 2295102
Mote3: Time of 1st sample = 2291450
Time Difference in Mote 1 & 2 = 2 uSec
Time Difference in Mote 2 & 3 = 3652 uSec
Time Difference in Mote 3 & 1 = -3654 uSec
Mote1: Time of last sample = 3969020
Mote2: Time of last sample = 3981714
Mote3: Time of last sample = 3978948
Time Difference in Mote 1 & 2 = -12694 uSec
Time Difference in Mote 2 & 3 = 2766 uSec
Time Difference in Mote 3 & 1 = 9928 uSec

```

(b)

Figure 4.8: Analyzing the effect of no. of samples on synchronization, (a) 4096 samples case, (b) 8192 samples case

1. Radio packet loss (about 20%)
2. Radio packet length variations
3. Serial packet length variations

The radio packet loss is a serious issue. We can lose critical data such as impulsive peak corresponding to the gunfire because of its short duration. A peak may be contained in 2 or 3 packets only because of the low sampling frequency and 20% loss is "enough" to lose this peak.

The second issue is the packet length variation. The received packet at PC does not have a constant length. The length varies by a few bytes but this also complicates the programming at the PC. The issues may be due to two reasons:

1. The length of packet varies because of the local clock drifts at the transmitting node and receiving node (BaseStation)
2. The length of packet varies because of local clock drift in the BaseStation relative to PC clock.

Now this situation is also arbitrary and uncertain. So this is another uncertainty adding to the issues we are facing in our application. The problem of packet length variation can be solved by padding (truncating) in the case of short (long) packets.

Joint Uncertainty

The uncertainties discussed in previous subsections 1 – 5 contribute to a joint uncertainty that makes the task of debugging and application analysis difficult. It drastically affects our application being sensitive to synchronization, sampling and communication.

It is to be noted that these uncertainties come from the split-phase design

of the interfaces and components of the TinyOS. One can develop his/her own interfaces and components directly accessing the hardware and taking care of minute details. For example for continuous sampling we can take advantage of the free running mode of ATmega1281 microcontroller [36]. In free running mode the MCU constantly samples the sensor and updates its buffer. The user has to take care of taking the data out of the register. Similarly for fast flash reading/writing hardware specific code can be written. In [29] the authors have written their own code for microphone sampling, flash reading/writing and other services. They were able to achieve a sampling rate of 17.73 kHz and a good synchronization among the notes by modifying the FTSP algorithm and incorporating some other techniques from the literature. The authors, however, did not share their code as it is proprietary.

4.6.2 An Example Experiment

Fig. 4.9 shows a configuration of three sensor nodes placed on a line 50cm apart from each other. The nodes were programmed with the sync application of Section 4.5. Fig. 4.10 shows the plotted sampled data that was sampled from the three notes. The audio source was an unloaded toy gun. The gun was fired at node 1.

The data peak which represents the shot occurs for node 1 at sample time 6356, for node 2 at sample time 7024 and for node 3 at sample time 6487. All the nodes are sampling at an approximate average sampling rate of 4.9 kHz. Using this sampling rate we can convert the peak sample time to real time. These are



Figure 4.9: Nodes configuration for the experiment, (a) Front View, (b) Side View.

given by

$$\begin{aligned}
 t_1 &= \frac{6356}{4.9kSPS} = 1.297s, & t_2 &= \frac{7024}{4.9kSPS} = 1.433s, \\
 t_3 &= \frac{6487}{4.9kSPS} = 1.323s
 \end{aligned} \tag{4.2}$$

while t_1, t_2, t_3 represent the peak time for node 1, 2 and 3 respectively.

Clearly the timings are invalid because a distance of 50cm corresponds to a time delay of $\frac{50cm}{340m/s} = 1.47mSec$ while the time difference of arrival at each node for the gunshot is in the order of hundreds of milliseconds.

This strange behavior of the nodes can be explained by the following reasons:

1. The local clocks of all nodes are not synchronized to each other.
2. Several packets were lost in the radio/serial communication and the peaks do not correspond to the same value of the signal but to different peaks in the same signal at different times. Note that the experiment was performed in a hall so the echoes may correspond to the peaks as well.

3. The data were not properly written to /read from flash. For example, some of the data were overwritten because of fast sampling compared to slow writing of the flash.
4. The data were overwritten in the ADC buffer by the ADC before the application was able to extract data out of the buffer and write it to the flash.

So any or all of the uncertainty discussed in Section 4.6.1 may contribute to the erroneous results of the experiment. This shows that the standard TinyOS interfaces and their relevant components are not suitable for applications which involve high sampling rate, strict synchronization and fast data storage and trans-

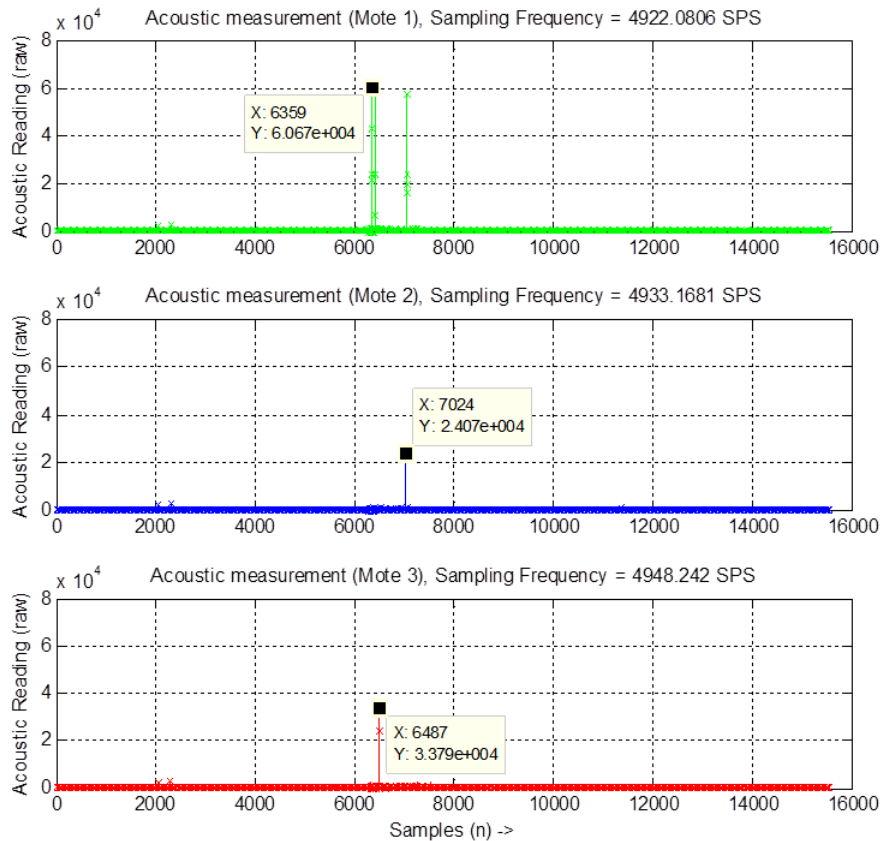


Figure 4.10: Results for the experiment conducted in Fig. 4.9

fer. Instead, one needs to change the existing system code of TinyOS or develop his/her own code entirely from the hardware level to operating system level to achieve the accuracy needed.

4.7 Conclusion

In this chapter we presented the development of a WSN for impulsive acoustic source localization and the issues encountered during the development process. The development was totally based on using the TinyOS 2.x standard interfaces and components.

Five error sources were identified that can significantly degrade the WSN performance if not handled in a custom way. The effect of each of these error sources was verified through a specific experiment and the development of specific applications. Several recommendations were given to overcome such errors.

CHAPTER 5

THE ORTHOGONAL CLUSTERING ALGORITHM AND EXPERIMENTAL SETUP

In this chapter we will discuss the Orthogonal Clustering (OC) algorithm that we will use for our acoustic source localization application as well as the hardware system and TDOA geometries. Section 5.1 describes the steps that are involved in the OC algorithm for TDE. Section 5.2 describes in detail the TDOA geometries we are using to analyze and compare the performance of the OC and CC TDE methods. We discuss the hardware system used to carry out the experiments in Section 5.3.

5.1 The Orthogonal Clustering (OC) Algorithm

In this section we describe the main steps that are involved in the OC method for TDE estimation. We will only give the important mathematical expressions of the algorithm and will not go into the mathematical detail of the algorithm as it has already been explained very well in [5] and [6].

5.1.1 Problem Development

Suppose there are N sensors and they are placed in a Cartesian Coordinate system. An impulsive acoustic source generates a signal in the same system. Every sensor will capture the signal. Let sensor 0 be the reference sensor for all other sensors. The signal received at a particular sensor will be an advanced or delayed version of the signal received at the sensor 0. If the size of the Cartesian Coordinate system is $X \times Y$, where X (in meters) is the maximum size of x -axis and Y (in meters) is that of y -axis, then the maximum time difference between the signals of any two sensors j and i that can be observed will be

$$\tau_{ji} = \frac{\sqrt{X^2 + Y^2}}{c} \quad (5.1)$$

where c is the speed of the sound wave. We can formulate this scenario into a matrix notation. Let Ψ be an $M \times N$ matrix and each column of the Ψ represents the delayed version of the original signal. Column 1 represents the zero-delay signal (at time 0) as shown in Fig. 5.1, column 2 represents the signal delayed by

one sample and so on. N is the length of the signal (e.g., 16000 samples) and M is the expected number of delays. We keep $M = 16000$ to accommodate a time delay up to 1 second (at 16 kHz sampling rate) which is enough for our application (in our application the delays are not more than few milliseconds). The Ψ matrix is called the Dictionary or Sensing matrix. Let \mathbf{y} be the observed signal we captured at a sensor then we can write,

$$\mathbf{y} = \Psi\mathbf{x} + \mathbf{n} \tag{5.2}$$

where \mathbf{x} is a vector representing the Channel Impulse Response (CIR) of the environment (indoor or outdoor) and \mathbf{n} is the complex additive white Gaussian noise. Since we are assuming a single acoustic source, \mathbf{x} will be a sparse vector and will have few nonzero values corresponding to the original signal and its reflections. We call this number of nonzero values as the Sparsity or the Support of the \mathbf{x} and represent it by \mathcal{S} .

The Dictionary matrix Ψ has a structure (resembling that of a Toeplitz matrix) and this information can be utilized to reduce computational complexity to find \mathbf{x} . The authors in [5] established an algorithm which they denote as Orthogonal Clustering (OC). They utilize the CS knowledge and the structure of Ψ along with other a priori statistical information to efficiently estimate the CIR while reducing the computational burden on the processing system. The next subsection describes the algorithm in details.

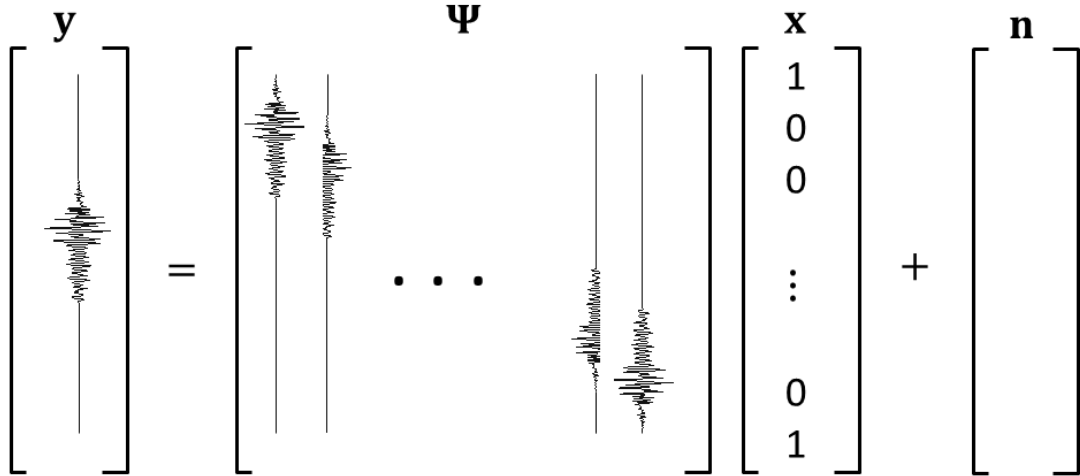


Figure 5.1: Problem formulation for OC

5.1.2 The OC Algorithm

The \mathbf{x} in (5.2) is modeled as $\mathbf{x} = \mathbf{x}_B \odot \mathbf{x}_G$, where \odot represent element by element multiplication. \mathbf{x}_B is a random process whose elements are independent and identically distributed (iid) with Bernoulli distribution and those of \mathbf{x}_G are also iid but with some unknown zero mean distribution. If \mathbf{x} has sparsity \mathcal{S} then (5.2) can be written as

$$\mathbf{y} = \Psi_{\mathcal{S}} \mathbf{x}_{\mathcal{S}} + \mathbf{n} \quad (5.3)$$

where $\Psi_{\mathcal{S}}$ is the sub-matrix formed by only those columns corresponding to nonzero values of \mathbf{x} i.e. $\{\psi_i : i \in \mathcal{S}\}$. The Minimum Mean Square Estimate (MMSE) of \mathbf{x} given the observation \mathbf{y} is given by [5]

$$\hat{\mathbf{x}}_{MMSE} = \mathbb{E}[\mathbf{x}|\mathbf{y}] = \sum_{\mathcal{S}} p(\mathcal{S}|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}] \quad (5.4)$$

where $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$ is simply the linear MMSE of \mathbf{x} given \mathbf{y} if \mathbf{x} conditioned on its support is Gaussian [5][6] i.e.

$$\mathbf{x}_{\mathcal{S}} = \sigma_x^2 \Psi_{\mathcal{S}}^H \Sigma_{\mathcal{S}}^{-1} \mathbf{y} \quad (5.5)$$

where

$$\Sigma_{\mathcal{S}} = \frac{1}{\sigma_n^2} \mathbb{E}[\mathbf{y}\mathbf{y}^H | \mathcal{S}] = \mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2} \Psi_{\mathcal{S}} \Psi_{\mathcal{S}}^H \quad (5.6)$$

where σ_n, σ_x are the variances of \mathbf{x} and the noise respectively and H represents the complex conjugate transpose. If $\mathbf{x}|\mathcal{S}$ is non-Gaussian then the Best Linear Unbiased Estimate of \mathbf{x} is

$$\mathbf{x}_{\mathcal{S}} = (\Psi_{\mathcal{S}}^H \Psi_{\mathcal{S}})^{-1} \Psi_{\mathcal{S}}^H \mathbf{y} \quad (5.7)$$

$p(\mathcal{S}|\mathbf{y})$ is calculated using Bayes rule [5]

$$p(\mathcal{S}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{S})p(\mathcal{S})}{\sum_{\mathcal{S}} p(\mathbf{y}|\mathcal{S})p(\mathcal{S})} \quad (5.8)$$

The expressions for the two unknowns i.e. $p(\mathbf{y}|\mathcal{S})$ and $p(\mathcal{S})$ are given in [5]. The MMSE formulation in the case when the $p(\mathcal{S}|\mathbf{x})$ is unknown and the Maximum A Posteriori (MAP) estimate of \mathbf{x} have also been covered in [6] in detail. Since we are using the MMSE estimate of \mathbf{x} in our application we will not discuss the MAP estimation of \mathbf{x} .

Now let \mathcal{S} be the support of \mathbf{x} then (5.3) can be written as [5][6]

$$\mathbf{y} = \begin{bmatrix} \Psi_{\mathcal{S}_1} & \Psi_{\mathcal{S}_2} & \dots & \Psi_{\mathcal{S}_P} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_P \end{bmatrix} + \mathbf{n}. \quad (5.9)$$

where P is the maximum number of clusters formed from the columns of Ψ because of orthogonality (or semi-orthogonality as the authors in [5] call it) among them due to the structure of the matrix. \mathcal{S} is the support set corresponding to the i th cluster (with $i = 1, 2, \dots, P$). This clustering allows us to write (5.4) as

$$\hat{\mathbf{x}}_{AMMSE} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_P \end{bmatrix} = \begin{bmatrix} \mathbb{E}[\mathbf{x}_1|\mathbf{y}] \\ \mathbb{E}[\mathbf{x}_2|\mathbf{y}] \\ \vdots \\ \mathbb{E}[\mathbf{x}_P|\mathbf{y}] \end{bmatrix} = \begin{bmatrix} \sum_{\mathcal{S}_1} p(\mathcal{S}_1|\mathbf{y})\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}_1] \\ \sum_{\mathcal{S}_2} p(\mathcal{S}_2|\mathbf{y})\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}_2] \\ \vdots \\ \sum_{\mathcal{S}_P} p(\mathcal{S}_P|\mathbf{y})\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}_P] \end{bmatrix} \quad (5.10)$$

$\hat{\mathbf{x}}_{AMMSE}$ means approximate MMSE of \mathbf{x} . Equation (5.10) means that $\hat{\mathbf{x}}_{AMMSE}$ can be calculated in a divide-and-conquer manner by separately evaluating each cluster either in parallel, if possible, or one by one which reduces the computation burden on a processing platform considerably.

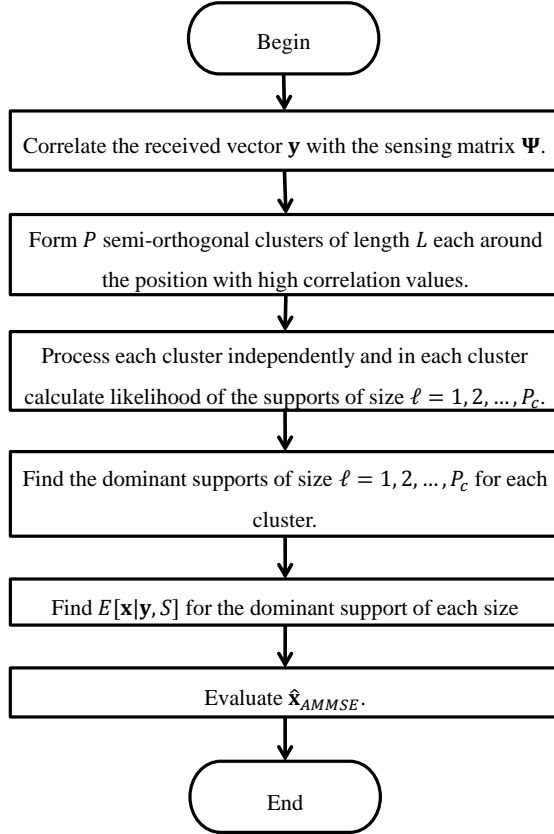


Figure 5.2: Flowchart of the OC algorithm showing the main steps

Now we describe the steps that the OC algorithm takes to estimate $\hat{\mathbf{x}}_{MMSE}$.

Fig. 5.2 shows these steps in the form of a flow chart. We discuss these steps next in detail.

Step1: Determining Dominant Position

The first step of the algorithm is to find the dominant support of $\hat{\mathbf{x}}$ and this can be achieved by correlating the observation vector \mathbf{y} with the sensing matrix Ψ in (5.2). This will give us an initial guess of the sparsity of $\hat{\mathbf{x}}$.

Step2: Forming Semi-orthogonal Clusters

The vector of N correlations obtained from Step1 can help us create clusters. First an index i_1 is selected that corresponds to the largest correlation and a cluster of size L is formed around it. The value of L is selected according to a correlation tolerance ϵ . Then another cluster is formed around the next largest correlation with index i_2 and so on until a total P non-overlapping clusters are formed. If two clusters are overlapping then they are combined into one big cluster.

Step3: Finding the Dominant Support and their Likelihood

The next step is to find the maximum possible support size P_c in a cluster i of size L_i for each of the P clusters formed in Step 2. This is done by finding the likelihood for all supports of size $\ell = 1, 2, \dots, P_c$. P_c is calculated by the following formula [5]

$$P_c = \lceil \operatorname{erfc}^{-1}(10^{-2})\sqrt{2L_i p(1-p)} + L_i p \rceil \quad (5.11)$$

where p is the probability of success for Binomial Distribution $B(L, p)$ and $\lceil \cdot \rceil$ is the Ceil function which maps its argument to the smallest following integer. Each cluster is processed independently because of the semi-orthogonality between them.

Step4: Finding $\mathbb{E}[\mathbf{x}|\mathbf{y}]$

After we have the dominant supports and their likelihood then we can find $\mathbb{E}[\mathbf{x}|\mathbf{y}]$ using (5.5) or (5.7) for each size of the support.

Step5: Evaluating the Estimate of \mathbf{x}

Finally using (5.4) we can find the $\hat{\mathbf{x}}_{MMSE}$ as we have all the necessary components determined.

After developing the mathematical model now we are ready to move ahead with the implementation of the algorithm. Section 5.2 discusses the TDOA geometries in different scenarios we have developed for implementing the impulsive acoustic source localization system using CC and OC methods for TDE.

5.2 Experimental Scenarios and Sensors Geometries

Fig 5.3 shows the experimental scenarios that we are taking into consideration for the implementation of our application. Broadly there are two scenarios: i) Indoor experiments and ii) Outdoor experiments. There are two different wireless sound acquisition systems: i) RevoLabs[®] Wireless HD Microphone System and ii) VocoPro[®] UHF8800 Wireless Microphone System. Three different TDOA geometries were considered to analyze not only the performance of OC/CC but also the effect of geometry on the efficiency of a source localization system. Furthermore, we considered two different scenarios in indoor experiments i.e. i) experimenting in the center of the hall and ii) experimenting in a corner of the hall to see the effect of reverberation and reflections on the performance of the system. Note that in Fig 5.3 we display three sampling rates i.e. 4kHz, 8kHz and 16kHz but the

OC algorithm could not be run for the 16kHz case due to its huge requirement of the PC memory. The available PC were not able to run the OC algorithm. Now we will discuss the sensor geometries in Section 5.2.1.

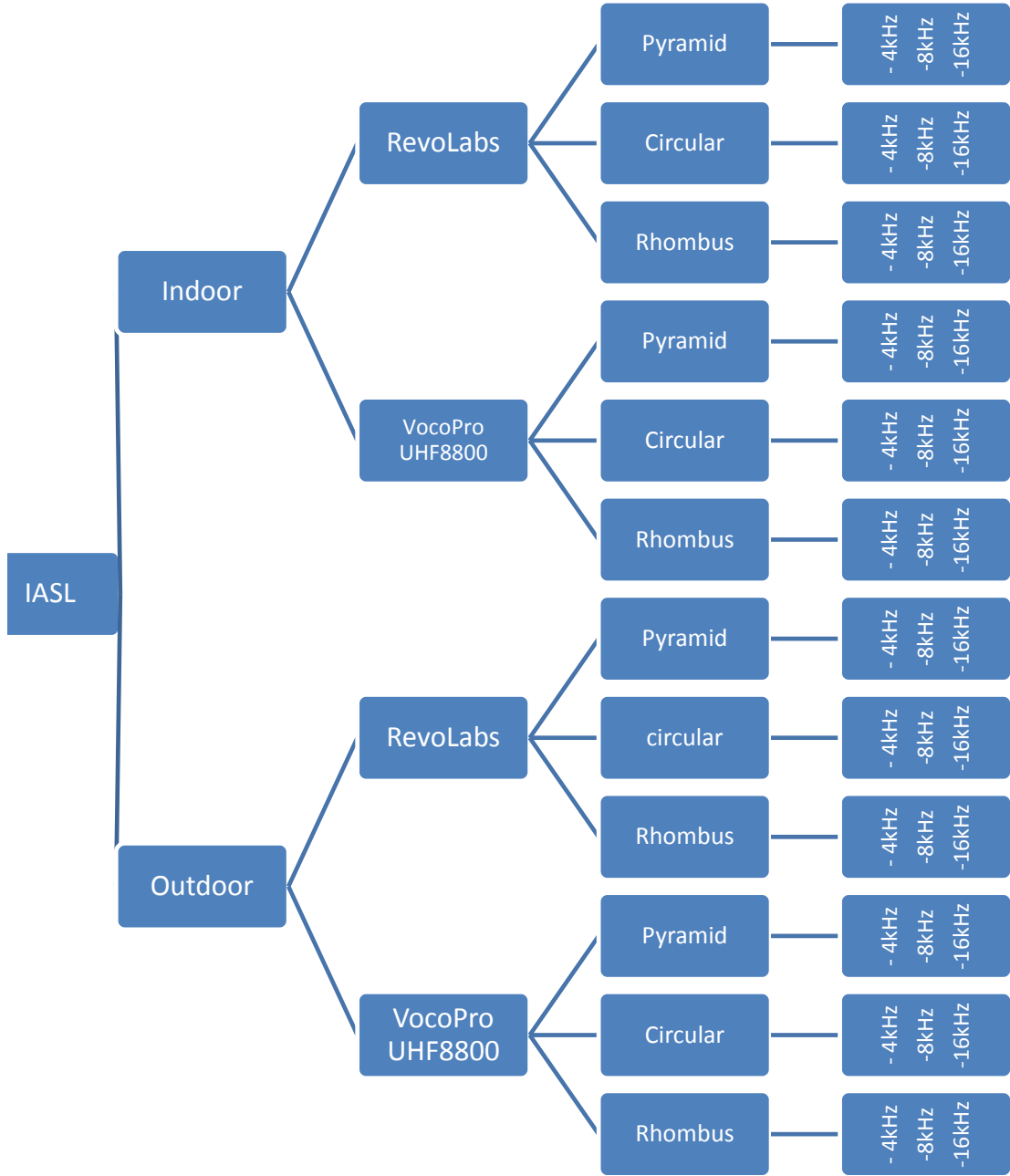


Figure 5.3: Different Scenarios Considered in the Experimentation

5.2.1 TDOA Geometries

In this section we discuss the sensor geometries to be used in the experimentations.

There are three different geometries and they are discussed next.

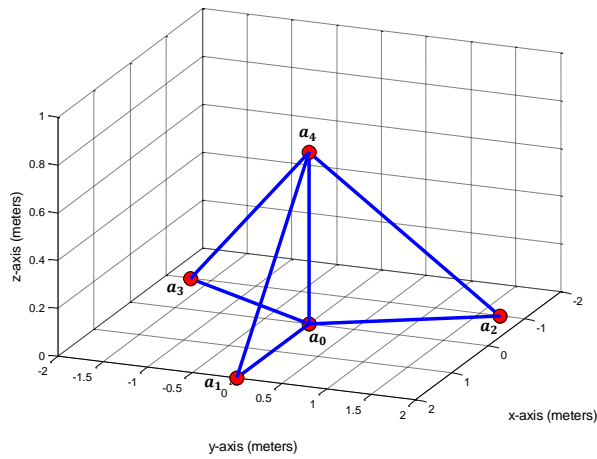
Pyramid Geometry

Fig. 5.4 shows the Pyramid geometry of five microphones. One of the microphones is at origin and has been taken as a reference for the remaining microphones. One of the sensors is on the z -axis (elevated) and the remaining three sensors are in the xy -plane at an angle of 120° from each other. Although the array size can be scaled up or down but we have kept the sensors positions as follows to restrict the array to a $2 \times 2 \times 2m^3$ space:

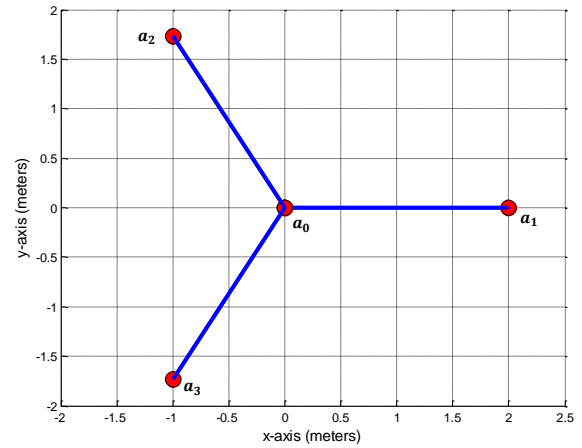
$$\begin{aligned} mic0 = a_0 &= (0, 0, 0), & mic1 = a_1 &= (2, 0, 0), \\ mic2 = a_2 &= (2 \cos 120^\circ, 2 \sin 120^\circ, 0), & mic3 = a_3 &= (2 \cos 240^\circ, 2 \sin 240^\circ, 0), \\ mic4 = a_4 &= (0, 0, 0.72) \end{aligned}$$

Note that in the practical implementation (Fig 5.4c) the array ground was considered at $89cm$ height from the real ground to avoid reflections from the ground and other effects of the ground as was discussed in Section 2.3.3.

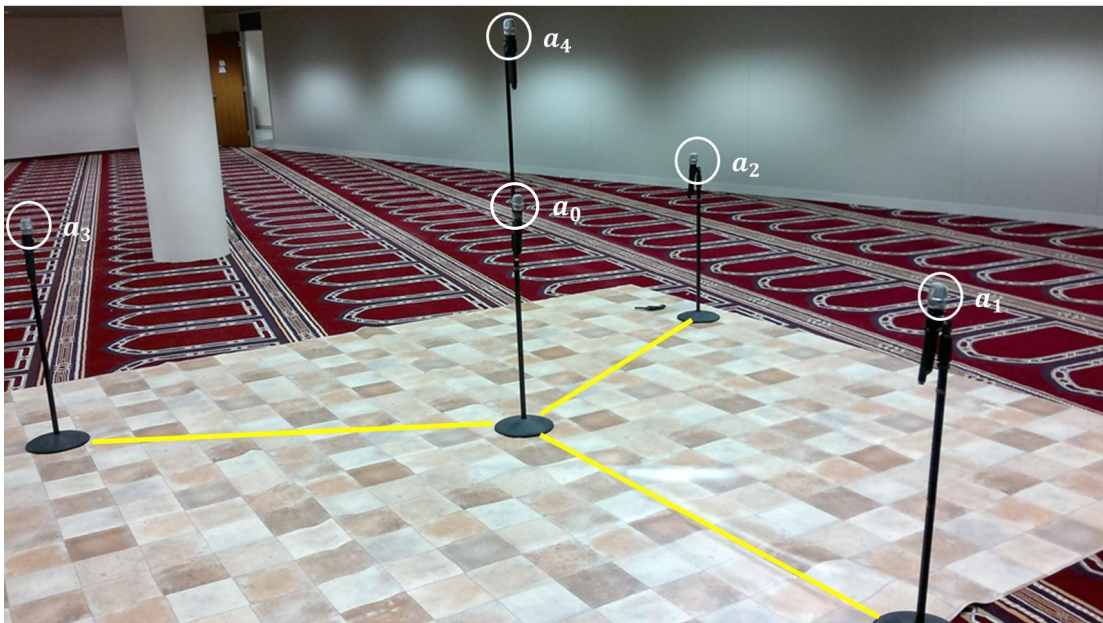
To select source positions for the experimentation we selected locations which cover some important areas of the geometry. Since the geometry is symmetric around the origin, the results of these points will represent other areas of the geometry as well.



(a)



(b)



(c)

Figure 5.4: TDOA Pyramid Geometry with VocoPro System. (a) Pyramid Geometry 3D plot (b) Pyramid Geometry 2D plot (c) Pyramid Geometry, Indoor Experiment with VocoPro System

Fig. 5.5 shows the source locations (shown as green dots) that we have selected for our experimentation. These points will be used in all our experiments both in

indoor and outdoor scenarios. Table 5.1 shows these source locations with their description and mathematical representation.

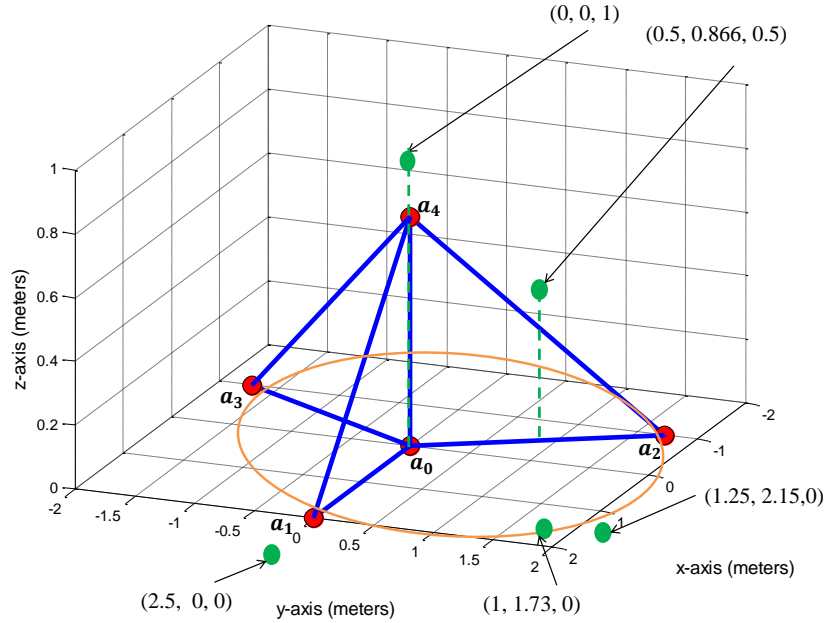


Figure 5.5: Selection of source locations of Pyramid geometry for experimentation

Table 5.1: Source locations for Pyramid geometry experiments

Source Location	$(\cos 60^\circ, \sin 60^\circ, 0.5) = (0.5, 0.866, 0.5)$	$(0, 0, 1)$	$(2.5 \cos 60^\circ, 2.5 \sin 60^\circ, 0) = (1.25, 2.16, 0)$	$(2 \cos 60^\circ, 2 \sin 60^\circ, 0) = (1, 1.73, 0)$	$(2.5, 0, 0)$
Description	In-between mic1 and 2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between mic1 & 2, outside the array	In-between mic1 & 2 on the circle inscribing the array.	Along the line (outside array) passing through mic1 and center of the array

Circular Geometry

Fig. 5.6 shows the arrangement of microphones which form a shape of circle when viewed from top. We cannot place all the sensors on a planer circle as then the Φ matrix discussed in Section 3.2.2 will become singular and no solution can be

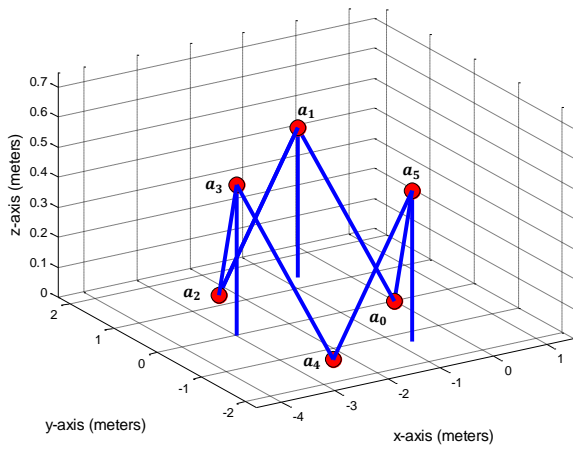
found. The sensor (microphone) locations are:

$$\begin{aligned}
 a_0 &= (0, 0, 0), & a_1 &= (r \cos 60^\circ, r \sin 60^\circ, 0.5), \\
 a_2 &= (r \cos 120^\circ, r \sin 120^\circ, 0), & a_3 &= (r \cos 180^\circ, r \sin 180^\circ, 0.5), \\
 a_4 &= (r \cos 240^\circ, r \sin 240^\circ, 0), & a_5 &= (r \cos 300^\circ, r \sin 300^\circ, 0.5)
 \end{aligned}$$

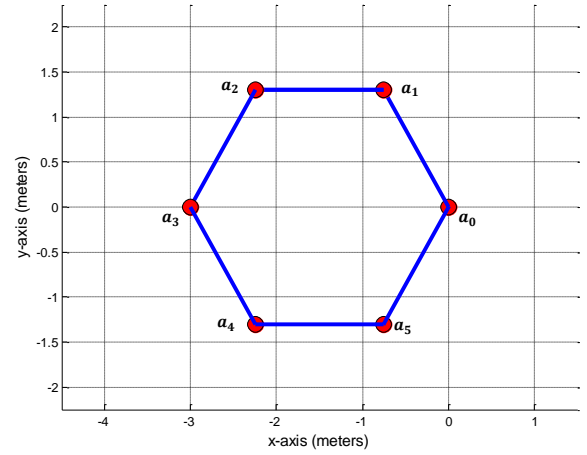
where $r = 1.5m$ is the radius of the circle. Since one of the microphones should be at the origin to satisfy the equations we developed in Section 3.2.2, we shifted the array along the x -axis by $-r$ such that mic0 would be at the origin and the array would be in two quadrants, in this case, it is in quadrant 2 and 3.

Note that in the practical implementation (Fig 5.6c) the array ground was considered at $89cm$ height from the real ground to avoid reflections from the ground and other effects of the ground as was discussed in Section 2.3.3.

To select the source positions for the experimentation we selected locations which cover the important areas of the geometry. Fig. 5.7 shows the source locations (shown as green dots) that we have selected for our experimentation. These points will be used in all our experiments both in indoor and outdoor scenarios. Table 5.2 shows the source locations with their description and mathematical representation.



(a)



(b)



(c)

Figure 5.6: TDOA Circular Geometry with RevoLabs System. (a) Circular Geometry 3D plot (b) Circular Geometry 2D plot (c) Circular Geometry, Indoor Experiment with RevoLabs System

Table 5.2: Source locations for Circular geometry experiments

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$	$(2r \cos 180^\circ - r, 2r \sin 180^\circ, 0)$	$(2r \cos 60^\circ - r, 2r \sin 60^\circ, 0.5)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$	$(0, 2.6, 0.5)$	$(0, 2.6, 1)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic1 & 2, outside the array at height of 1m	In-between mic1 & 2, inside the array at height of 0m	Along the line (outside array) passing through mic1 and center of the array at height 0.5m	Along the line (inside array) passing through mic1 and center of the array at height 0m	In-between mic0 and 1 on the circumference at height of 0.5m	Along the line (outside array) passing through mic1 and center of the array at height of 1m

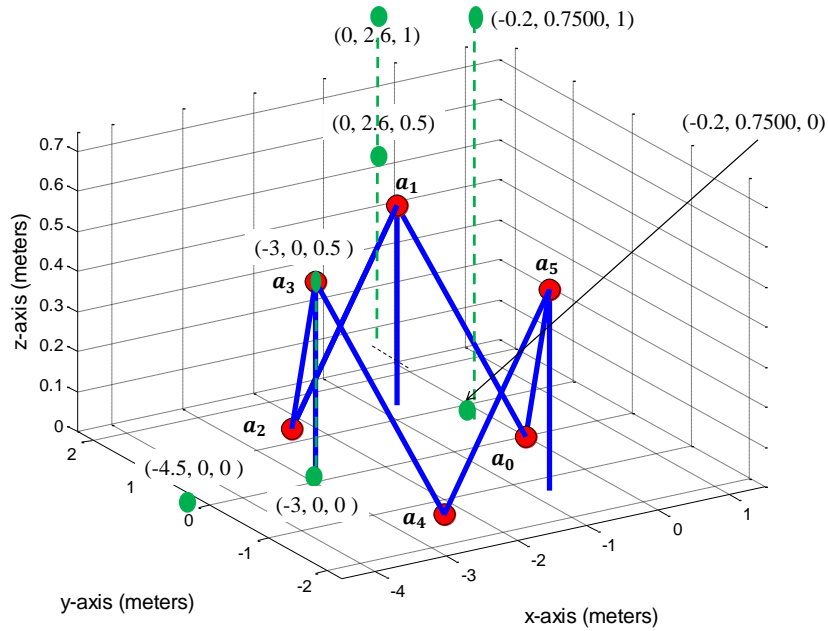


Figure 5.7: Selection of source locations of Circular geometry for experimentation

Rhombus Geometry

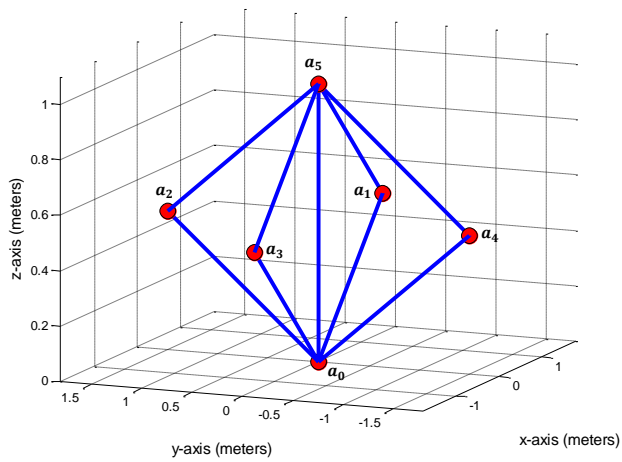
This geometry is a modified version of the Pyramid geometry. We added an extra sensor to the bottom of the array and elevated them by 0.5m while mic0 remained at the origin. The array converts to a rhombus after this modification as shown

in Fig. 5.8. The sensor (microphone) locations are:

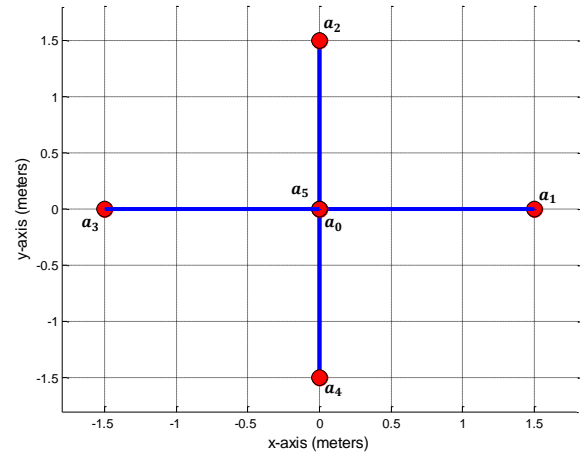
$$\begin{aligned} a_0 &= (0, 0, 0), & a_1 &= (1.5, 0, 0.5), & a_2 &= (0, 1.5, 0.5), \\ a_3 &= (-1.5, 0, 0.5), & a_4 &= (0, -1.5, 0.5), & a_5 &= (0, 0, 1) \end{aligned}$$

Note that in the practical implementation (Fig 5.8c) the array ground was considered at 39cm height from the real ground to avoid reflections from the ground and other effects of the ground as was discussed in Section 2.3.3. This elevation of 39cm for Rhombus geometry is half of the elevation we chose for Pyramid and Circular geometries because of the limited height of the microphone stands. Here the base of the geometry is a_0 while the highest sensor a_5 is at elevation of 1m from the base of the geometry. As we will see in Chapter 6 and 7 this reduced elevation of the Rhombus geometry has significant effect on the accuracy of the geometry.

To select the source positions for the experimentation we selected locations which cover the important areas of the geometry. Fig. 5.9 shows the source locations (shown as green dots) that we have selected for our experimentation. These points will be used in all our experiments both in indoor and outdoor scenarios. Table 5.3 shows the source locations with their description and mathematical representation.



(a)



(b)



(c)

Figure 5.8: TDOA Rhombus Geometry with RevoLabs System. (a) Rhombus Geometry 3D plot (b) Rhombus Geometry 2D plot (c) Rhombus Geometry, Indoor Experiment with RevoLabs System

Table 5.3: Source locations for Rhombus geometry experiments

Source Location	$(2r, 0, h)$	$(2r, 0, 2h)$	$(2r \cos 45^\circ, 2r \sin 45^\circ, 2h)$	$(2r \cos 75^\circ, 2r \sin 75^\circ, h)$	$(2r \cos 88^\circ, 2r \sin 88^\circ, h)$
Value	$(3, 0, 0.5)$	$(3, 0, 1)$	$(2.12, 2.12, 1.00)$	$(0.77, 2.89, 0.5)$	$(0.10, 2.99, 0.5)$
Description	On the line passing through mic1 and center of the array at height 0.5m	On the line passing through mic1 and center of the array at height 1m	In-between mic1 and 2 outside the circumference on which mics ly.	At angle of 75° between mic1 & 2.	At angle of 88° between mic1 & 2.

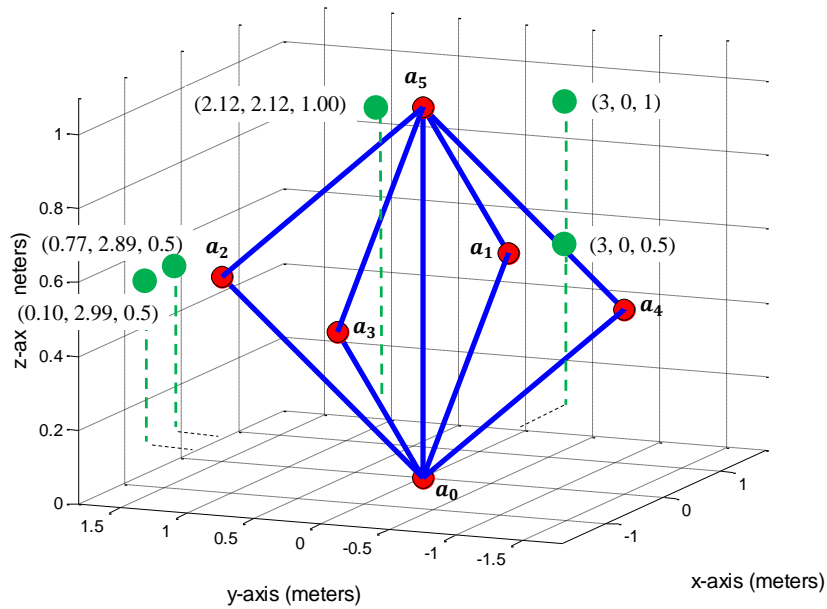


Figure 5.9: Selection of source locations of Rhombus geometry for experimentation

5.3 Hardware Systems

As discussed in Chapter 4 we tried to implement our acoustic source localization system on WSN but because of the issues encountered and the HW limitations we could not continue with that implementations. To verify the OC algorithm and compare its performance with CC TDE method we chose two different wireless

microphone systems for signal acquisition in the indoor and outdoor environments. Both of these systems are for commercial use like in meeting rooms, music concerts etc. These hardware systems are discussed next.

5.3.1 RevoLabs[®] Wireless HD Microphone System

This is an 8-channel wireless microphone system (shown in Fig. 5.10) with advanced features. Its specifications are [39]:

1. Audio Bandwidth: 50 Hz - 20 kHz
2. Radio Frequency: 1.88 - 1.90 GHz
3. Ethernet - RJ45
4. Rechargeable batteries powered microphones, battery life = 8 hours.
5. Range of Operation: 300 ft. = 91m.
6. Impervious to RF interference due to RF Armor technology.
7. Omni-directional wireless microphones

The basestation provides the output through a Mini Phoenix Connector for each channel (mic).

5.3.2 VocoPro[®] UHF8800 Wireless Microphone System

UHF-8800 [40] is an 8-channel wireless microphone system that is produced by VocoPro[®]. The system consists of a receiver basestation and 8 microphones as

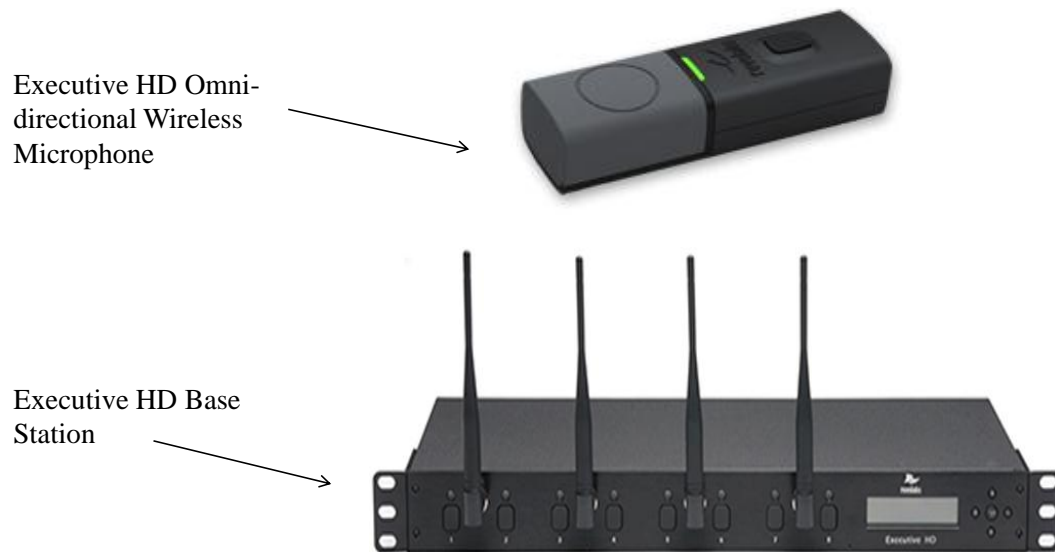


Figure 5.10: RevoLabs HD Wireless Microphone System

shown in Fig. 5.11. Its features are:

1. UHF Band Operation
2. 8 XLR Microphone Outputs
3. Frequency bands of the 8 channels: M = 656.825 MHz, N = 685.96 MHz, O = 694.11 MHz, P = 629.40 MHz, Q = 676.74 MHz, R = 614.15 MHz, S = 619.12 MHz.
4. Receiver's Frequency Response: 40 Hz - 16 kHz.
5. Receiver's Max. Output Level: Balanced: 0 - 400mV, Unbalanced: 0 - 200mV
6. Receiver's spurious rejection: 75 dB typical

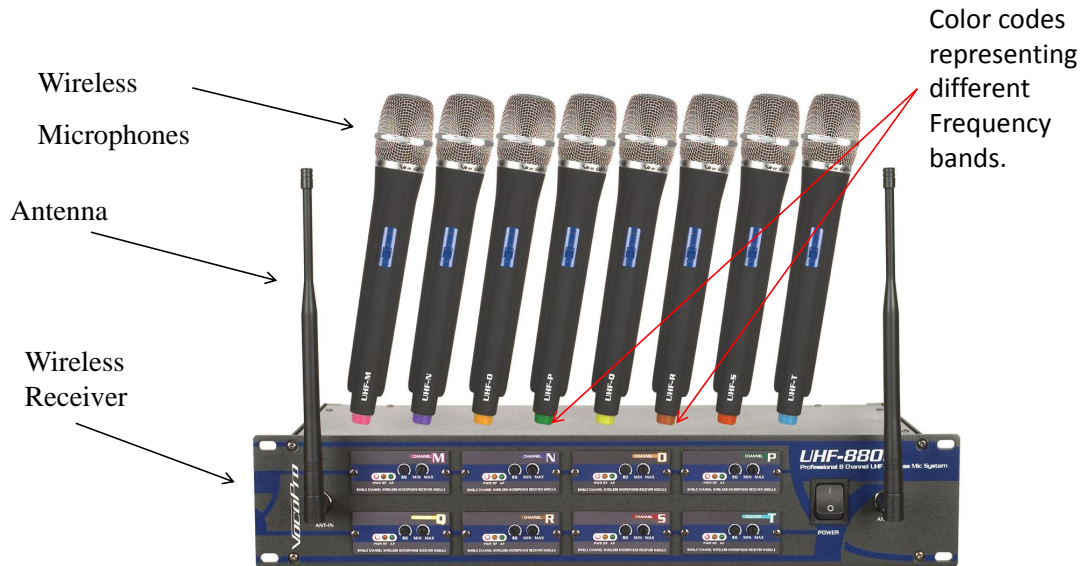


Figure 5.11: VocoPro UHF8800 Wireless Microphone System

7. Transmitter Power: 30mV
8. Transmitter Spurious Emission: <40dB(with carrier)
9. Transmitter Battery Voltage: Two AA 1.5V, Battery Life: 12 hours

The microphones of this system are directional and are specifically designed for vocals thus their sensitivity is quite low than RevoLabs microphones. This feature has special effects on the experimentation and the results as we will discuss in Chapter 6 and 7 in detail.

5.3.3 Data Acquisition Setup

The VocoPro system provides output through XLR balanced audio jacks while the RevoLabs system provides output through Mini Phoenix connectors. For both

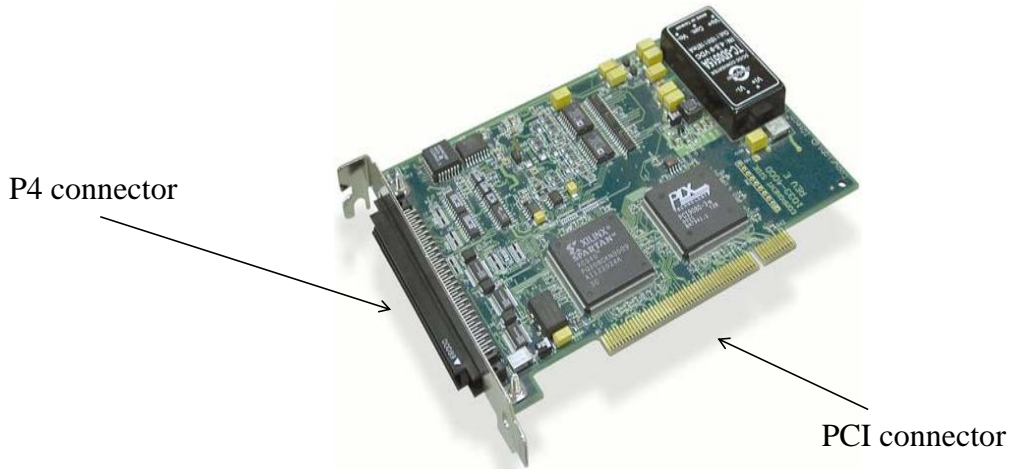


Figure 5.12: MCC's DaqBoard/2000 Data Acquisition PCI Card

of the systems a data acquisition (DAQ) device is needed that converts analog signals to digital and presents the data in an acceptable form to the processing environment like MATLAB. For this purpose we use two different DAQ systems: i) DaqBoard/2000 along with DBK202 [41] and ii) USB1608FS USB based DAQ card [42]. Both of the boards are from Measurement Computing Corporation (MCC[®]).

The DaqBoard/2000 (shown in Fig. 5.12) is fully supported by MATLAB and we can acquire data directly from the board. The maximum sampling frequency of the board is 200 kHz accumulative for all channels. A maximum of 16 single-ended analog/digital channels can be sampled. Additional hardware can be attached to the P4 connector through the CA-195 cable for ease of signal acquisition, for example, we are using DBK202 [41] adapter (shown in Fig. 5.13).

The DaqBoard/2000 can be attached to a PC having a PCI expansion card

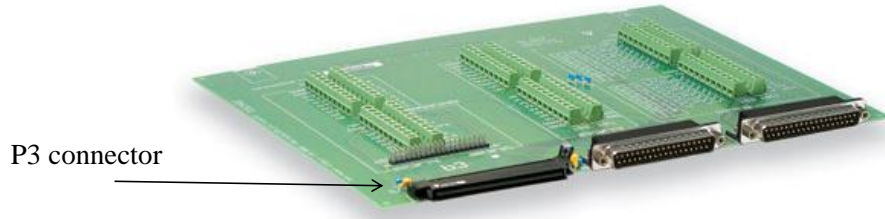


Figure 5.13: MCC's DBK202 Expansion Card

only. Thus it cannot be connected to a laptop limiting its mobility. Furthermore, we tested the DabBoard/2000 for data acquisition and we found that there is heavy coupling between adjacent channels. Fig. 5.14 shows such a test in which we connected the output of a microphone of RevoLabs system to channel 0 of the DaqBoard. All other channels of the DaqBoard were open and no signals were connected to them. Fig. 5.14 clearly shows that there is coupling between channel 0 and 1. Besides that the DaqBoard adds an unwanted bias to the signals deteriorating the shape of the signal which affects the results of the TDE algorithms.

We decided to use USB1608FS DAQ shown in Fig. 5.15 from the same company. We tested it and there were no issues found which were found in the case of DaqBoard. Furthermore, the USB1608FS DAQ is a USB device and small in size which can easily be connected to a laptop and thus very portable, especially helpful in outdoor experiments. USB1608FS can support up to 16 single-ended analog channels with 100kHz maximum sampling rate for all channels accumulative. Since we are using a maximum of six microphones in our experiments we can easily sample each mic at 16 kHz sampling rate which is enough for our

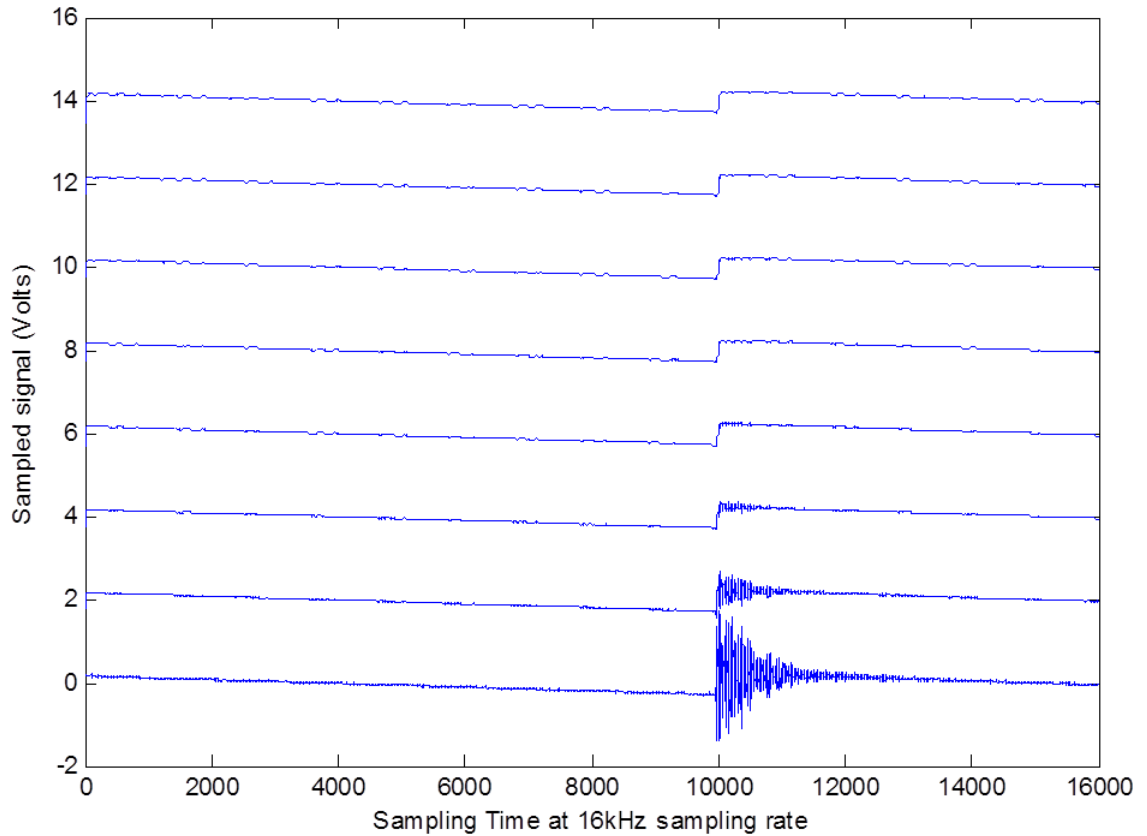


Figure 5.14: Signal acquired from DaqBoard/2000 with channel 0 connected to a microphone and channel 1-7 unconnected to any signal but are open.

application.



Figure 5.15: MCC's USB1608FS USB Data Acquisition Board

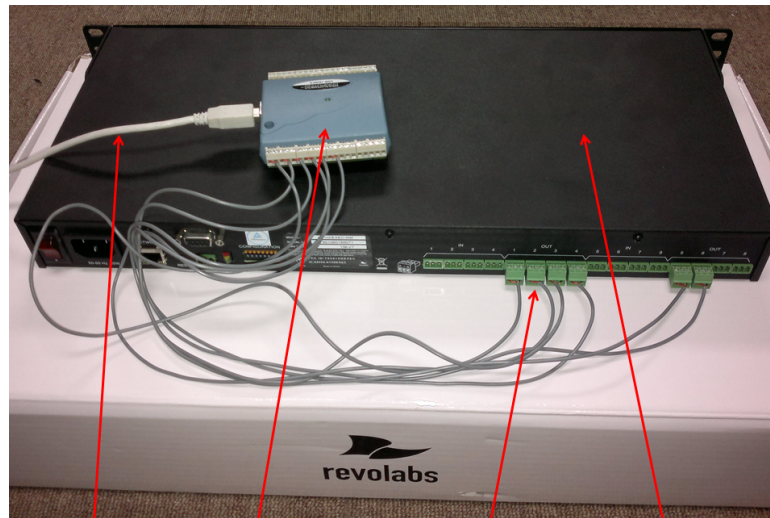
The next step was to connect the outputs of RevoLabs and VocoPro bases-

tations to USB1608FS properly. As was mentioned earlier, the output of the RevoLabs system is provided through Mini Phoenix connectors. We connected the signal and ground wire of the Mini Phoenix to USB1608FS directly as shown in Fig. 5.16a while for the VocoPro system we have to use two additional connectors: first an XLR to standard Audio Jack and then Audio Jack to 3.5mm 5-pin Stereo Jack (SJ-3535NGS) converter produced by Digikey. Fig. 5.16b shows the connection of UHF8800 connections to USB1608FS. After setting up successfully the hardware system and TDOA geometries in MATLAB we headed towards experimentations and the detailed results are given in Chapter 6 and 7.

5.3.4 Computing Platform

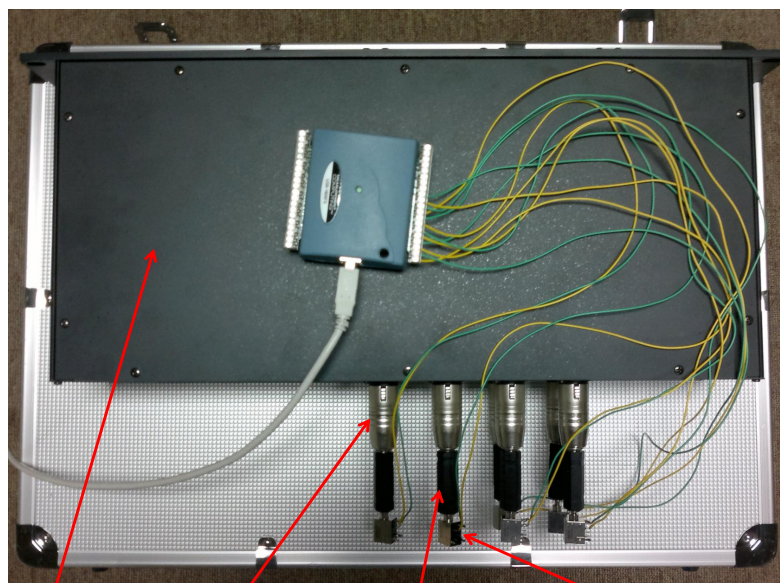
Because of the extensive amount of experiments and their evaluation we needed to use several computers to process the data using the OC algorithm. There were two kinds of computers:

1. Dell[®]'s Precision T1500 with the following specifications:
 - Processor: Intel[®] Core[™] i7
 - Memory (RAM): 8 GB
 - Operating System: Microsoft[®] Windows 7
2. Dell[®]'s Precision T1500 with the following specifications:
 - Processor: Intel[®] Core[™] i5
 - Memory (RAM): 4 GB



USB cable to the Laptop USB1608FS Mini Phoenix Connectors RevoLabs Basestation

(a)



VocoPro Basestation XLR Audio Jack Standard Audio Jack Converter Digikey 3.5mm 5-pin Stereo Jack (SJ-3535NGS)

(b)

Figure 5.16: USB1608FS connection to: a) RevoLabs system (b) VocoPro system

- Operating System: Microsoft[®] Windows 7

We will refer to the first computer as Fast Computer and to the second one as Slow Computer in our results in the rest of this thesis.

5.4 Conclusions

In this chapter we introduced the OC algorithm and its detailed steps. All the experimental scenarios were described in detail and the TDOA geometries that we will be using in our indoor and outdoor experiments were presented. Moreover, the target locations we have chosen for our experimentation were also explained. The data acquisition setup including the wireless microphone systems, data acquisition boards and the processing platforms were illustrated and their features were highlighted.

CHAPTER 6

INDOOR EXPERIMENTS AND RESULTS

In this chapter we discuss and present the detailed analysis and results of the indoor experimentation conducted. There are several scenarios for indoor experimentation. We draw a part of Fig. 5.3 again here in Fig. 6.1 to show the indoor experimentation scenarios investigated in this chapter.

Before discussing the experimental results, it is better to first simulate the TDOA geometries for the source locations specified in Section 5.2.1 in order to compare them with experimental results. We provide the parametric analysis of the OC algorithm in Section 6.1. Section 6.2 discusses the indoor experiments and results using RevoLabs wireless microphone system while Section 6.3 discusses the indoor experiments and results using VocoPro wireless microphone system. Section 6.4 concludes this chapter.

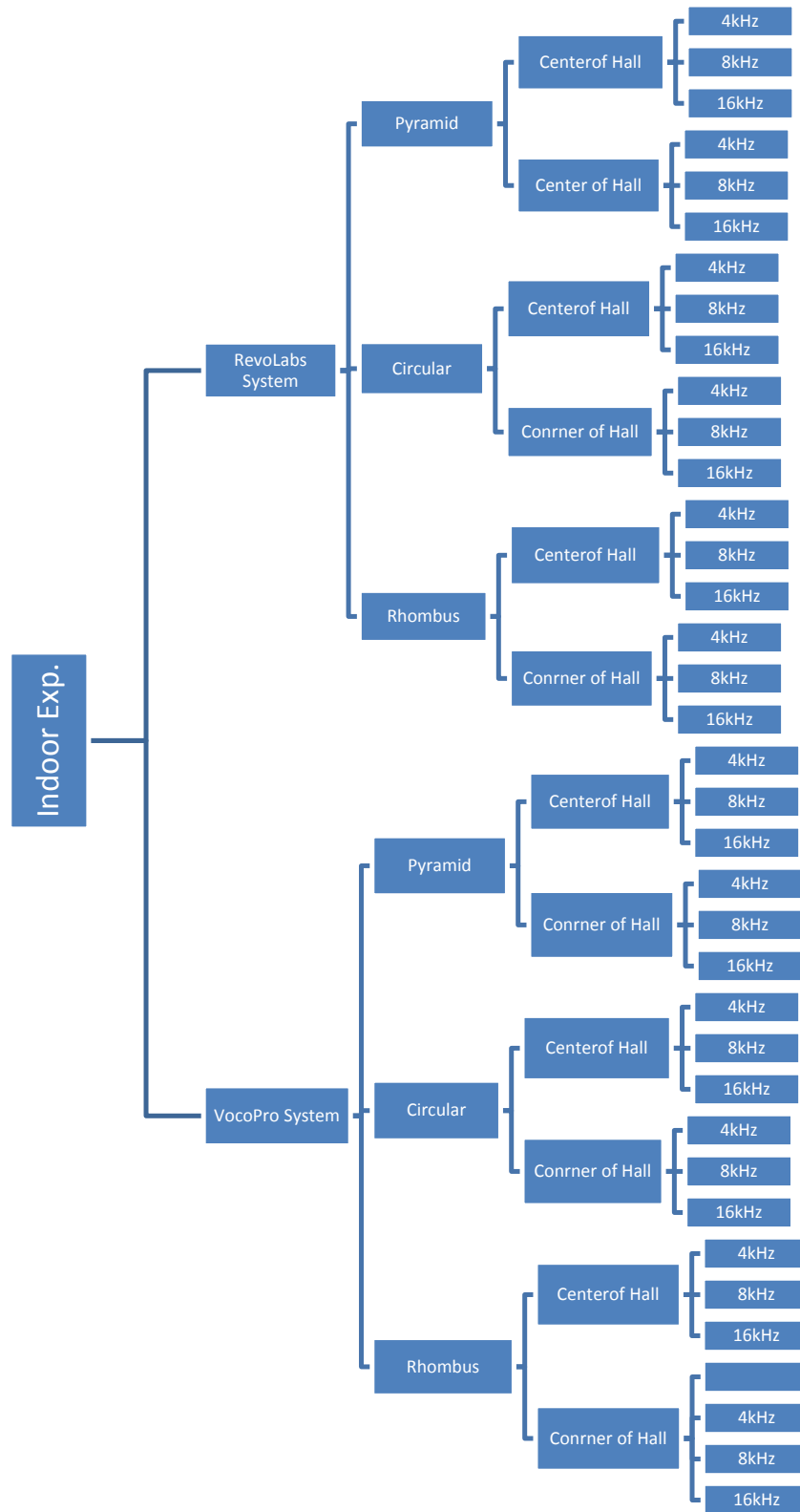


Figure 6.1: Indoor experiments scenarios

Note that in Fig. 6.1 we show the 16kHz case but we were not able to run the OC algorithm on the available PCs due to the huge amount of memory required for OC for the case of 16kHz sampling rate. The signals were acquired at 16kHz sampling rate only for the Dictionary matrix (Ψ) development.

6.1 Parametric Analysis of the OC Algorithm

In this section we want to analyze the variation effects of different parameters on the results of the OC algorithm. The parameters investigated are:

1. The Dictionary matrix Ψ
2. The index of CIR used to calculate the TDOA
3. The Probability of nonzero values (p) in the CIR
4. Signal Variance (σ_{imp})
5. Noise Variance (N_0)

We chose the Pyramid geometry discussed in Section 5.2.1 for this analysis (we chose this geometry randomly as we could choose any other). Pyramid geometry has five sensors (numbered from 0 - 4). The case of indoor experiment in the center of the hall was taken and the sampling rate for signal capture was 8 kHz. We used RevoLabs HD Wireless Microphone System for acoustic signal acquisition with Measurement Computing (MCC) USB1608FS data acquisition card.

Now we will discuss these parametric analysis one by one in the following pages.

6.1.1 Effect of Ψ on the results

The Dictionary matrix Ψ plays very important role in the accuracy of the results of the OC algorithm. In this section we will show that the algorithm is very sensitive to the Ψ which limits the practicality of the algorithm. As was discussed in Section 5.1.1 the Ψ columns consist of delayed versions of a signal. We captured five different signals at different locations and scenarios at 16 kHz and developed Ψ from them. Then using the Pyramid geometry and the experimental setup described in chapter 5 we conducted several experiments to see the effect of Ψ on the results. It is to be noted that we captured the signal at 16 kHz for Dictionary building but when we carried out the experiment for localization we used 8 kHz thus we downsample the Ψ to 8 kHz for successful operation of the algorithm. The results are discussed next.

Ψ Based on a Signal Captured in a Big Hall (Ψ_0)

We used the Ψ matrix based on a signal captured in the middle of a big hall (whose dimensions were larger than $10 \times 15m^2$) thus there were almost no reverberations. This Ψ was adapted from [6]. Table 6.1 gives details of the results when we run the algorithm using this Dictionary. The top row of Table 6.1 represents the source locations we discussed in Section 5.2.1. TDE_{OC} is Time Delay Estimate (TDE) using Orthogonal Clustering (OC) algorithm and X_{OC} is the source location estimate using OC. MSE stands for Mean Square Error (ϵ) which

is calculated using

$$\epsilon_{OC} = \sqrt{(x_s - x_s^{oc})^2 + (y_s - y_s^{oc})^2 + (z_s - z_s^{oc})^2} \quad (6.1)$$

and τ_{ij} is the time delay between mic i and j .

Table 6.1: Results for 8 kHz case of Triangular geometry when array was placed in the center of the hall using Ψ_0 . All the measurements are in meters

Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	(2.5cos60°, 2.5sin60°, 0) = (1.25, 2.16, 0)	(2cos60°, 2sin60°, 0) = (1, 1.73, 0)	(2.5, 0, 0)
Description	In-between mic1 and 2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between mic1 & 2, outside the array	In-between mic1 & 2 on the circle inscribing the array.	Along the line (outside array) passing through mic1 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE _{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}$)	(0.0019, 0.0006, 0.0040, -0.0009)	(0.0031, 0.0018, 0.0046, -0.0008)	(-0.0010, -0.0005, 0.0041, -0.0014)	(0.0047, 0.0011, 0.0071, 0.0024)	(-0.0018, 0.0030, 0.0020, 0.0020)
X_{oc} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.1990, 0.9873, 1.2052)	(0.0562, 0.6572, 0.7559)	(1.9129, 2.7945, 3.8052)	(0.0395, 1.0509, -0.5213)	(2.2769, -0.5339, -4.2958)
MSE for OC	0.7763	0.7033	3.9143	1.2867	4.3346
Run Time (Sec)	38.345	35.4579	44.5371	66.0748	39.8441

Ψ Based on a Signal Captured in a relatively Small Hall with Source being Near a Wall (Ψ_1)

The signal that was used to build this Dictionary was captured from an unloaded gunshot at 16 kHz; the sensor was 3m away from the wall. The source was 160cm away from the sensor and in the same plane in which the sensor was lying. Fig. 6.2 shows the plot of the captured signal. Table 6.2 displays the results when the Ψ_1 was used.

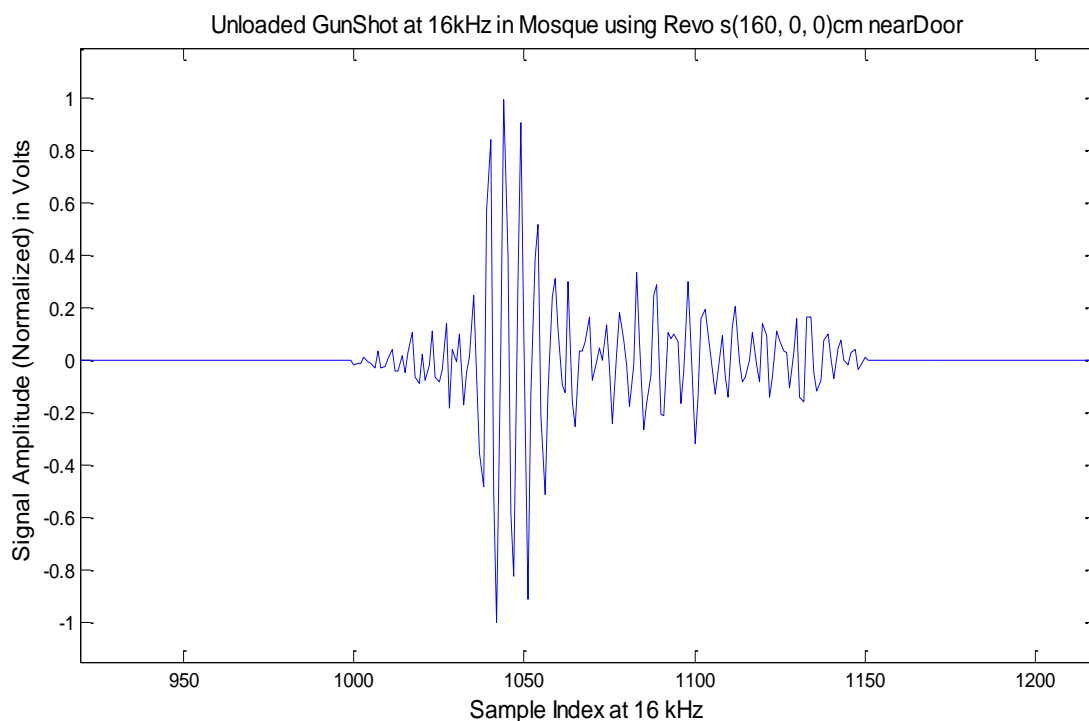


Figure 6.2: Indoor Unloaded Gunshot at 16 kHz near the wall captured using RevoLabs System

Table 6.2: Results for the case of Ψ_1

Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	(2.5cos60°, 2.5sin60°, 0) = (1.25, 2.16, 0)	(2cos60°, 2sin60°, 0) = (1, 1.73, 0)	(2.5, 0, 0)
Description	In-between mic1 and 2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between mic1 & 2, outside the array	In-between mic1 & 2 on the circle inscribing the array.	Along the line (outside array) passing through mic1 and center of the array
Exact TDs ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_4^{oc}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE _{oc} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_4^{oc}$)	(0.0016, 0.0010, 0.0054, -0.0011)	(0.0006, -0.0010, 0.0018, -0.0049)	(-0.0030, -0.0027, 0.0014, 0.0001)	(-0.0009, -0.0001, 0.0041, -0.0027)	(-0.0015, 0.0021, 0.0027, -0.0008)
X _{oc} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.5053, 1.1157, 1.0630)	(-0.3164, 3.3499, 26.7301)	(-0.9516, -1.4374, 0.5545)	(1.6739, 2.2354, 5.8290)	(2.0843, 0.3273, 1.9111)
MSE for OC	0.6160	25.9492	4.2539	5.8895	1.9830
OC Time (Sec)	57.2618	48.4149	88.6293	49.1725	42.1681

Ψ Based on a Signal Captured in a relatively Small Hall with source being in the middle of the hall and near the sensor (Ψ_2)

The signal shown in Fig. 6.3 for this Dictionary was captured in the same hall and same equipment was used as in the previous subsection but this time the source was in the middle of the hall. The source was at 10cm height just right on top of the sensor. Please note the effect of the source being near the sensor. There are more peaks in this case than the case of Ψ_1 . Table 6.3 shows the results when Ψ_2 was used with the OC algorithm.

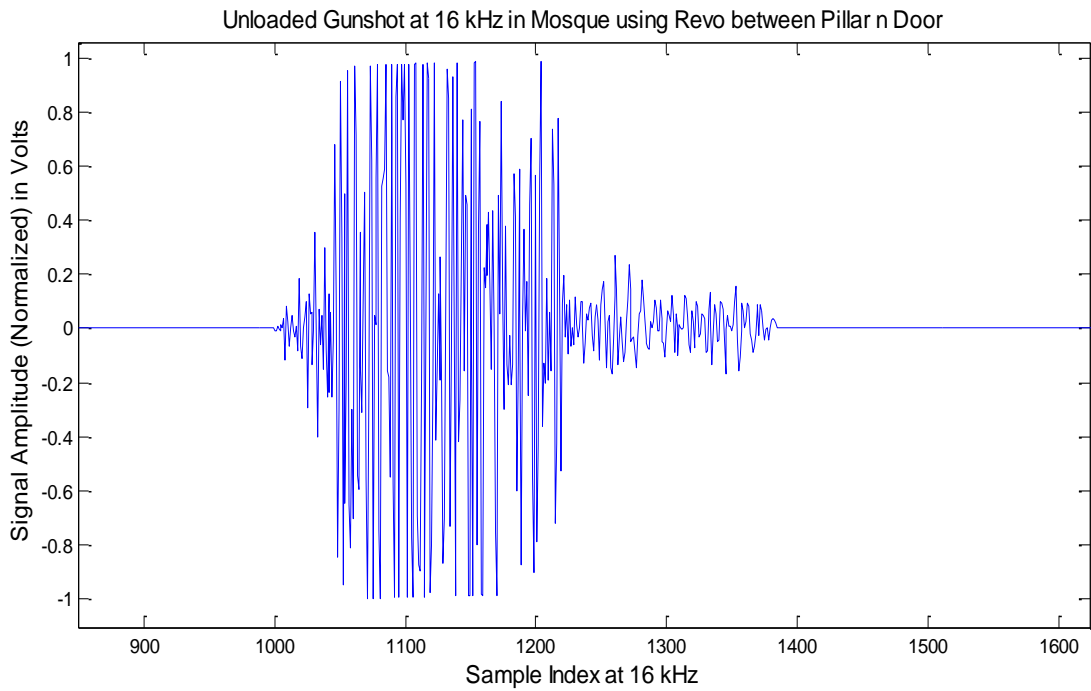


Figure 6.3: Indoor Unloaded Gunshot at 16 kHz using RevoLabs System when Source was in the middle of the hall and near the sensor

Table 6.3: Results for the case of Ψ_2

Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	(2.5cos60°, 2.5sin60°, 0) = (1.25, 2.16, 0)	(2cos60°, 2sin60°, 0) = (1, 1.73, 0)	(2.5, 0, 0)
Description	In-between mic1 and 2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between mic1 & 2, outside the array	In-between mic1 & 2 on the circle inscribing the array.	Along the line (outside array) passing through mic1 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE _{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}$)	(0.0049, 0, 0.0073, 0.0027)	(0.0076, 0.0027, 0.0106, 0.0049)	(0.0056, 0.0027, 0.0094, 0.0039)	(-0.0005, 0.0001, 0.0031, 0.0018)	(-0.0015, 0.0015, 0.0025, 0.0032)
X _{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.0061, 1.1512, -0.7474)	(0.1045, 1.2895, -0.1541)	(0.2989, 1.1953, -0.4373)	(1.4855, 1.8696, -4.6764)	(2.5486, 0.6878, -10.1994)
MSE for OC	1.3760	1.7337	1.4236	4.7036	10.2227
OC Time (Sec)	61.1333	58.7058	53.7054	48.9618	205.5077

Ψ Based on a Signal Captured Outdoor when Source was Far from the Sensor (Ψ_3)

The signal that was used to build the Ψ_3 was captured from an unloaded gunshot at 16 kHz; the sensor was placed in an outdoor environment. The source was 2m away from the sensor and in the same plane in which the sensor was lying. Fig. 6.4 shows the plot of the captured signal. Table 6.4 displays the results when Ψ_3 was used on the data for Triangular geometry for 8kHz case and array being in the center of the hall.

Table 6.4: Results for the case of Ψ_3

Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	(2.5cos60°, 2.5sin60°, 0) = (1.25, 2.16, 0)	(2cos60°, 2sin60°, 0) = (1, 1.73, 0)	(2.5, 0, 0)
Description	In-between mic1 and 2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between mic1 & 2, outside the array	In-between mic1 & 2 on the circle inscribing the array.	Along the line (outside array) passing through mic1 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_4$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE _{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_4^{oc}$)	(0.0040, 0.0016, 0.0057, 0.0003)	(0.0026, 0.0004, 0.0050, -0.0013)	(-0.0004, -0.0006, 0.0040, 0.0022)	(0.0004, -0.0003, 0.0044, 0.0013)	(-0.0027, 0.0022, 0.0032, 0.0010)
X _{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.0081, 0.8229, 0.2631)	(0.1202, 1.1071, 1.1349)	(1.3112, 2.5055, -5.3011)	(0.7925, 1.7668, -1.6489)	(3.1113, 0.5809, -2.0737)
MSE for OC	0.5477	1.1218	5.3127	1.6623	2.2386
OC Time (Sec)	47.5013	43.8546	49.3129	60.1322	40.7117

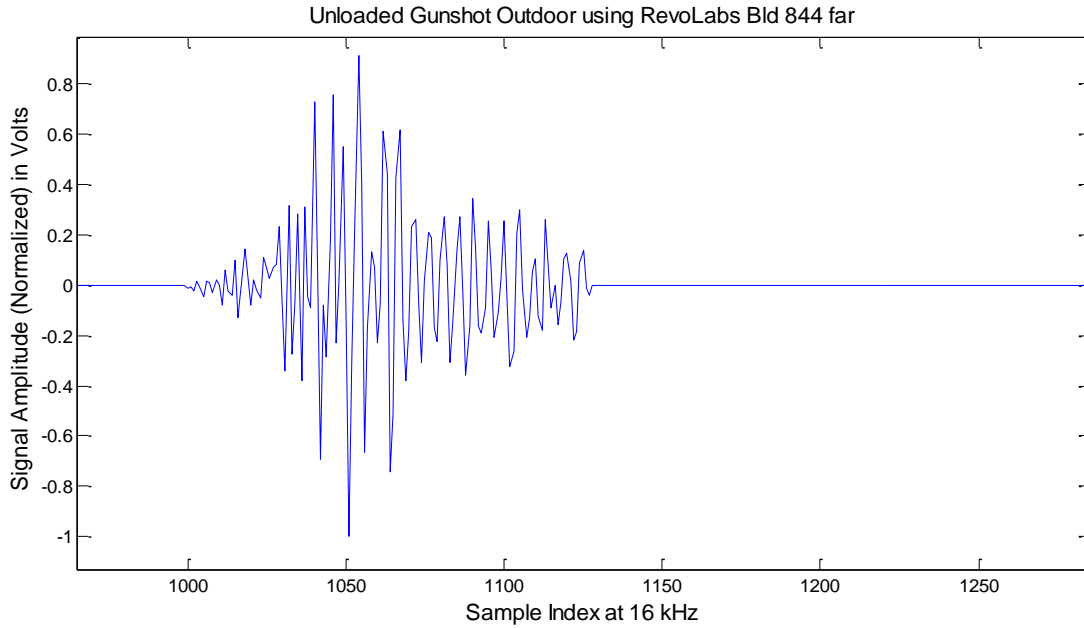


Figure 6.4: Outdoor Unloaded Gunshot at 16 kHz using RevoLabs System when Source was Far from the Sensor

Ψ Based on a Signal Captured Outdoor when Source was Near the Sensor (Ψ_4)

Ψ_4 was built using a signal captured outdoor using same environment and equipment as in the previous section but this time the source was 10cm away from the sensor and in the same plane in which the sensor was lying. Fig. 6.5 shows the plot of the captured signal and Table 6.5 shows the results when we used Ψ_4 with the OC algorithm.

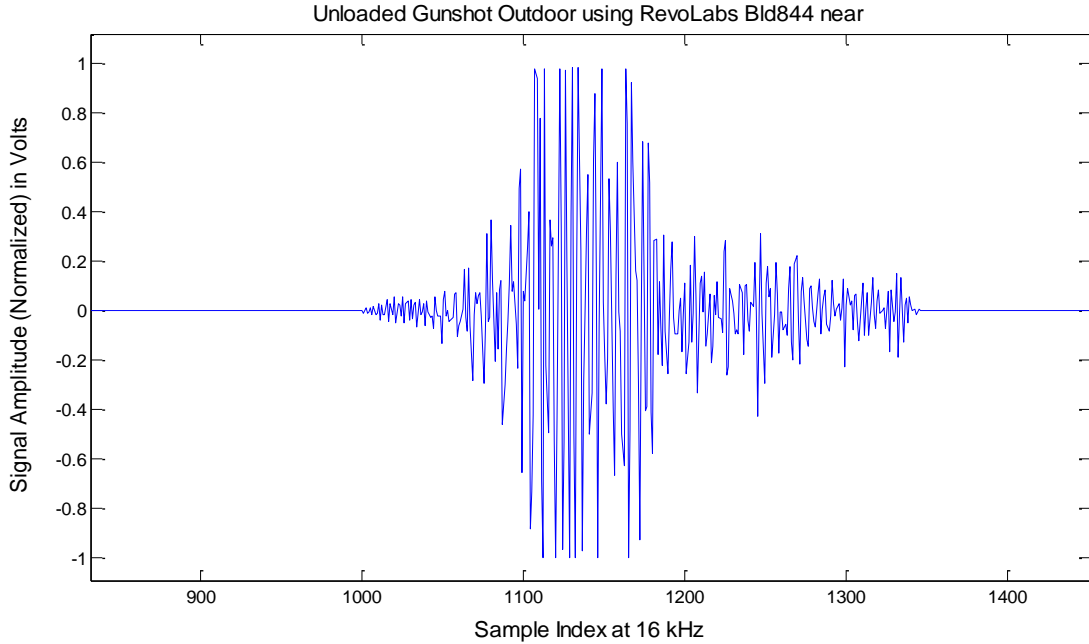


Figure 6.5: Outdoor Unloaded Gunshot at 16 kHz using RevoLabs System when Source was near to the Sensor

Table 6.5: Results for the case of Ψ_4

Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	(2.5cos60°, 2.5sin60°, 0) = (1.25, 2.16, 0)	(2cos60°, 2sin60°, 0) = (1, 1.73, 0)	(2.5, 0, 0)
Description	In-between mic1 and 2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between mic1 & 2, outside the array	In-between mic1 & 2 on the circle inscribing the array.	Along the line (outside array) passing through mic1 and center of the array
Exact TDs ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_4^{oc}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{oc} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_4^{oc}$)	(0.0059, 0.0011, 0.0088, -0.0003)	(0.0025, 0.0043, 0.0051, -0.0010)	(0.0008, -0.0019, 0.0027, -0.0011)	(0.0043, 0.0037, 0.0095, 0.0047)	(-0.0035, 0.0015, 0.0022, -0.0001)
X_{oc} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.0945, 1.1874, 0.3441)	(0.4955, 0.2023, 0.6396)	(-0.2456, 4.4444, 5.3807)	(0.6083, 1.1694, -1.0467)	(34.7310, 4.2639, 3.7402)
MSE for OC	0.5404	0.6453	6.0338	1.2503	32.7262
OC Time (Sec)	60.4040	52.9025	55.8619	48.4347	62.7220

Selection of Preferred Ψ

Looking at the results in Table 6.1 to 6.5 we observe that other than the two specific source locations (1.25, 2.16,0) and (2.5,0,0) the Dictionary Ψ_4 has the least error for OC compared to other dictionaries. The least runtime on averages is observed for Ψ_0 but its MSE is more than the Ψ_4 in for three out of five source

locations (60%). Thus we will use this Dictionary onward in our experiments.

6.1.2 The index of CIR used to calculate the TDOA

We carried out an experiment to decide how to calculate correct TDOA for OC algorithm. The values of different parameters of the algorithm were: $N_0 = 1.5 \times 10^{-4}$, $\sigma_{imp} = 1$, $p = 5$ which were chosen based on the experimental readings. We chose two different conditions for index selection for the purpose of TDOA calculation:

1. Using the index of first nonzero value of CIR to calculate TDOA
2. Using the index of maximum value of CIR to calculate TDOA

Table 6.6 shows the results when we used both the aforementioned conditions along with the comments in the last column. For three out of five source locations the choice of index of the maximum amplitude produces more correct results compared to the choice of index of first nonzero value of the CIR.

Table 6.6: Index choice for TDOA calculation

Source Location	Exact TDOA	TDOA based on Maximum Amplitude	TDOA based on first nonzero value	Comments
(0.5, 0.866, 0.5)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.005, 0.0011, 0.0088, -0.0003)	(0.0022, 0.0011, 0.0056, -0.00025)	First nonzero based TDOA is more correct compared to the Max. Amp case
(0, 0, 1)	(0.0036, 0.0036, 0.0036, -0.0021)	(0.0025, 0.0043, 0.0051, -0.0010)	(0.0027, -0.0034, 0.0053, -0.0010)	Max. Amp based TDOA is more correct compared to the First Nonzero case
(1.25, 2.16, 0)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0.0008, -0.0019, 0.0027, -0.0011)	(0.0075, 0.0047, 0.0094, 0.0056)	Max. Amp based TDOA is more correct compared to the First Nonzero case
(1, 1.73, 0)	(0, 0.0000, 0.0059, 0.0004)	(0.0043, 0.0037, 0.0095, 0.0047)	(0.0032, 0.0029, 0.0044, -0.0035)	Max. Amp based TDOA is more correct compared to the First Nonzero case
(2.5, 0, 0)	(-0.0059, 0.0041, 0.0041, 0.0003)	(-0.0035, 0.0015, 0.0022, -0.0001)	(-0.0043, 0.0016, 0.0025, 0)	First nonzero based TDOA is more correct compared to the Max. Amp case

6.1.3 Effect of Probability of Impulses (p)

We conducted another experiment which deals with the probability of impulses in a CIR signal (p) which is equal to the expected non-zero values in the CIR divided by the length of the CIR. We used index of maximum value of CIR for TDOA calculation. Other parameters were kept as follows:

$$\sigma_{imp} = 0.5, N_0 = 1.5 \times 10^{-4}, c = 340m/s$$

where $c = 340m/s$ is the speed of sound, N_0 is the noise variance and σ_{imp} is the variance of the source signal. We chose the values of p in the range of 3 - 15 as it covers the expected number of reflections for both of the indoor and outdoor experimentation. The speed of sound (c), in our case, is constant because during our experimentation the temperature range was 15 - 25 °C which corresponds to the speed of sound in the range 340.31 – 346.18m/s. Using this range of the speed of sound the time delay for 1m distance is almost constant and is equal to 0.0029 seconds. So we can safely choose $c = 340m/s$. Table 6.7 shows the results which tell us that changing the value of p has no effect on the results at all. Instead it increases the run time for the algorithm significantly. In other words, the algorithm does not care about what value of p we provide to it, it looks for the presence of the dictionary signal (of which Ψ is made of) or its reflected version in the signal of a sensor.

Table 6.7: Effect of change in p on the results of the algorithm

p	Source Location	Description	Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	MSE for CC	MSE for OC	CC Time (Sec)	OC Time (Sec)
3	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0354	41.0290
5	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0351	64.6451
8	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0813	2158.5=35minutes 58 seconds
10	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.2134	899.1354=14minutes 8 seconds
15	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.1903	7661.2=2hrs 7minutes 41 seconds

6.1.4 Signal Variance (σ_{imp})

Now we will investigate the effects of variation in signal variance (σ_{imp}) on the results of the algorithm using the same signals and Ψ matrix as in Section 6.1.1.

We used index of maximum absolute value of CIR for TDOA calculation. Other parameters were kept as follows:

$$p = 5, N_0 = 1.5 \times 10^{-4}, c = 340m/s \quad (6.2)$$

We kept the range of signal variance from 0.3 to 2.0 volts because during experimentation we found out that the signal peak amplitude was in this range.

Table 6.8 shows the results for different values of σ_{imp} . As can be seen from Table 6.8 changing the value of σ_{imp} has no effect on the results at all and we can choose any value we want. In fact, the results depend on the source signal, the Dictionary matrix and sampling rate. We will choose $\sigma_{imp} = 1$ as it is favorable for normalized signals.

Table 6.8: Effect of change in σ_{imp} on the results of the algorithm

σ_{imp}	Source Location	Description	Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	TDE_{cc} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	TDE_{oc} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	X_{cc} ($x_1^{cc}, y_1^{cc}, z_1^{cc}$)	X_{oc} ($x_1^{oc}, y_1^{oc}, z_1^{oc}$)	MSE for CC	MSE for OC	CC Time (Sec)	OC Time (Sec)
0.3	(0.5, 0.866, 0.5)	In-between mic 1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0644	63.3623
0.8	(0.5, 0.866, 0.5)	In-between mic 1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0606	59.1333
1.0	(0.5, 0.866, 0.5)	In-between mic 1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0458	60.0819
1.5	(0.5, 0.866, 0.5)	In-between mic 1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0351	58.2630
2.0	(0.5, 0.866, 0.5)	In-between mic 1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0359	58.6135

6.1.5 Noise Variance (N_0)

Looking at the results of previous subsections we can safely choose $p = 5$, $\sigma_{imp} = 1$, $c = 340m/s$ without causing any effect to the results of the algorithm. We used index of first nonzero value of CIR for TDOA calculation. Now we will investigate the effect of variations in the variance of noise on the results of the algorithm.

After analyzing all the signals we captured both indoor and outdoor we found out that the noise level was in the range of -0.04 to 0.04 volts (these values represent the maximum and minimum noise level) which when normalized w.r.t. the peak of the signal reduces to -0.003 to 0.003 volts. Now we will select a range of 0.005 - 0.09 volts for signal noise level to see its effect on the output of the algorithm.

We will use the same signals and dictionary matrix which we used in the previous subsections. Table 6.9 shows the results when we varied the value of N_0 .

Table 6.9: Effect of change in N_0 on the results of the algorithm

N_0	Source Location	Description	Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	MSE for CC	MSE for OC	CC Time (Sec)	OC Time (Sec)
0.005	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0024, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(0.3633, 1.0631, 0.4935)	0.3448	0.2400	0.0355	80.6391
0.01	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0059, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(-0.4631, 0.7422, 0.4100)	0.3448	0.9752	0.0861	87.9956
0.03	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0059, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(-0.4631, 0.7422, 0.4100)	0.3448	0.9752	0.0357	144.4692
0.05	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0059, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(-0.4631, 0.7422, 0.4100)	0.3448	0.9752	0.4645	66.9509
0.09	(0.5, 0.866, 0.5)	In-between mic1 and 2 inside the array at height of 0.5m at 60°	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0059, 0.0011, 0.0057, -0.0003)	(0.3669, 1.1491, 0.3550)	(-0.4631, 0.7422, 0.4100)	0.3448	0.9752	0.2364	66.9509

Looking at the results in Table 6.9 it is clear that increasing the noise level has great effect on the results. However, one trend is clear that after crossing a

certain threshold in N_0 the results stay the same. We tried different values of N_0 below 0.005 (which we did not include in the report because of space limitations) and we got the same results as for 0.005 case. Similarly for N_0 above 0.09 we got the same results as for the case of 0.09.

6.1.6 Conclusion of This Section

Since p and σ_{imp} has no significant effect on the results of the OC algorithm we can safely choose them as $p = 5, \sigma_{imp} = 1$ to cover their most expected values. As was discussed in Subsection 6.1.1 the best Dictionary in our case is Ψ_4 thus we will use it onward in our experiments. As was discussed in Subsection 6.1.5 any value below 0.005 for N_0 produce the same results as 0.005 thus we choose to keep $N_0 = 1.5 \times 10^{-4}$ which produce good results. Also based on the results in Subsection 6.1.2 the selection of index of maximum amplitude produce efficient results. This finding was also observed by the author of [6] in his work and thus in our coming experiments we use this index for time delay calculation.

6.2 Results using RevoLabs Wireless Microphone System

In this section we discuss in detail the indoor experiments we conducted using RevoLabs system and their results and analysis. As there are three different geometries and each geometry has different scenarios thus we need to organize

the discussion in subsections. Subsections 6.2.1, 6.2.2 and 6.2.3 discuss the results for Pyramid, Circular and Rhombus geometries in indoor environment using RevoLabs System, respectively. To save space we will not discuss the observation of individual scenario. The observations for all the indoor scenarios using RevoLab system will be discussed in detailed in subsection 6.2.4. It is to be noted that we used an unloaded toy gunshot for all the experiments using RevoLabs system.

6.2.1 Pyramid Geometry Experiments and Results

In this section we discuss the results for the Pyramid geometry obtained using RevoLabs wireless microphone system. We are using the geometry with the source locations discussed in Section 5.2.1.

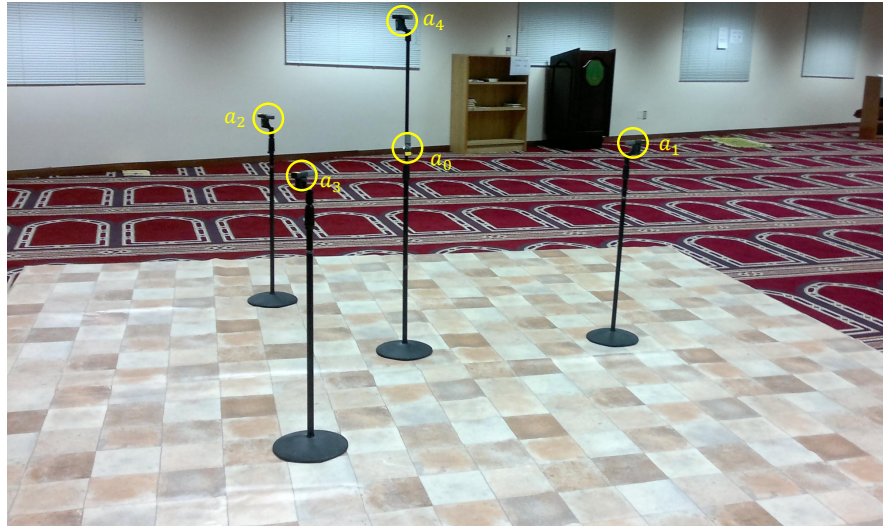
There are several experimental scenarios as was shown in Fig. 6.1. The results are organized with respect to the sampling rate and follow next.

Results for 8 kHz Case

Fig. 6.6 shows the experimental setup for the Pyramid geometry in the indoor environment. Fig. 6.6a shows the case when the array of microphones was placed in the center of the hall which represents a less-reverberant environment. Fig. 6.6b shows the case when array was placed in a corner of the hall which represents a more-reverberant environment. Note that the radius of geometry in Fig. 6.6a is small compared to Fig. 6.6b which we used to analyze the effects of the geometry size and here it is given only to show the experimental scenario.

Table 6.10 shows the results we obtained for the case when the array was

placed in the center of the hall where ϵ_{CC} represents the 3D error for CC method.



(a)



(b)

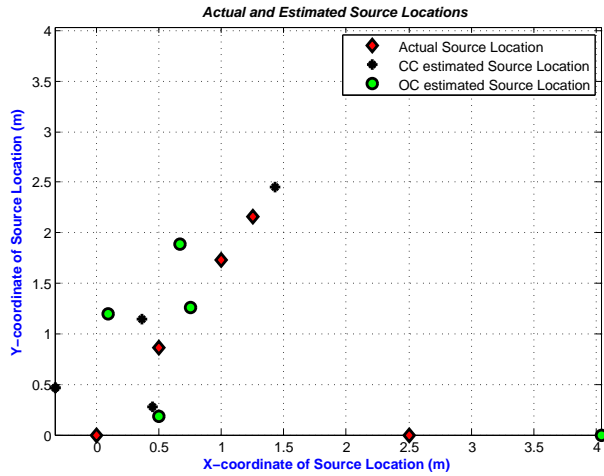
Figure 6.6: Indoor Pyramid Geometry Setup Using RevoLabs System, (a) Indoor Pyramid Geometry in the Center of the Hall using RevoLabs System, (b) Indoor Pyramid Geometry in a Corner of the Hall using RevoLabs System.

Fig. 7.4 shows the plot of the original source locations and the estimated source locations using CC and OC algorithms. Fig. 6.7a shows the 2D plot of x, y -coordinates of the estimated source locations for the less-reverberant case

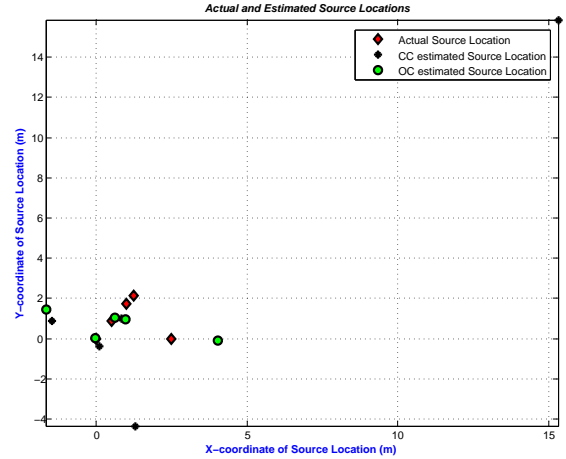
Table 6.10: Pyramid indoor results using RevoLabs at 8 kHz for the case when the array was placed in the center of the hall. Fast Machine was used.

Source Location	(0.5, 0.866, 0.5)	$(2\cos60^\circ, 2\sin60^\circ, 0) = (1, 1.73, 0)$	$(2.5\cos60^\circ, 2.5\sin60^\circ, 0) = (1.25, 2.16, 0)$	(0, 0, 1)	(2.5, 0, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	In-between a_1 & a_2 , outside the array	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{cc} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0045, 0.0024, 0.0085, 0.0001)	(0.0075, 0.0046, 0.0073, -0.0021)	(-0.0019, -0.0018, 0.0076, -0.0018)	(0.0075, 0.0046, 0.0073, -0.0021)	(-0.0064, 0.0044, 0.0056, 0.0118)
TDE_{oc} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0059, 0.0011, 0.0088, -0.0003)	(0.0013, 0.0015, 0.0070, 0.0012)	(0.0010, -0.0011, 0.0062, 0.0005)	(0.0025, 0.0043, 0.0051, -0.0010)	(-0.0049, 0.0035, 0.0035, 0.0010)
X_{cc} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(0.3669, 1.1491, 0.3550)	(-0.3303, 0.4604, -0.2349)	(1.4279, 2.4476, 1.4870)	(-0.3303, 0.4604, -0.2349)	(0.4549, 0.2798, -13.9459)
X_{oc} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.0982, 1.1934, 0.3381)	(0.7562, 1.2564, -0.2556)	(0.6719, 1.8823, -0.0756)	(0.4972, 0.1850, 0.6376)	(4.0349, 0, -1.8341)
ϵ_{cc}	0.3448	1.3587	1.5249	1.3587	14.0978
$\epsilon_{oc,3D}$	0.5430	0.5909	0.6458	0.6425	2.3916
$\epsilon_{oc,2D}$	0.5183	0.5327	0.6414	0.5305	1.5349
Time for CC (Sec)	0.0677	0.0334	0.0354	0.0348	0.0347
Time for OC (Sec)	62.0450	48.0434	55.2455	52.6566	121.6402

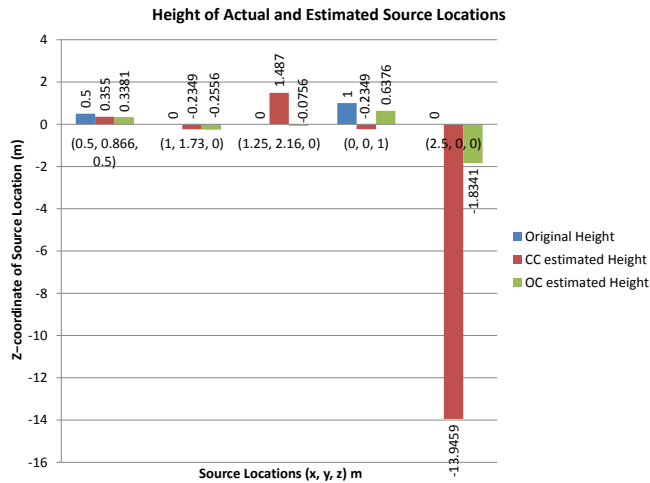
while Fig. 6.7b shows the same for the more-reverberant case. The height of the source positions are separately plotted in the form of bargraph. Fig. 6.7c and 6.7d shows the bargraphs of the heights for the less- and more-reverberant cases respectively.



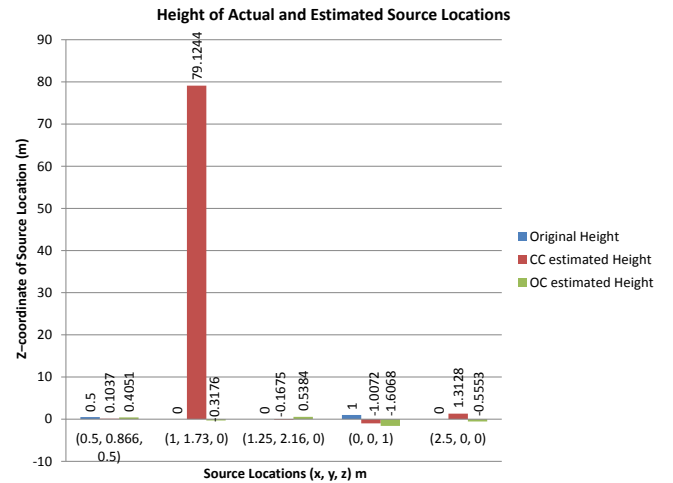
(a)



(b)



(c)



(d)

Figure 6.7: Pyramid Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case

Table 6.11 shows the results we obtained for the case when the array was placed in a corner of the hall.

Table 6.11: Pyramid indoor results using RevoLabs at 8 kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.

Source Location	(0.5, 0.866, 0.5)	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	(0, 0, 1)	(2.5, 0, 0)	(-4, 4, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array	On the line through a_2 and origin outside the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0016, 0.0058, 0.0005)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)	(0.0046, -0.0056, 0.0024, 0.0001)
TDE_{cc} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0041, 0.0060, 0.0104, -0.0009)	(-0.0021, -0.0008, 0.0032, -0.0043)	(0.0066, 0.0080, 0.0061, -0.0018)	(-0.0180, 0.0073, -0.0056, 0.0009)	(0.0060, -0.0041, 0.0031, -0.0019)
TDE_{oc} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0030, 0.0031, 0.0080, -0.0007)	(0.00010, 0.0023, 0.0060, 0.0010)	(0.0045, 0.0043, 0.0045, -0.0009)	(4.0336, -0.1112, -1.6068)	(0.0050, -0.0040, 0.0030, 0.0008)
X_{cc} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(0.8388, 0.9929, 0.1037)	(15.3081, 15.8584, 79.1244)	(0.1154, -0.3792, -0.1675)	(1.3065, -4.3425, -1.0072)	(-1.4631, 0.8716, 1.3128)
X_{oc} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.6096, 1.0304, 0.4051)	(0.9782, 0.9718, -0.3176)	(-0.0234, 0.0406, 0.5384)	(4.0336, -0.1112, -1.6068)	(-1.6662, 1.4543, -0.5553)
ϵ_{cc}	0.5366	81.7377	1.2329	4.6148	4.2363
$\epsilon_{oc,3D}$	0.2192	0.8223	0.4639	2.2240	3.4979
$\epsilon_{oc,2D}$	0.1976	0.7585	0.0469	1.5376	3.4535
Time for CC (Sec)	0.0356	0.0349	0.0355	0.0346	0.1918
Time for OC (Sec)	315.4692	352.7878	56.6916	134.9118	609.6994

Results for 4 kHz Case

The same experimental setup was used (shown in Fig. 6.6) as that for the case of 8 kHz. Table 6.12 shows the results we obtained for the case when the array was placed in the center of the hall while Table 6.13 shows the results for the case when the array was placed in a corner of the hall.

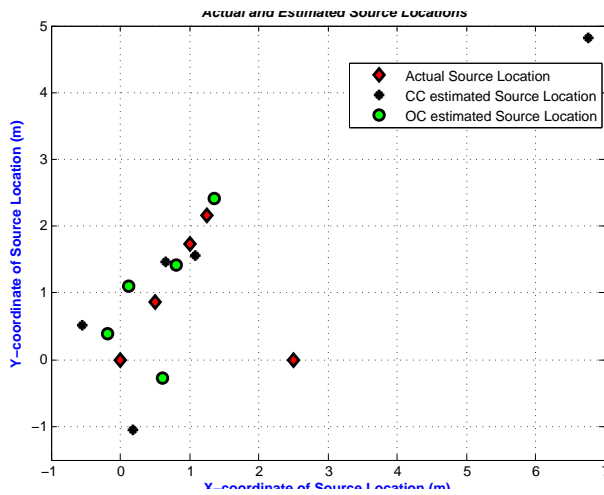
Using the results from Table 6.12 and 6.13 we plot the actual and estimated x, y coordinates of the source locations in Fig. 6.8a and Fig. 6.8b for the case of less-reverberant (center of hall) and more-reverberant (corner of hall) scenarios respectively. Fig. 6.8c and Fig. 6.8d compares the estimate of source height obtained using CC and OC with the actual heights.

Table 6.12: Pyramid indoor results using RevoLabs at 4 kHz for the case when the array was placed in the center of the hall. Slow Machine was used.

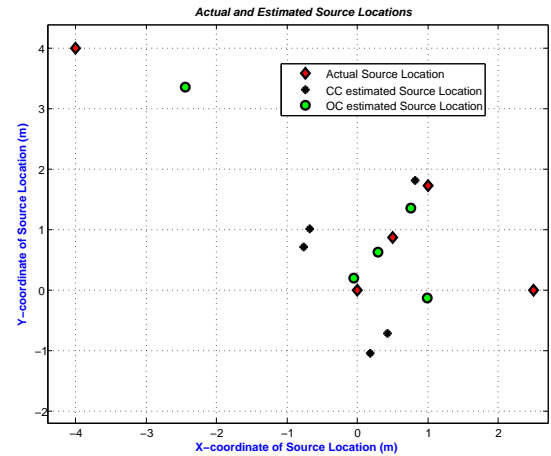
Source Location	(0.5, 0.866, 0.5)	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	$(2.5\cos 60^\circ, 2.5\sin 60^\circ, 0) = (1.25, 2.16, 0)$	(0, 0, 1)	(2.5, 0, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	In-between a_1 & a_2 , outside the array	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0025, 0.005, 0.0120, 0.0005)	(0.0013, 0.0003, 0.0063, 0.0015)	(-0.0177, 0.0003, 0.0080, -0.0145)	(0.0060, 0.0090, 0.0032, -0.0005)	(-0.0067, 0.0037, 0.0070, 0.0013)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0043, 0.0010, 0.0071, 0.0009)	(0.0010, 0.0009, 0.0071, 0.0015)	(-0.0017, -0.0021, 0.0078, 0.0010)	(0.0046, 0.0028, 0.0048, -0.0041)	(-0.0049, 0.0063, 0.0051, 0.0009)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(1.0841, 1.5576, 0.4870)	(0.6606, 1.4661, -0.8013)	(6.7612, 4.8144, 17.2162)	(0.1853, -1.0504, 0.2877)	(-0.5518, 0.5176, 0.3555)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.1258, 1.1027, 0.0974)	(0.8055, 1.4205, -0.5108)	(1.3545, 2.4144, -0.4361)	(-0.1756, 0.3952, 0.4071)	(0.6132, -0.2717, 0.1383)
ϵ_{cc}	0.9054	0.9094	18.2706	1.2826	3.1157
$\epsilon_{oc,3D}$	0.5984	0.6281	0.5156	0.7339	1.9113
$\epsilon_{oc,2D}$	0.4428	0.3655	0.2751	0.4325	1.9062
Time for CC (Sec)	0.0346	0.0344	0.0344	0.0327	0.0346
Time for OC (Sec)	21.1277	13.9726	13.0115	10.9131	14.2258

Table 6.13: Pyramid indoor results using RevoLabs at 4 kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.

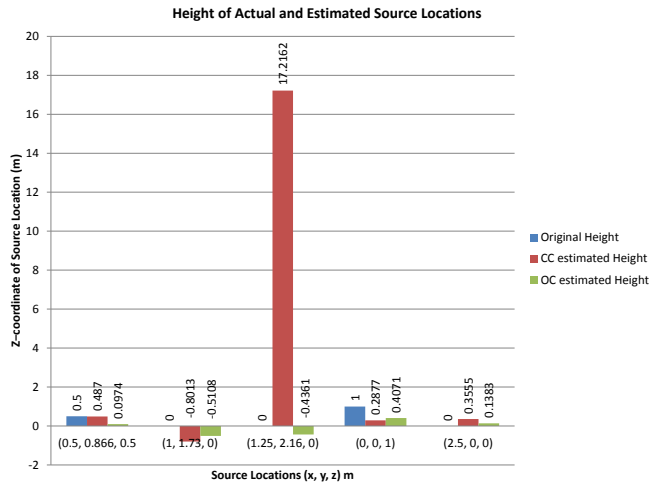
Source Location	(0.5, 0.866, 0.5)	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	(0, 0, 1)	(2.5, 0, 0)	(-4, 4, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array	On the line through a_2 and origin outside the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0000, 0.0059, 0.0004)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)	(0.0046, -0.0056, 0.0024, 0.0001)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0110, 0.0022, 0.0107, 0.0005)	(0.0060, 0.0090, 0.0032, -0.0005)	(-0.0005, 0.0005, 0.0177, 0.0010)	(-0.0043, -0.0077, 0.0047, 0.0015)	(0.0070, -0.0018, 0.0057, -0.0003)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0049, 0.0045, 0.0076, -0.0023)	(0.0013, 0.0011, 0.0071, 0.0015)	(0.0039, 0.0032, 0.0041, 0.0034)	(-0.0041, 0.0057, 0.0052, 0.0011)	(0.0034, -0.0039, 0.0027, 0.0015)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-0.6810, 1.0194, 0.5692)	(0.1853, -1.0504, 0.2877)	(0.8204, 1.8191, 1.2374)	(0.4252, -0.7154, 0.2312)	(-0.7597, 0.7131, 0.3891)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.2912, 0.6313, -0.0453)	(0.7515, 1.3531, -0.4607)	(-0.0589, 0.1921, -2.0679)	(0.9928, -0.1246, -0.0938)	(-2.4492, 3.3591, -3.6382)
ϵ_{cc}	1.1929	1.2826	1.2536	2.2069	4.6319
$\epsilon_{oc,3D}$	0.6293	0.6450	3.0745	1.5153	4.0065
$\epsilon_{oc,2D}$	0.3141	0.4515	0.2010	1.5124	1.6780
Time for CC (Sec)	0.0346	0.0344	0.0327	0.0346	0.0326
Time for OC (Sec)	21.1277	13.9726	10.9131	14.2258	17.4918



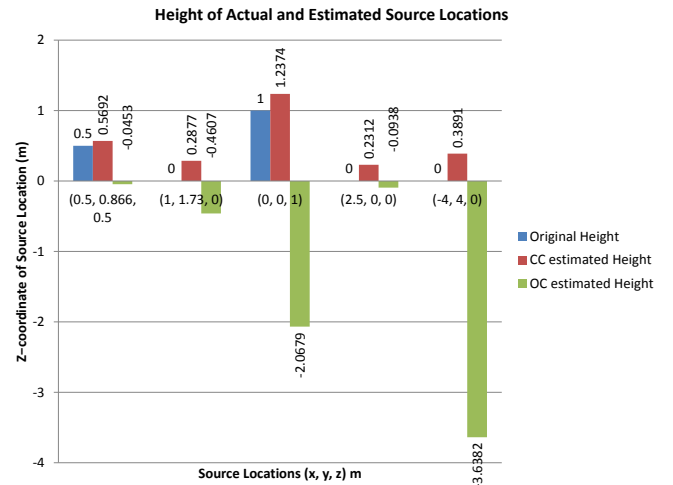
(a)



(b)



(c)



(d)

Figure 6.8: Pyramid Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.

6.2.2 Circular Geometry Experiments and Results

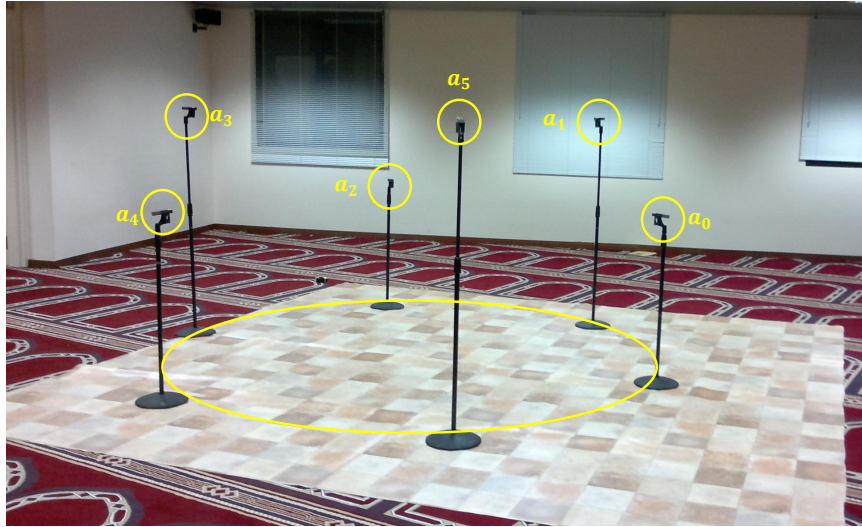
This section experiments use the circular geometry with the source locations discussed in Section 5.2.1. The values of different parameters of the OC algorithm were the same as were decided in Section 6.1.6.

There are two different experimental scenarios i.e. i). experimenting in the center of the hall and ii). experimenting in a corner of the hall. We call the first scenario as less-reverberant case and the second as more-reverberant case due to reflections from the wall. Two different sampling rates of 8 kHz and 4 kHz were used during experiments. The results for all these scenarios follows.

Results for the 8 kHz Case

Fig. 6.9 shows the experimental setup for the Circular geometry in indoor environment. Fig. 6.9a shows the case when array was placed in the center of the hall while Fig. 6.9b shows the case when array was placed in a corner of the hall.

Table 6.14 shows the results for the center of the hall case while Table 6.15 shows the results for the corner of the hall case. We plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.10a and Fig. 6.10b plots the x, y coordinates for the more-reverberant case.



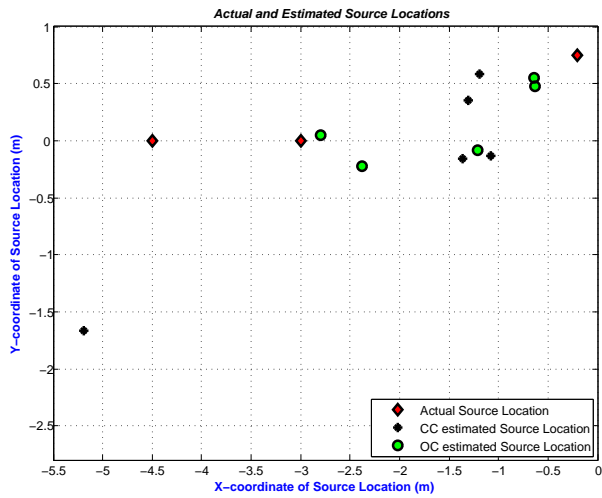
(a)



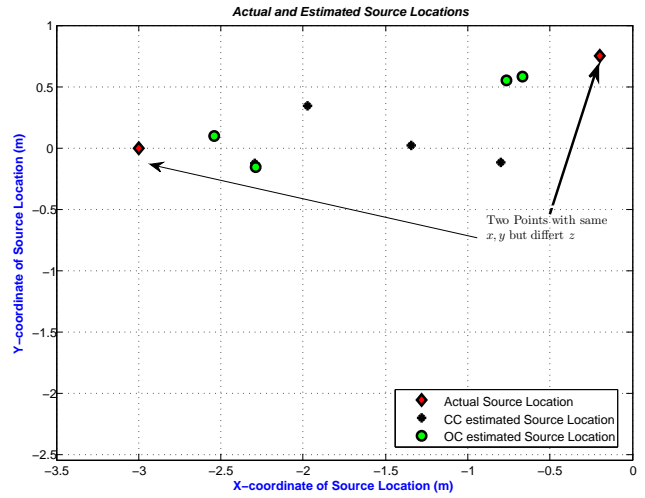
(b)

Figure 6.9: Indoor Circular Geometry Setup Using RevoLabs System, (a) Indoor Circular Geometry in the Center of the Hall using RevoLabs System, (b) Indoor Circular Geometry in a Corner of the Hall using RevoLabs System.

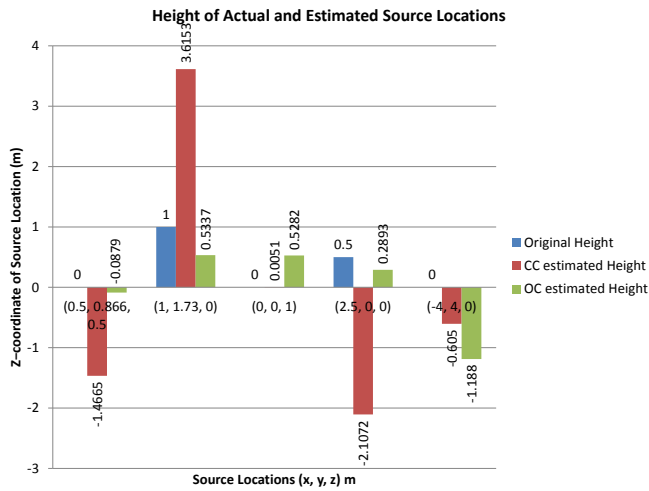
Note that in the case of array being in the corner of the hall, one source location $(-4.5, 0, 0)$ was crossing the dimension of the room thus it was not possible to test it. Fig. 6.10c and Fig. 6.10d show the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.



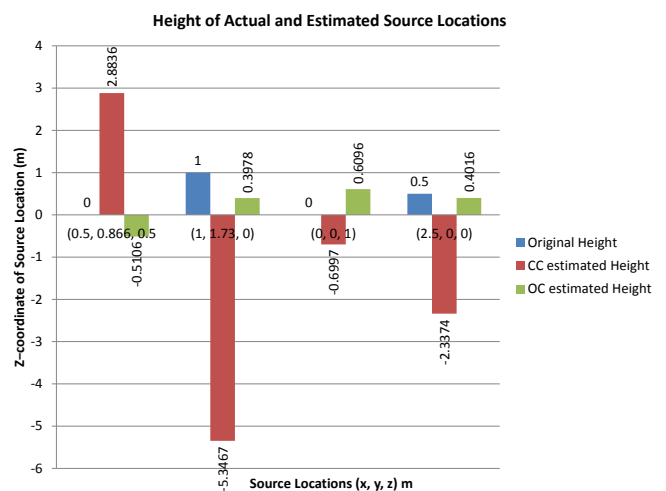
(a)



(b)



(c)



(d)

Figure 6.10: Circular Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case

Table 6.14: Circular indoor results using RevoLabs at 8 kHz for the case when the array was placed in the center of the hall. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0004, 0.0040, 0.0064, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0020, 0.0049, 0.0113, 0.0079, 0.0092)	(0.0013, 0.0041, 0.0119, 0.0079, 0.0045)	(0.0015, -0.0071, -0.0069, -0.0056, 0.0018)	(-0.0021, -0.0014, -0.0064, -0.0022, 0.0056)	(0.0029, 0.0010, -0.0061, -0.0065, 0.0005)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0014, 0.0033, 0.0057, 0.0056, 0.0035)	(-0.0007, 0.0024, 0.0041, 0.0043, 0.0024)	(-0.0013, -0.0047, -0.0063, -0.0038, -0.0015)	(-0.0003, -0.0031, -0.0078, -0.0037, -0.0017)	(0.0089, 0.0024, -0.0036, -0.0030, 0.0014)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-1.1901, 0.5856, -1.4665)	(-5.1927, -1.6669, 3.6153)	(-1.0795, -0.1313, 0.0051)	(-1.3075, 0.3541, -2.1072)	(-1.3625, -0.1550, -0.6050)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.6310, 0.4794, -0.0879)	(-0.6365, 0.5531, 0.5337)	(-2.7991, 0.0512, 0.5282)	(-2.3801, -0.2262, -2.893)	(-1.2150, -0.0819, -1.188)
$\epsilon_{cc,3D}$	1.7771	6.1325	1.9250	3.1284	3.1991
$\epsilon_{cc,2D}$	1.0036	5.5469	1.9250	1.7291	3.1414
$\epsilon_{oc,3D}$	0.5165	0.6684	0.5675	0.6927	3.4943
$\epsilon_{oc,2D}$	0.5089	0.4789	0.2074	0.6599	3.2860
Time for CC (Sec)	0.2172	0.1406	0.1597	1.3229	0.1470
Time for OC (Sec)	46.6671	50.7984	50.2909	53.5083	45.1788

Table 6.15: Circular indoor results using RevoLabs at 8 kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0022, 0.0098, 0.0116, 0.0145, 0.0040)	(-0.0041, 0.0027, 0.0077, 0.0083, 0.0134)	(0.0010, -0.0051, -0.0045, -0.0003, 0.0039)	(0.0044, -0.0069, -0.0074, -0.0029, 0.0070)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0008, 0.0026, 0.0051, 0.0048, 0.0038)	(-0.0005, 0.0021, 0.0038, 0.0041, 0.0024)	(-0.0015, -0.0047, -0.0059, -0.0035, -0.0014)	(-0.0005, -0.0029, -0.0068, -0.0034, -0.0015)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-1.9743, 0.3453, 2.8836)	(-2.2941, -0.1232, -5.3467)	(-1.3421, 0.0204, -0.6997)	(-0.7989, -0.1164, -2.3374)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.7621, 0.5498, -0.5106)	(-0.6702, 0.5848, 0.3978)	(-2.5398, 0.0998, 0.6096)	(-2.2858, -0.1586, 0.4016)
$\epsilon_{cc,3D}$	3.4098	6.7400	1.7997	3.5930
$\epsilon_{cc,2D}$	1.8198	2.2688	1.6581	2.2041
$\epsilon_{oc,3D}$	0.7853	0.7817	0.7703	0.7381
$\epsilon_{oc,2D}$	0.5967	0.4984	0.4709	0.7316
Time for CC (Sec)	0.2172	0.1406	0.1597	1.3229
Time for OC (Sec)	46.6671	50.7984	50.2909	53.5083

Results for the 4 kHz Case

The same experimental setup was used (shown in Fig. 6.9) as that for the case of 8 kHz. Table 6.16 shows the results we obtained for the less reverberant environment and Table 6.17 shows the results for the case when the array was placed in a corner of the hall which represents an indoor more-reverberant environment.

Table 6.16: Circular indoor results using RevoLabs at 4 kHz for the case when the array was placed in the center of the hall. Slow Machine was used.

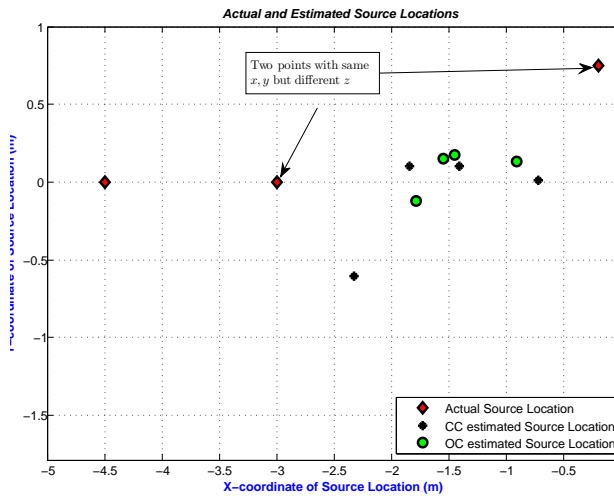
Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs ($\tau_{10}^c, \tau_{20}^c, \tau_{30}^c, \tau_{40}^c$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0, 0.0125, 0.0092, 0.0055, 0.0118)	(0.0018, 0.0032, 0.0073, 0.0047, 0.0083)	(0.0040, -0.0013, -0.0045, -0.0020, 0.0047)	(0.0040, -0.0013, -0.0045, -0.0020, 0.0047)	(0.0010, 0.0047, -0.0063, -0.0065, 0.0005)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0, -0.0013, 0.0053, 0.0013, -0.0008)	(0.2500, -0.7500, 0.2500, 0.2500, 0.7500)	(0.0018, -0.0020, -0.0030, -0.0005, -0.0003)	(0.0018, -0.0020, -0.0030, -0.0005, -0.0003)	(-0.0003, 0.0010, -0.0015, -0.0013, 0.0050)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-2.3313, -0.6064, 0.4805)	(-1.8464, 0.1032, 0.0713)	(-1.4135, 0.1001, -1.7072)	(-1.4135, 0.1001, -1.7072)	(-0.7240, 0.0093, 1.3066)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.9155, 0.1295, -0.0934)	(-1.5528, 0.1499, -0.3683)	(-1.7876, -0.1206, -0.3870)	(-1.7876, -0.1206, -0.3870)	(-1.4542, 0.1771, -0.2648)
$\epsilon_{cc,3D}$	2.5716	1.9978	2.3327	2.7201	3.9957
$\epsilon_{cc,2D}$	2.5263	1.7689	1.5897	1.5897	3.7760
$\epsilon_{oc,3D}$	0.9517	2.0156	1.2784	1.5071	3.0625
$\epsilon_{oc,2D}$	0.9471	1.4799	1.2184	1.2184	3.0510
Time for CC (Sec)	0.0522	0.0454	0.0457	0.0456	0.0468
Time for OC (Sec)	21.1344	17.5049	20.8175	20.7418	17.8947

We plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.11a and Fig. 6.11b plots the x, y coordinates for the more-reverberant case. Fig. 6.11c and Fig. 6.11d plots the bargraphs for

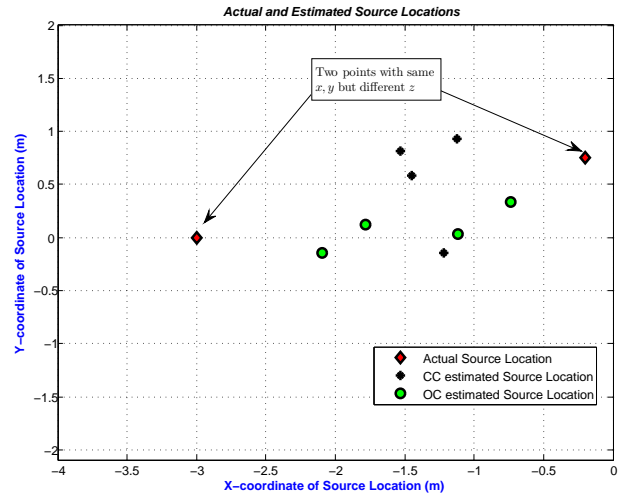
Table 6.17: Circular indoor results using RevoLabs at 4 kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)
TDE_{CC} ($\tau_{10}^{CC}, \tau_{20}^{CC}, \tau_{30}^{CC}, \tau_{40}^{CC}$)	(0.0047, 0.0147, 0.0115, 0.0187, 0.0085)	(0.0008, -0.0020, 0.0100, 0.0057, -0.0010)	(0.0010, 0.0005, -0.0077, -0.0035, 0.0088)	(-0.0005, -0.0053, -0.0115, -0.0063, 0.0053)
TDE_{OC} ($\tau_{10}^{OC}, \tau_{20}^{OC}, \tau_{30}^{OC}, \tau_{40}^{OC}$)	(0.0022, 0.0035, 0.0027, 0.0060, 0.0020)	(-0.0015, 0.0027, 0, 0.0010, 0.0015)	(0.0010, -0.0010, -0.0013, -0.0008, 0.0018)	(0.0005, -0.0032, -0.0027, -0.0032, -0.0008)
X_{CC} ($x_s^{CC}, y_s^{CC}, z_s^{CC}$)	(-1.4518, 0.5873, 7.7525)	(-1.2221, -0.1445, -0.1691)	(-1.1253, 0.9294, -3.8466)	(-1.5333, 0.8111, -4.1501)
X_{OC} ($x_s^{OC}, y_s^{OC}, z_s^{OC}$)	(-0.7377, 0.3325, -0.0450)	(-1.1163, 0.0331, 1.1544)	(-1.7843, 0.1202, -1.4160)	(-2.0937, -0.1438, -0.6772)
$\epsilon_{CC,3D}$	7.8546	1.7922	4.3789	4.9430
$\epsilon_{CC,2D}$	1.2624	1.3583	2.0925	1.6761
$\epsilon_{OC,3D}$	0.6823	1.1736	1.8701	1.4926
$\epsilon_{OC,2D}$	0.6808	1.1634	1.2216	0.9177
Time for CC (Sec)	0.0464	0.0472	0.0465	0.0543
Time for OC (Sec)	26.0555	22.3360	22.8007	18.6865

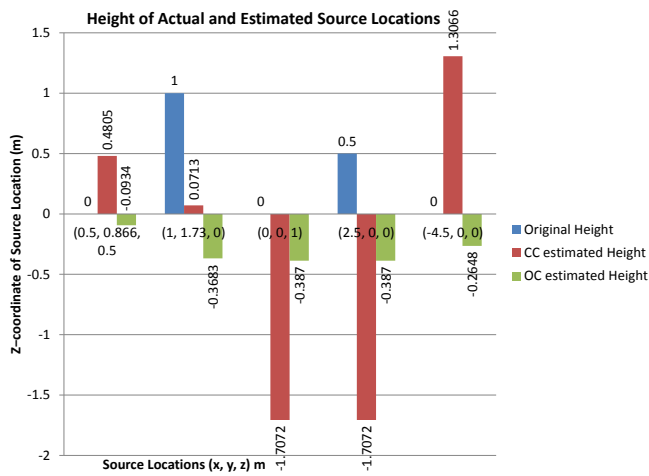
the height of source locations for the case of less- and more-reverberant scenarios respectively.



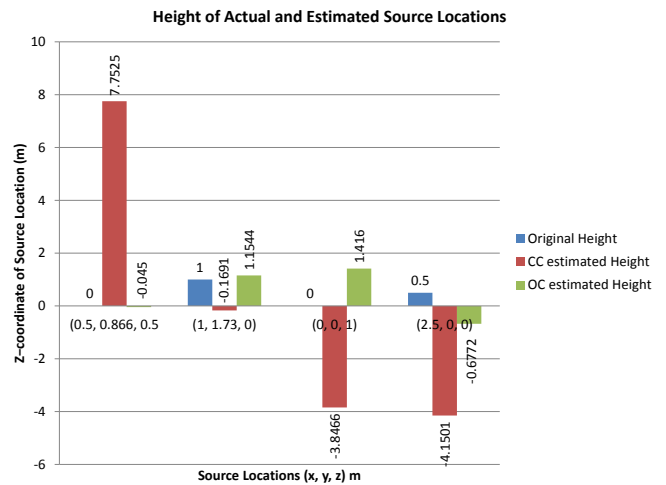
(a)



(b)



(c)



(d)

Figure 6.11: Circular Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case

6.2.3 Rhombus Geometry Experiments and Results

Now we will discuss the Rhombus geometry results in indoor environment. We are using the geometry with the source locations discussed in Section 5.2.1. The values of different parameters of the OC algorithm were the same as were decided in Section 6.1.



(a)



(b)

Figure 6.12: Indoor Rhombus Geometry Experimental Setup Using RevoLabs System, (a) Indoor Rhombus Geometry in the Center of the Hall using RevoLabs System, (b) Indoor Rhombus Geometry in a Corner of the Hall using RevoLabs System.

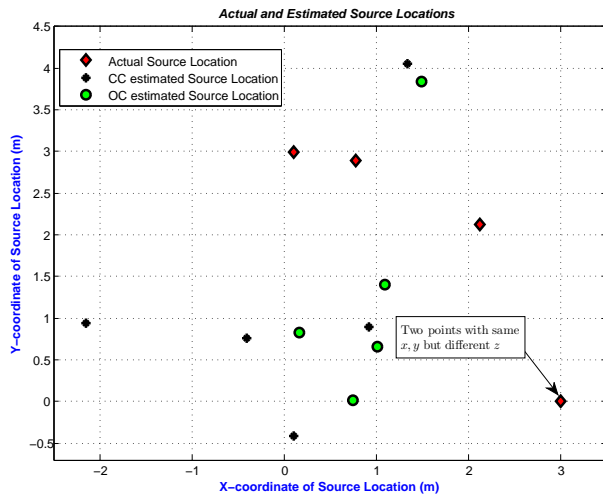
Like Pyramid and Circular geometries there are two different experimental scenarios i.e. i) experimenting in the center of the hall and ii) experimenting in a corner of the hall. The experimental setup for these scenarios is shown in Fig. 6.12. Fig. 6.12a shows the case of array placed in the center of hall while Fig. 6.12b shows the case of array placed in a corner of the hall.

Two different sampling rates of 8 kHz and 4 kHz were used during experiments. We also captured signals at 16 kHz sampling rate but we could not run the OC algorithm for this case as the memory required for OC algorithm is quite large (our system had 8 gigabyte RAM and still it was not able to run the MATLAB code). CC can be run for 16kHz rate but then we would not be able to compare it with OC and that is why we did not run it for 16kHz. The results for all these scenarios follow next.

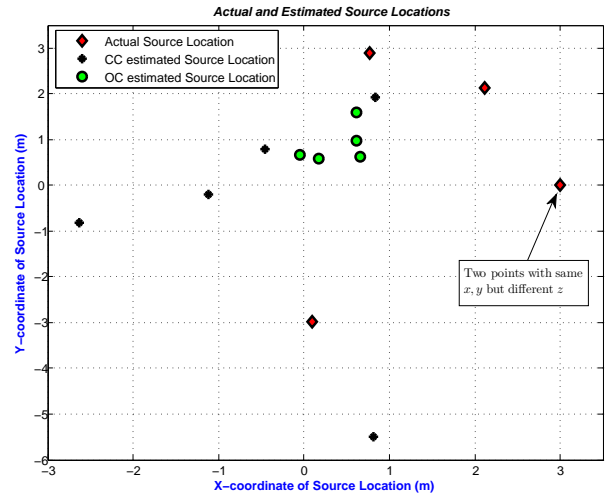
Results for the Case of 8 kHz

We run the OC algorithm for the experimental signals captured at 8 kHz. The results for the case of array being in the center of the hall are given in Table 6.18 while Table 6.19 shows the results for the case of array placed in a corner of the hall.

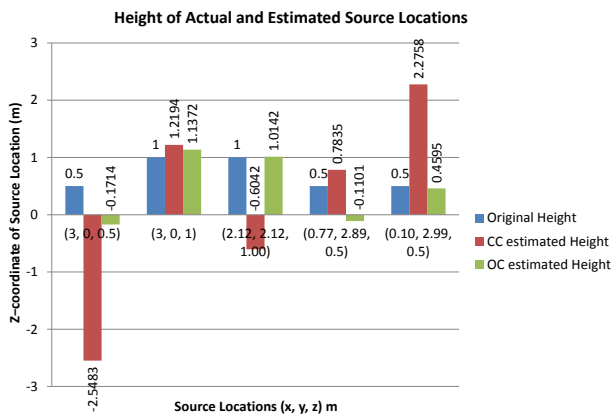
We plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.13a and Fig. 6.13b plots the x, y coordinates for the more-reverberant case. Fig. 6.13c and Fig. 6.13d plots the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.



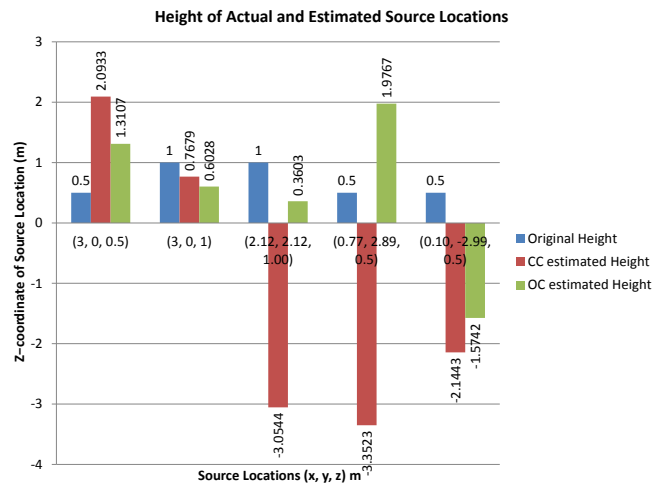
(a)



(b)



(c)



(d)

Figure 6.13: Rhombus Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.

Table 6.18: Rhombus indoor results using RevoLabs at 8 kHz for the case when the array was placed in the center of the hall. Fast Machine was used.

Source Location	(2r, 0, h)	(2r, 0, 2h)	(2r cos 45°, 2r sin 45°, 2h)	(2r cos 75°, 2r sin 75°, h)	(2r cos 88°, 2r sin 88°, h)
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, 2.99, 0.5)
Description	On the line passing through mic1 and center of the array at height of 0.5m	On the line passing through mic1 and center of the array at height of 1m	In-between mic1 and 2 outside the circumference on which mics lie.	At angle of 75° between mic1 and 2	At angle of 88° between mic1 and 2
Exact TDs (τ ₁₀ ^{cc} , τ ₂₀ ^{cc} , τ ₃₀ ^{cc} , τ ₄₀ ^{cc})	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE _{cc} (τ ₁₀ ^{cc} , τ ₂₀ ^{cc} , τ ₃₀ ^{cc} , τ ₄₀ ^{cc})	(0.0001, -0.0014, 0.0016, 0.0015, 0.0011)	(-0.0037, 0.0011, 0.0092, 0.0001, 0.0008)	(-0.0016, -0.0025, 0.0066, 0.0041, 0.0040)	(0.0026, -0.0034, -0.0019, 0.0022, -0.0027)	(0, -0.0009, 0.0113, 0.0036, 0.0039)
TDE _{oc} (τ ₁₀ ^{oc} , τ ₂₀ ^{oc} , τ ₃₀ ^{oc} , τ ₄₀ ^{oc})	(-0.0003, 0.0005, 0.0036, 0.0009, 0.0011)	(-0.0012, 0.0013, 0.0030, -0.0009, -0.0020)	(-0.0026, -0.0032, 0.0029, 0.0012, -0.009)	(-0.0009, -0.0100, 0.0030, 0.0020, 0.0005)	(0.0015, -0.00030, 0.0029, 0.0036, 0.0010)
X _{cc} (x ₁₀ ^{cc} , y ₁₀ ^{cc} , z ₁₀ ^{cc})	(1.3348, 4.0596, -2.5483)	(-2.1503, 0.9416, 1.2194)	(0.9136, 0.9012, -0.6042)	(-0.4036, 0.7654, 0.7835)	(0.1023, -0.4085, 2.2758)
X _{oc} (x ₁₀ ^{oc} , y ₁₀ ^{oc} , z ₁₀ ^{oc})	(1.0140, 0.6643, -0.1714)	(0.7455, 0.0120, 1.1372)	(1.0907, 1.4063, 1.0142)	(1.4865, 3.8414, -0.1101)	(0.1592, 0.8330, 0.4595)
ε _{cc,3D}	5.3428	5.2403	2.3483	2.4437	3.8345
ε _{cc,2D}	4.3878	5.2357	1.7149	2.4272	3.3985
ε _{oc,3D}	2.1991	2.2587	1.2526	1.3382	2.1461
ε _{oc,2D}	2.0941	2.2545	1.2525	1.1910	2.1457
Time for CC (Sec)	0.0977	0.2304	0.0985	0.0446	0.2216
Time for OC (Sec)	191.6126	309.6996	62.0895	56.1844	83.4175

Table 6.19: Rhombus indoor results using RevoLabs at 8 kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.

Source Location	(2r, 0, h)	(2r, 0, 2h)	(2r cos 45°, 2r sin 45°, 2h)	(2r cos 75°, 2r sin 75°, h)	(2r cos 88°, -2r sin 88°, h)
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, -2.99, 0.5)
Description	On the line passing through mic1 and center of the array at height of 0.5m	On the line passing through mic1 and center of the array at height of 1m	In-between mic1 and 2 outside the circumference on which mics lie.	At angle of 75° between mic1 and 2	At angle of 88° between mic1 and 2
Exact TDs (τ ₁₀ ^{cc} , τ ₂₀ ^{cc} , τ ₃₀ ^{cc} , τ ₄₀ ^{cc})	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE _{cc} (τ ₁₀ ^{cc} , τ ₂₀ ^{cc} , τ ₃₀ ^{cc} , τ ₄₀ ^{cc})	(-0.0029, -0.0003, 0.0140, 0.0121, 0.0031)	(-0.0032, 0.0024, 0.0061, 0.0001, 0.0005)	(0.0035, -0.0044, 0.0063, 0.0034, -0.0116)	(-0.0109, -0.0079, -0.0010, 0.0076, -0.0050)	(0.0021, 0.0101, 0.0049, -0.0024, 0.0031)
TDE _{oc} (τ ₁₀ ^{oc} , τ ₂₀ ^{oc} , τ ₃₀ ^{oc} , τ ₄₀ ^{oc})	(-0.0006, -0.0026, 0.0010, -0.0018, -0.0006)	(0.0020, 0.0008, 0.0069, 0.0026, 0.0006)	(-0.0004, -0.0013, 0.0036, 0.0024, 0.0004)	(0.0020, 0.0019, 0.0029, 0.0074, 0.0026)	(0.0026, -0.0021, 0.0054, 0.0049, 0.0043)
X _{cc} (x ₁₀ ^{cc} , y ₁₀ ^{cc} , z ₁₀ ^{cc})	(-1.1178, -0.1888, 2.0933)	(-0.4590, 0.7892, 0.7679)	(0.8369, 1.9320, -3.0544)	(-2.6302, -0.8123, -3.3523)	(0.8101, -5.4995, -2.1443)
X _{oc} (x ₁₀ ^{oc} , y ₁₀ ^{oc} , z ₁₀ ^{oc})	(0.1703, 0.5928, 1.3107)	(0.6552, 0.6313, 0.6028)	(0.6115, 0.9707, 0.3603)	(-0.0445, 0.6594, 1.9767)	(0.6103, 1.5906, -1.5742)
ε _{cc,3D}	4.4194	3.5555	4.2567	6.3921	3.7140
ε _{cc,2D}	4.1221	3.5479	1.2968	5.1009	2.6080
ε _{oc,3D}	3.0027	2.4606	2.0014	2.7964	2.5536
ε _{oc,2D}	2.8912	2.4283	1.8964	2.3747	1.4895
Time for CC (Sec)	0.0370	0.0370	0.0355	0.0369	0.0369
Time for OC (Sec)	67.6747	62.1999	70.1450	369.7572	50.6591

Results for the Case of 4 kHz

For the 4 kHz the same experimental setup was used as for the 8 kHz case shown in Fig. 6.12. The results for the case of array being in the center of the hall are

given in Table 6.20 while Table 6.21 shows the results for the case of array placed in a corner of the hall.

Table 6.20: Rhombus indoor results using RevoLabs at 4 kHz for the case when the array was placed in the center of the hall. Slow Machine was used.

Source Location	(2r, 0, h)	(2r, 0, 2h)	(2r cos 45°, 2r sin 45°, 2h)	(2r cos 75°, 2r sin 75°, h)	(2r cos 88°, 2r sin 88°, h)
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, 2.99, 0.5)
Description	On the line passing through mic1 and center of the array at height of 0.5m	On the line passing through mic1 and center of the array at height of 1m	In-between mic1 and 2 outside the circumference on which mics lye.	At angle of 75° between mic1 and 2	At angle of 88° between mic1 and 2
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE _{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(-0.0003, 0.0095, 0.0080, 0.0045, 0.0027)	(0.0015, 0.0018, 0.0080, -0.0005, 0.0018)	(-0.0010, 0.0005, 0.0063, 0.0037, 0.0040)	(0.0018, -0.0027, -0.0022, -0.0022, -0.0018)	(0.0103, 0.0008, -0.0010, 0.0063, 0.0013)
TDE _{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(-0.0027, -0.0018, 0.0027, -0.0020, 0)	(0.0003, 0.0008, 0.0025, 0.0060, 0.0003)	(0.0005, 0, 0.0037, 0.0020, 0.0037)	(-0.0010, -0.0037, -0.0013, -0.0015, -0.0027)	(0.0055, -0.0010, 0.0030, 0.0020, 0.0022)
X _{CC} ($x_1^{cc}, y_1^{cc}, z_1^{cc}$)	(0.2274, -0.6758, 1.1663)	(0.7030, 0.6986, 0.7819)	(1.2434, 1.1928, -1.2197)	(-0.3748, 0.5857, 1.4128)	(-0.4298, 0.6130, 0.9153)
X _{OC} ($x_1^{oc}, y_1^{oc}, z_1^{oc}$)	(0.1469, 0.5171, 0.8753)	(-0.1002, 0.6371, 1.2008)	(1.0207, 1.4610, -2.1807)	(-0.0481, 1.0586, 2.7918)	(-0.3984, 0.6275, 0.3538)
$\epsilon_{cc,3D}$	2.9305	2.4108	2.5603	2.7301	2.4705
$\epsilon_{oc,3D}$	2.8538	2.4009	1.2760	2.5730	2.4354
$\epsilon_{cc,2D}$	2.9237	3.1713	3.4292	3.0456	2.4189
$\epsilon_{oc,2D}$	2.8995	3.1650	1.2817	2.0058	2.4145
Time for CC (Sec)	0.0464	0.1200	0.8762	0.0481	0.0528
Time for OC (Sec)	51.0664	198.7423	143.7500	246.7433	177.3496

Table 6.21: Rhombus indoor results using RevoLabs at 4 kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.

Source Location	(2r, 0, h)	(2r, 0, 2h)	(2r cos 45°, 2r sin 45°, 2h)	(2r cos 75°, 2r sin 75°, h)	(2r cos 88°, 2r sin(π + 88°), h)
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.99, 0.5)	(0.10, -2.99, 0.5)
Description	On the line passing through mic1 and center of the array at height of 0.5m	On the line passing through mic1 and center of the array at height of 1m	In-between mic1 and 2 outside the circumference on which mics lye.	At angle of 75° between mic1 and 2	At angle of 88° between mic4 and 5
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0001, -0.0043, 0.0018, 0.0013, 0)	(0.0008, 0.0043, 0.0011, -0.0016, 0)
TDE _{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0018, 0.0110, 0.0168, 0.0075, 0.0035)	(-0.0040, 0.0045, 0.0113, 0.0008, 0.0043)	(0.0018, 0.0010, 0.0053, 0.0040, 0.0015)	(-0.0045, -0.0100, 0.0065, 0.0095, -0.0055)	(0.0035, -0.0053, 0.0182, -0.0168, 0.0010)
TDE _{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0, 0.0020, 0.0043, 0.0030, 0.0035)	(-0.0013, -0.0008, 0.0020, -0.0030, -0.0027)	(-0.0005, 0.0022, 0.0035, 0.0015, 0.0022)	(0.0010, -0.0020, -0.0005, -0.0040, -0.0032)	(0.0025, -0.0020, -0.0018, -0.0043, -0.0035)
X _{CC} ($x_1^{cc}, y_1^{cc}, z_1^{cc}$)	(1.4493, 0.5969, 2.6610)	(-6.2309, 2.9453, 6.2661)	(0.4504, 0.7719, 0.3735)	(-1.3827, -4.9766, -3.2025)	(6.9532, 4.8568, -9.5040)
X _{OC} ($x_1^{oc}, y_1^{oc}, z_1^{oc}$)	(0.6940, 0.6608, -0.7102)	(0.4880, 0.4083, 1.5759)	(1.3885, -0.0381, -1.6142)	(-0.2182, 0.5655, 1.7190)	(-0.5450, 0.6186, 1.3547)
$\epsilon_{cc,3D}$	2.7260	11.0280	2.2355	9.0448	14.4437
$\epsilon_{oc,3D}$	1.6616	9.6894	2.1460	8.2523	10.4182
$\epsilon_{cc,2D}$	2.6868	2.6093	3.4679	2.8881	3.7641
$\epsilon_{oc,2D}$	2.3988	2.5450	2.2787	2.6182	3.6658
Time for CC (Sec)	0.1255	1.1706	1.1435	0.2140	2.4819
Time for OC (Sec)	283.6486	111.0679	110.458	710.8866	279.8250

We plot the x, y coordinates of the actual and estimated source locations for

the less-reverberant case in Fig. 6.14a and Fig. 6.14b plots the x, y coordinates for the more-reverberant case.

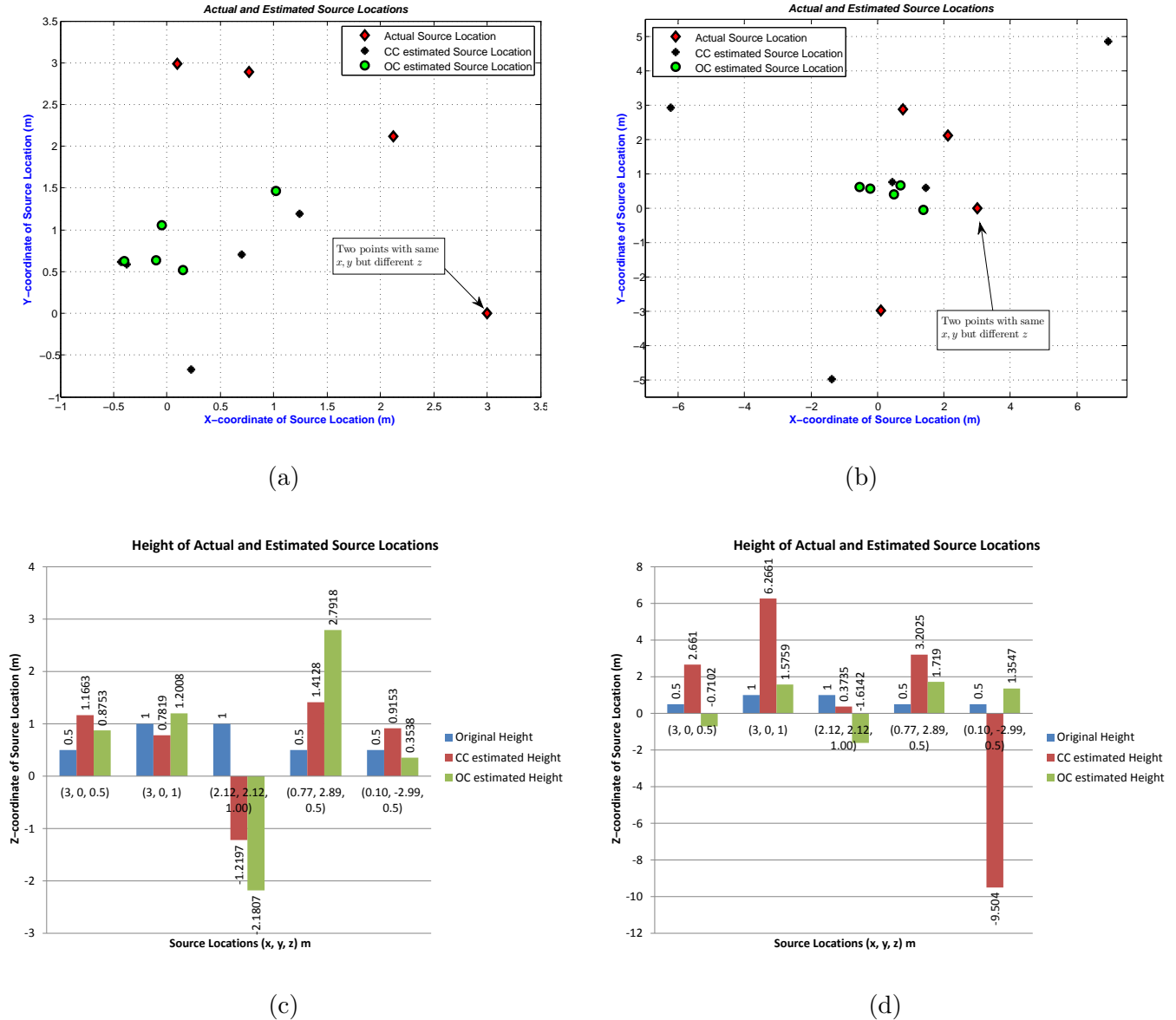


Figure 6.14: Rhombus Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using RevoLabs system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.

Fig. 6.14c and Fig. 6.14d plots the bargraphs for the height of source locations

for the case of less- and more-reverberant scenarios respectively.

6.2.4 Observations for the Experiments using RevoLabs System

Looking at the results in subsections 6.2.1 to 6.2.3 we can find the numerical facts about the results. We present these numerical facts in subsection 6.2.4 and then based on these numerical facts we deduce the observations and present them in subsection 6.2.4

Numerical Facts

1. For the Pyramid 8kHz, less-reverberant case: 80% of the time OC produces more correct results than CC by a factor of at least 2.3 i.e. OC 3D-MSE ($\epsilon_{OC.3D}$) is at least 2.3 times less than CC 3D-MSE ($\epsilon_{CC.3D}$) for 80% of the source locations. The runtime for OC is at least 900 times more than the runtime for CC.
2. For the Pyramid 8kHz, more-reverberant case: 100% of the time OC produces more correct results than CC by a factor of at least 1.21. The runtime for OC is at least 1000 times more than the runtime for CC.
3. For the Pyramid 4kHz, less-reverberant case: 100% of the time OC produces more correct results than CC by a factor of at least 1.44. The runtime for OC is at least 300 times more than the runtime for CC.

4. For the Pyramid 4kHz, more-reverberant case: 80% of the time OC produces more correct results than CC by a factor of at least 1.15. The runtime for OC is at least 300 times more than the runtime for CC.
5. For the Circular 8kHz, less-reverberant case: 80% of the time OC produces more correct results than CC by a factor of at least 3.44. The runtime for OC is at least 200 times more than the runtime for CC.
6. For the Circular 8kHz, more-reverberant case: 100% of the time OC produces more correct results than CC by a factor of at least 2.33. The runtime for OC is at least 200 times more than the runtime for CC.
7. For the Circular 4kHz, less-reverberant case: 80% of the time OC produces more correct results than CC by a factor of at least 1.3. The runtime for OC is at least 380 times more than the runtime for CC.
8. For the Circular 4kHz, more-reverberant case: 100% of the time OC produces more correct results than CC by a factor of at least 1.52. The runtime for OC is at least 340 times more than the runtime for CC.
9. For the Rhombus 8kHz, less-reverberant case: 100% of the time OC produces more correct results than CC by a factor of at least 1.78. The runtime for OC is at least 600 times more than the runtime for CC.
10. For the Rhombus 8kHz, more-reverberant case: 100% of the time OC produces more correct results than CC by a factor of at least 1.44. The runtime for OC is at least 1300 times more than the runtime for CC.

11. For the Rhombus 4kHz, less-reverberant case: 60% of the time CC produces more correct results than OC by a factor of at least 1.115. The runtime for OC is at least 1100 times more than the runtime for CC.
12. For the Rhombus 4kHz, more-reverberant case: 80% of the time OC produces more correct results than CC by a factor of at least 1.014. The runtime for OC is at least 90 times more than the runtime for CC.

Indoor RevoLabs Experiments Observations

Looking at the numerical facts in subsection 6.2.4 we can make the following observations for the experiments carried out indoor using RevoLabs system.

1. 80% of the time OC produces better results for the Pyramid geometry for the less reverberant case by a factor of at least 1.44 while 100% times for the more-reverberant case by a factor of at least 1.15 than the CC. But this accuracy comes at the cost of at least 300 times more runtime taken by OC than CC both for the less- and more-reverberant environments.
2. 80% of the time OC produces better results for the Circular geometry for the less reverberant case by a factor of at least 1.3 while 100% times for the more-reverberant case by a factor of at least 1.52 than the CC. But this accuracy comes at the cost of at least 200 times more runtime taken by OC than CC both for the less- and more-reverberant environments.
3. At least 70% of the time OC produces better results for the Rhombus geometry for the less reverberant case by a factor of at least 1.78 while at least

80% of the time for the more-reverberant case by a factor of at least 1.014 than the CC. But this accuracy comes at the cost of at least 600 times more runtime taken by OC than CC for the less-reverberant case and at least 90 times more runtime by OC than CC for the more-reverberant case.

4. The Rhombus geometry produces the least accurate results with a minimum $\epsilon_{OC,3D}$ of value 1.25m among the three geometries while the Pyramid geometry produces the most accurate results with a minimum $\epsilon_{OC,3D}$ of value 0.21m. The Circular is in-between Rhombus and Pyramid, accuracy-wise, with a minimum $\epsilon_{OC,3D}$ of value 0.51m.
5. The 8kHz case is taking at least 2 times more runtime for OC than 4kHz case for all the geometries.
6. The OC algorithm is taking at least 90 times more runtime than CC for all the cases of indoor experimentation using RevoLabs.
7. 100% of the time the OC produces more accurate results than CC in the more-reverberant environment for all the indoor experimental scenarios using RevoLabs while at least 70% of the time in the less-reverberant environment for all indoor experimental scenarios. This shows that as the reverberation increases the performance of both OC and CC decreases because of the signal shape distortion due to increased reverberations. However, CC is more disturbed by the reverberation than OC.

6.3 Results using VocoPro Wireless Microphone System

This section discusses the indoor experiments and their results and analysis for the Pyramid, Circular and Rhombus geometries using VocoPro Wireless Microphone System. It is to be noted that the sensitivity of VocoPro microphones were very weak because of its specific design for vocals [40] and we could not capture the unloaded toy gunshot thus we used a loaded toy gunshot for these experiments. However the loaded gunshot signal has different shape and characteristics which affect the results significantly as we will see in the coming pages.

Fig. 6.15 shows a sample loaded shot captured at 16kHz using VocoPro system in the indoor environment. It is to be noted that the maximum balanced output level of the VocoPro system is 400mV [40] which gets attenuated upon traveling from the base station of VocoPro to the PC through several connectors as was shown in section 5.3.2 thus we had to magnify the signal by a multiplicative factor of 100 in MATLAB. First we will present the results in both tabular and plotted forms and then we will discuss the observations for all the scenarios in subsection 6.3.4 for the VocoPro system.

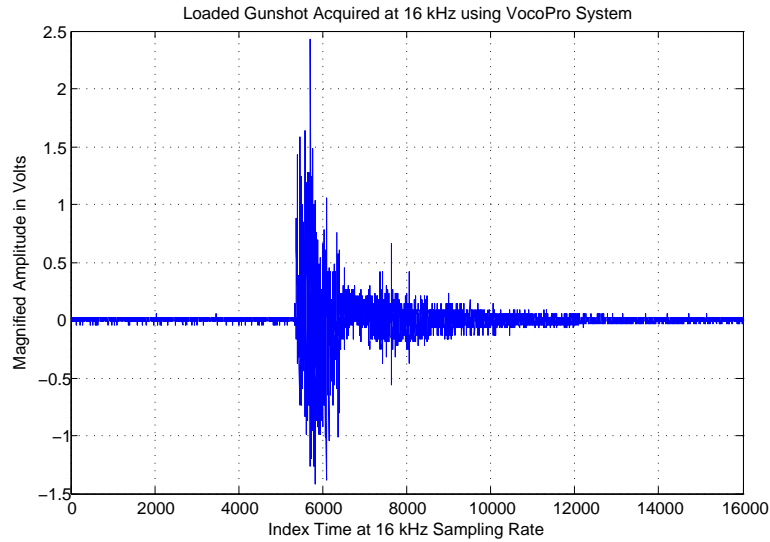


Figure 6.15: A sample unloaded gunshot acquired using VocoPro system at 16 kHz, amplified 100 times in MATLAB

6.3.1 Pyramid Geometry Experiments and Results

We discuss the results for the Pyramid geometry experiments in indoor environment using VocoPro system in this section. We conducted the experiments at 8kHz and 4kHz sampling rate. Fig. 6.16 shows the experimental setup for Pyramid geometry using VocoPro system indoors. Fig. 6.16a shows the experimental setup for the case of array placed in the center of the hall while Fig. 6.16b shows the experimental setup for the case of array placed in a corner of the hall. The VocoPro microphones are directional and thus can capture the sound waves coming perpendicularly (or at angles around 90°). We kept the mics standing upward rather than directing it to a specific locations for all the source locations and experiments for the sake of consistency.



(a)



(b)

Figure 6.16: Indoor Pyramid Geometry Experimental Setup Using VocoPro System, (a) Indoor Pyramid Geometry in the Center of the Hall using VocoPro System, (b) Indoor Pyramid Geometry in a Corner of the Hall using VocoPro System

For each sampling rate there are two different scenarios i) experimentation in the center of the hall and ii). experimentation in the corner of the hall. We divide the discussion further into two subsections w.r.t. to the sampling rate as follows.

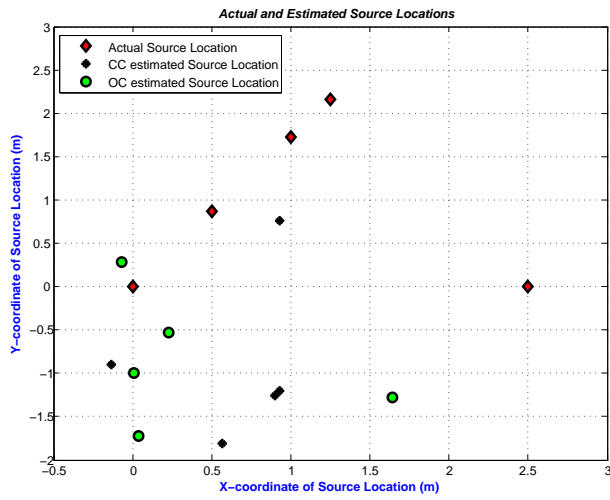
Results for 8kHz Sampling Rate

Table 6.22 shows the results for 8kHz when the array was placed in the center of the hall while Table 6.23 shows the results for 8kHz but this time the array was placed in a corner of the hall.

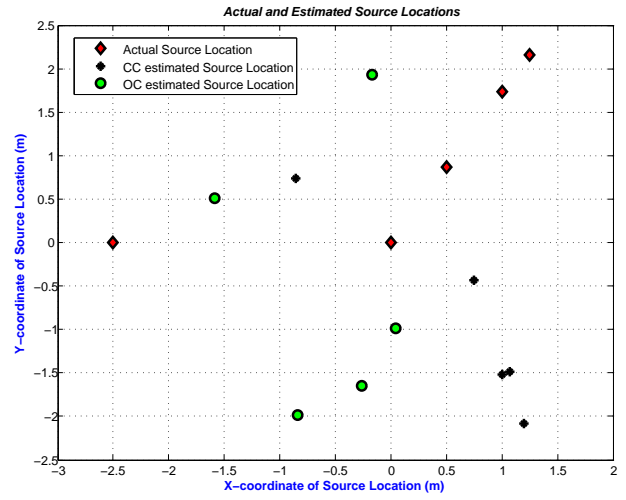
Table 6.22: Pyramid indoor results using VocoPro System at 8kHz for the case when the array was placed in the center of the hall. Fast Machine was used.

Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	$(2.5\cos60^\circ, 2.5\sin60^\circ, 0) = (1.25, 2.16, 0)$	$(2\cos60^\circ, 2\sin60^\circ, 0) = (1, 1.73, 0)$	(2.5, 0, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between a_1 & a_2 , outside the array	In-between a_1 & a_2 on the circle inscribing the array.	Along the line (outside array) passing through a_1 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0005, 0.0066, 0.0035, -0.0011)	(0.0088, -0.0070, 0.0019, -0.0008)	(-0.0013, 0.0096, 0.0041, -0.0019)	(0.0004, 0.0061, 0.0013, 0)	(-0.0054, 0.0087, 0.0034, 0.0024)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0043, 0.0064, 0.0010, 0.0018)	(0.0029, 0.0021, 0.0032, -0.0020)	(-0.0063, -0.0085, -0.0059, -0.0020)	(0.0015, -0.0048, -0.0014, 0.0044)	(-0.0030, -0.0061, 0.0037, 0.0069)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(0.9277, -0.7554, 0.6606)	(0.5659, -1.8107, -0.1469)	(0.9304, -1.2065, -0.0141)	(0.8994, -1.2575, 0.3600)	(-0.1363, -0.9037, 0.2600)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.0051, -1.0006, -0.4356)	(-0.0683, 0.2780, 1.6199)	(0.2307, -0.5347, 0.3720)	(1.6411, -1.2758, 4.5593)	(0.0350, -1.7220, 1.0418)
$\epsilon_{cc,3D}$	1.6845	2.2168	3.3817	3.0108	2.8801
$\epsilon_{cc,2D}$	1.6769	1.8970	3.3817	2.9892	2.8684
$\epsilon_{oc,3D}$	2.1458	0.6828	2.9050	5.4985	3.7326
$\epsilon_{oc,2D}$	1.9311	0.2863	2.8810	3.0734	3.5843
Time for CC (Sec)	0.0555	0.0337	0.1205	0.0342	0.0347
Time for OC (Sec)	46.5581	49.8827	48.7259	39.8475	40.3569

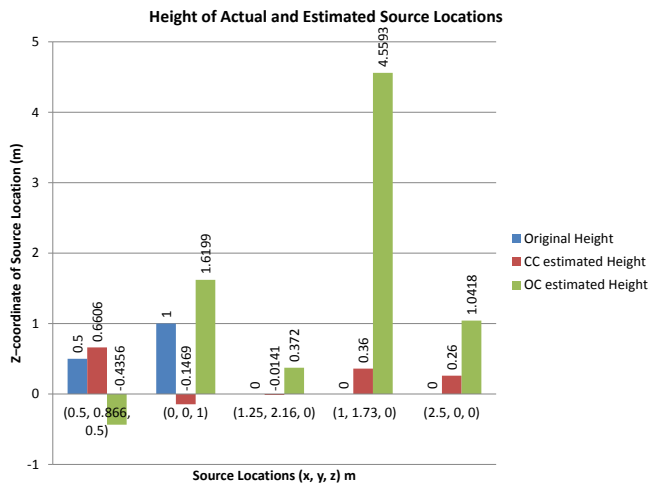
Next we plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.17a and Fig. 6.17b plots the x, y coordinates for the more-reverberant case. Fig. 6.17c and Fig. 6.17d plots the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.



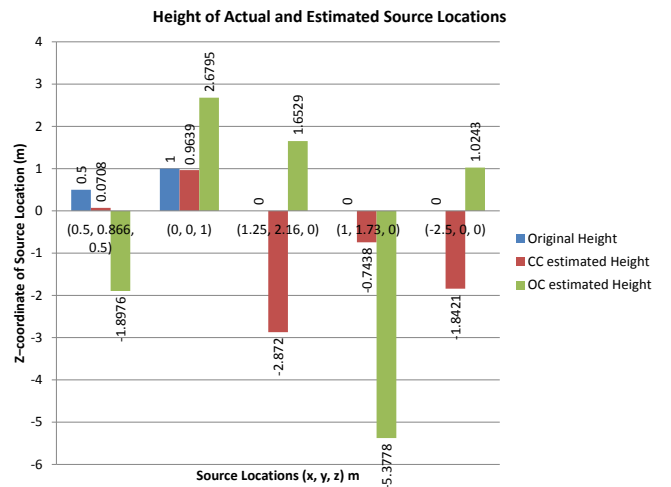
(a)



(b)



(c)



(d)

Figure 6.17: Pyramid Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.

Table 6.23: Pyramid indoor results using VocoPro System at 8kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.

Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	(1.25, 2.16, 0)	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	(-2.5, 0, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	Above the center of the array	On the line through a_2 and origin outside the array	In-between a_1 & a_2 on the circle inscribing the array.	Along the line (outside array) passing in-between a_2, a_3 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(0.0046, -0.0056, 0.0024, 0.0001)	(0, 0.0016, 0.0058, 0.0005)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0011, -0.0093, 0.0032, -0.0014)	(0.0011, 0.0051, 0.0034, -0.0014)	(-0.0003, -0.0316, -0.0003, 0.0014)	(0.0006, -0.0079, 0.0059, -0.0021)	(0.0066, -0.0014, 0.0051, 0.0039)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0026, -0.0025, 0.0050, 0.0019)	(0.0025, 0.0043, 0.0005, -0.0049)	(-0.0035, -0.0055, 0.0054, 0.0025)	(-0.0027, -0.0033, 0.0021, -0.0030)	(0.0059, -0.0010, 0.0020, -0.0010)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(1.0023, -1.5170, 0.0708)	(0.7497, -0.4332, 0.9639)	(1.1998, -2.0781, -2.8720)	(1.0667, -1.4845, -0.7438)	(-0.8566, 0.7411, -1.8421)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.1648, 1.9273, -1.8976)	(0.0393, -0.9889, 2.6795)	(-0.2585, -1.6453, 1.6529)	(-0.8356, -1.9845, -5.3778)	(-1.5875, 0.5143, 1.0243)
$\epsilon_{cc,3D}$	2.4729	1.4020	5.1198	3.3001	2.5775
$\epsilon_{cc,2D}$	2.4353	1.3230	4.2384	3.2152	1.8028
$\epsilon_{oc,3D}$	2.7050	2.8989	4.4145	6.7888	1.4651
$\epsilon_{oc,2D}$	1.2523	1.9113	4.0934	4.1433	1.0475
Time for CC (Sec)	0.0345	1.3950	0.0411	0.0605	0.0336
Time for OC (Sec)	52.7992	38.9396	31.9275	50.4358	47.8197

Results for 4kHz Sampling Rate

Table 6.24 shows the results for 4kHz when the array was placed in the center of the hall while Table 6.25 shows the results for 4kHz but this time the array was placed in a corner of the hall.

Next we plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.18a and Fig. 6.18b plots the x, y coordinates for the more-reverberant case. Fig. 6.18c and Fig. 6.18d plots the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.

Table 6.24: Pyramid indoor results using VocoPro System at 4kHz for the case when the array was placed in the center of the hall. Slow Machine was used.

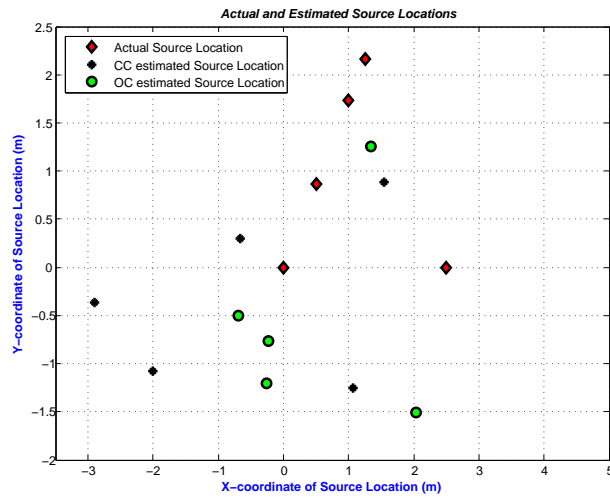
Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	$(2.5\cos60^\circ, 2.5\sin60^\circ, 0) = (1.25, 2.16, 0)$	$(2\cos60^\circ, 2\sin60^\circ, 0) = (1, 1.73, 0)$	(2.5, 0, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	Above the center of the array	In-between a_1 & a_2 , outside the array	In-between a_1 & a_2 on the circle inscribing the array.	Along the line (outside array) passing through a_1 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0005, 0.0122, 0.0070, -0.0022)	(0.0027, -0.0028, -0.0057, -0.0027)	(-0.0060, 0.0003, 0.0018, 0)	(0.0055, 0.0020, 0.0037, 0.0037)	(-0.0043, 0.0217, 0.0018, -0.0037)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0067, 0.0075, 0.0030, 0.0067)	(-0.0008, 0.0012, 0.0047, 0.0027)	(-0.0043, -0.0057, 0.0013, -0.0003)	(0.0065, 0.0050, 0.0018, -0.0057)	(0.0030, 0.0205, -0.0015, 0.0043)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(1.0619, -1.2506, -0.8738)	(1.5452, 0.8800, -2.3249)	(-2.9071, -0.3612, 0.3600)	(-0.6671, 0.3032, -2.2206)	(-2.0128, -1.0747, -6.8739)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.2260, -0.7607, -3.0456)	(1.3425, 1.2565, -3.6098)	(-0.2648, -1.2004, 0.2112)	(-0.6916, -0.4968, -1.1020)	(2.0313, -1.5068, 3.9919)
$\epsilon_{cc,3D}$	2.5852	3.7706	4.8752	3.1219	8.2928
$\epsilon_{cc,2D}$	2.1899	1.7782	4.8619	2.1943	4.6390
$\epsilon_{oc,3D}$	3.9680	4.9630	3.6921	3.0058	4.2925
$\epsilon_{oc,2D}$	1.7814	1.8388	3.6861	2.7965	1.5780
Time for CC (Sec)	0.0773	0.0445	0.0442	0.0440	0.0436
Time for OC (Sec)	15.5273	21.3783	20.4861	15.9091	16.9983

Table 6.25: Pyramid indoor results using VocoPro System at 4kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.

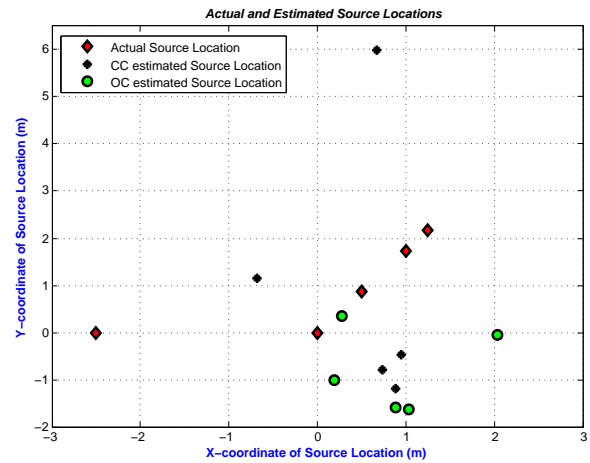
Source Location	(0.5, 0.866, 0.5)	(0, 0, 1)	(1.25, 2.16, 0)	$(2\cos60^\circ, 2\sin60^\circ, 0) = (1, 1.73, 0)$	(-2.5, 0, 0)
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	Above the center of the array	On the line through a_2 and origin outside the array	In-between a_1 & a_2 on the circle inscribing the array.	Along the line (outside array) passing in-between a_2, a_3 and center of the array
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(0.0046, -0.0056, 0.0024, 0.0001)	(0, 0.0000, 0.0059, 0.0004)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(-0.0047, 0.0020, 0.0123, -0.0015)	(0.0022, 0.0075, 0.0040, -0.0008)	(0.0018, 0.0085, 0.0065, -0.0053)	(0.0003, -0.0040, 0.0053, 0.0035)	(-0.0018, 0.0095, 0.0040, 0.0028)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0022, -0.0097, 0.0017, 0.0008)	(0.0035, 0.0063, 0.0013, 0.0070)	(0.0003, -0.0083, -0.0008, -0.0027)	(-0.0013, 0.0023, 0.0022, 0.0047)	(0.0050, 0.0063, 0.0063, 0)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-0.6794, 1.1405, -0.7435)	(0.7361, -0.7855, 0.4339)	(0.9495, -0.4729, -2.2479)	(0.6743, 5.9845, -11.0942)	(0.8811, -1.1943, -0.1602)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.8819, -1.5868, 0.3306)	(0.1887, -1.0112, -6.1138)	(1.0311, -1.6249, -1.0673)	(2.0311, -0.0555, -12.2592)	(0.2822, 0, 0.3600)
$\epsilon_{cc,3D}$	1.7357	1.2163	3.4750	11.8865	3.5894
$\epsilon_{cc,2D}$	1.2109	1.0765	2.6500	4.2670	3.5858
$\epsilon_{oc,3D}$	2.4882	7.1878	3.9386	12.4314	2.8054
$\epsilon_{oc,2D}$	2.4824	1.0286	3.7913	2.0618	2.7822
Time for CC (Sec)	0.0444	0.0433	0.0437	0.0438	0.0435
Time for OC (Sec)	15.9668	20.9052	17.2456	19.4981	16.0384

6.3.2 Circular Geometry Experiments and Results

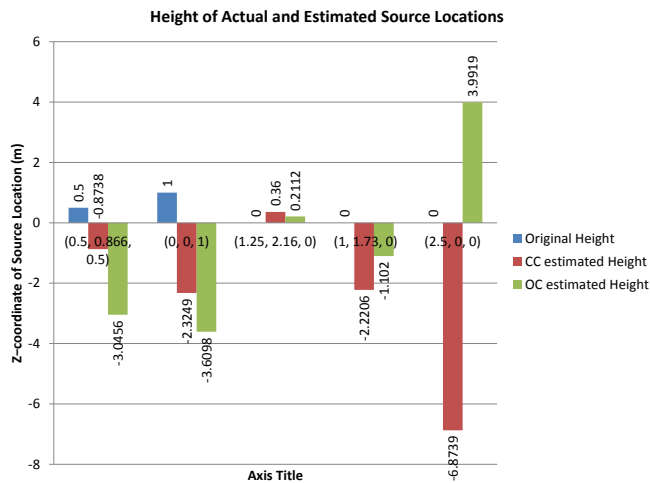
Now we will present the experiments we conducted and their results for the Circular geometry using VocoPro system. We conducted the experiments at 8kHz



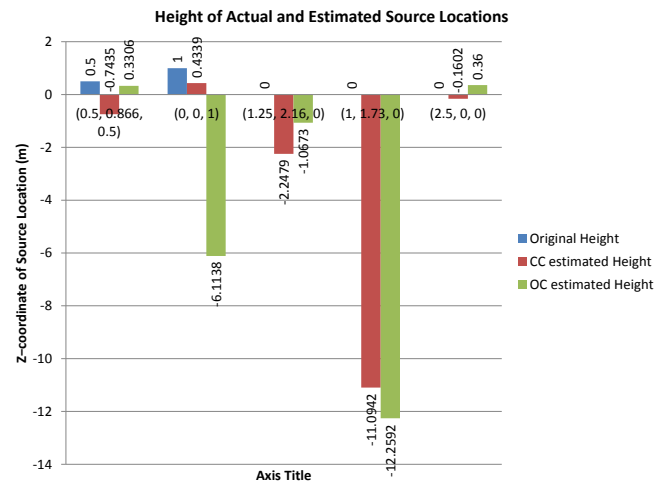
(a)



(b)



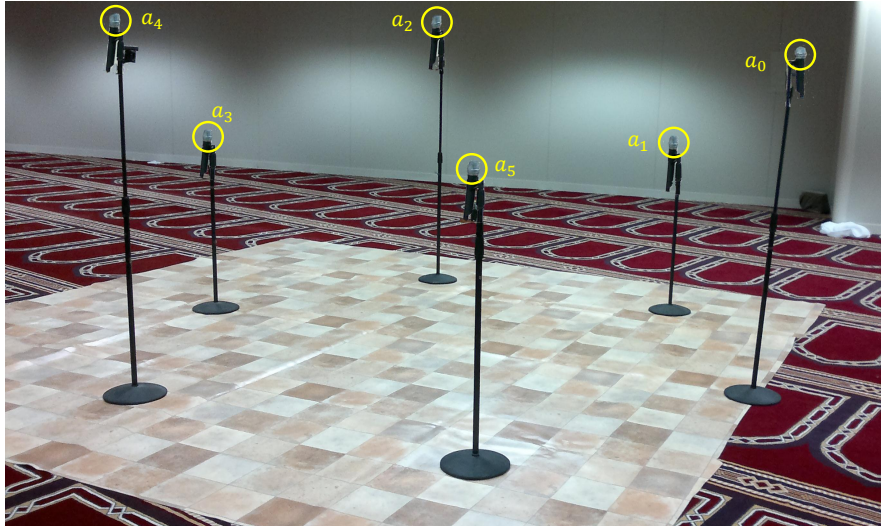
(c)



(d)

Figure 6.18: Pyramid Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.

and 4kHz sampling rate. Fig. 6.19 shows the experimental setup for Pyramid geometry using VocoPro system indoor. Fig. 6.19a shows the experimental setup for the case of array placed in the center of the hall while Fig. 6.19b shows the experimental setup for the case of array placed in a corner of the hall.



(a)



(b)

Figure 6.19: Indoor Circular Geometry Experimental Setup Using VocoPro System, (a) Indoor Circular Geometry in the Center of the Hall using VocoPro System, (b) Indoor Circular Geometry in a Corner of the Hall using VocoPro System.

For each sampling rate there are two different scenarios i) experimentation in the center of the hall and ii). experimentation in a corner of the hall. We divide the discussion further into two subsections w.r.t. to the sampling rate as follows.

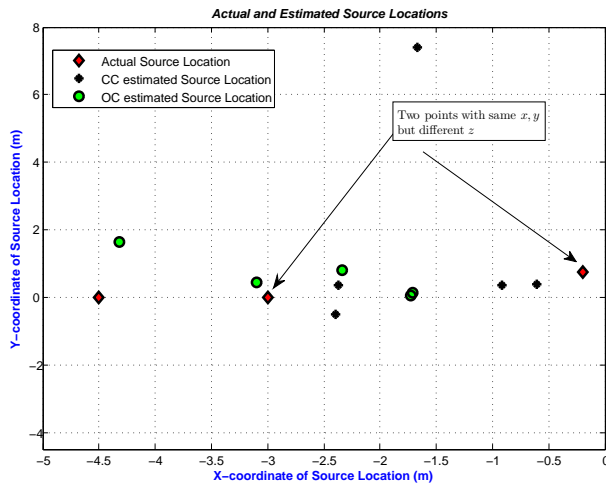
Results for 8kHz Sampling Rate

Table 6.26 shows the results for 8kHz when the array was placed in the center of the hall while Table 6.27 shows the results for 8kHz when the array was placed in a corner of the hall.

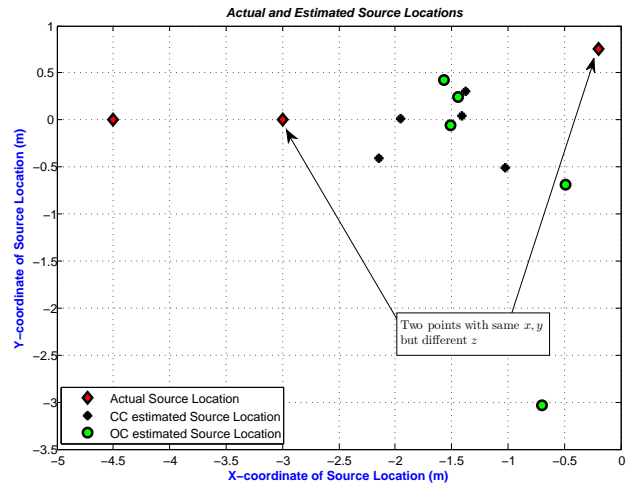
Table 6.26: Circular indoor results using VocoPro System at 8kHz for the case when the array was placed in the center of the hall. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs $(\tau_{10}^c, \tau_{20}^c, \tau_{30}^c, \tau_{40}^c)$	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0080, -0.0005, 0.0187, 0.0213, 0.0135)	(0.0415, 0.0001, 0.0150, -0.0065, -0.0050)	(-0.0076, -0.0049, 0.0030, 0.0051, 0.0001)	(0.0085, -0.0043, -0.0074, 0.0001, -0.0155)	(-0.0018, -0.0206, -0.0040, -0.0182, -0.0020)
TDE_{OC} $(\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc})$	(0.0139, 0.0039, 0.0226, 0.0279, 0.0119)	(0.0035, -0.0069, -0.0146, -0.0141, -0.0203)	(-0.0086, -0.0077, -0.0095, -0.0016, -0.0100)	(0.0073, 0.0081, 0.0018, 0.0070, 0.0090)	(-0.0009, -0.0098, -0.0037, -0.0053, 0.0035)
X_{CC} $(x_s^{cc}, y_s^{cc}, z_s^{cc})$	(-0.6085, 0.3906, 10.3074)	(-1.6646, 7.4002, 10.4321)	(-0.9197, 0.3618, 0.2642)	(-2.3683, 0.3564, -13.1643)	(-2.3994, -0.5120, 3.0382)
X_{OC} $(x_s^{oc}, y_s^{oc}, z_s^{oc})$	(-2.3367, 0.8083, 19.2068)	(-3.0968, 0.4411, -9.1498)	(-1.7063, 0.1253, -0.7380)	(-1.7280, 0.0656, 0.4371)	(-4.3159, 1.6332, -8.2502)
$\epsilon_{cc,3D}$	10.3217	11.6334	2.1280	13.6835	3.7290
$\epsilon_{cc,2D}$	0.5441	6.8095	2.1115	0.7253	2.1621
$\epsilon_{oc,3D}$	19.3253	10.5596	1.4947	1.2753	8.4124
$\epsilon_{oc,2D}$	2.1375	2.9133	1.2998	1.2737	1.6436
Time for CC (Sec)	0.1326	0.2568	0.1437	0.1364	0.1340
Time for OC (Sec)	34.5292	37.2538	35.9664	38.5995	39.6390

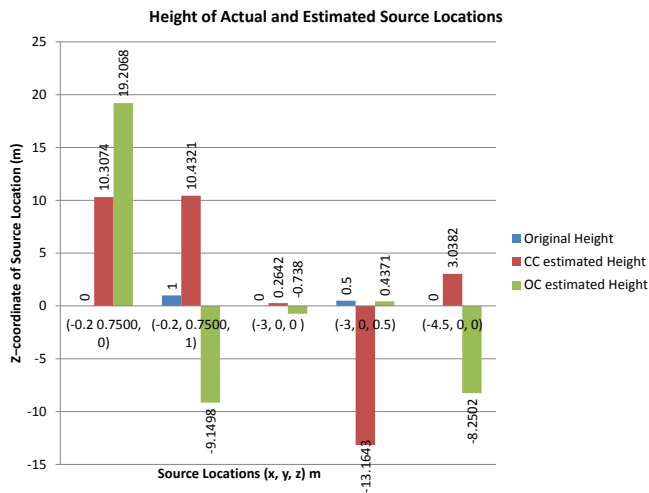
Next we plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.20a and Fig. 6.20b plots the x, y coordinates for the more-reverberant case. Fig. 6.20c and Fig. 6.20d plots the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.



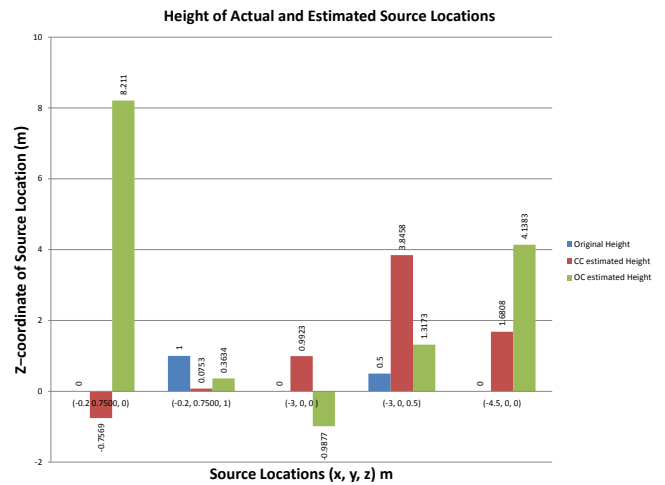
(a)



(b)



(c)



(d)

Figure 6.20: Circular Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case

Table 6.27: Circular indoor results using VocoPro System at 8kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs $(\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40})$	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0043, 0.0009, 0.0067, 0.0056, -0.0025)	(0.0005, 0.0025, 0.0031, 0.0077, 0.0005)	(-0.0014, -0.0054, -0.0008, 0.0010, -0.0037)	(0.0005, -0.0150, -0.0036, 0.0019, -0.0094)	(-0.0016, -0.0115, -0.0092, -0.0054, -0.0034)
TDE_{OC} $(\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc})$	(-0.0174, -0.0248, -0.0181, -0.0133, -0.0081)	(-0.0036, 0.0003, -0.0030, -0.0030, -0.0009)	(0.0022, -0.0011, 0.0027, -0.0031, 0.0043)	(-0.0049, -0.0027, -0.0018, 0.0010, -0.0005)	(-0.0064, -0.0145, -0.0120, -0.0105, -0.0050)
X_{CC} $(x_s^{cc}, y_s^{cc}, z_s^{cc})$	(-1.3727, 0.3006, -0.7569)	(-1.9563, 0.0145, 0.0753)	(-1.4084, 0.0421, 0.9923)	(-1.0281, -0.5055, 3.8458)	(-2.1488, -0.4130, 1.6808)
X_{OC} $(x_s^{oc}, y_s^{oc}, z_s^{oc})$	(-0.6978, -3.0374, 8.2110)	(-1.5110, -0.0580, 0.3634)	(-1.4462, 0.2392, -0.9877)	(-1.5661, 0.4255, 1.3173)	(-0.4923, -0.6927, 4.1383)
$\epsilon_{cc,3D}$	1.4663	2.1168	1.8760	3.9164	2.9195
$\epsilon_{cc,2D}$	1.2558	1.9041	1.5921	2.0356	2.3872
$\epsilon_{oc,3D}$	9.0561	1.6665	1.8566	1.7044	5.8023
$\epsilon_{oc,2D}$	3.8200	1.5401	1.5721	1.4957	4.0672
Time for CC (Sec)	0.1373	0.1368	0.1373	0.3102	0.6309
Time for OC (Sec)	41.3691	39.6792	47.3056	48.1760	44.0392

Results for 4kHz Sampling Rate

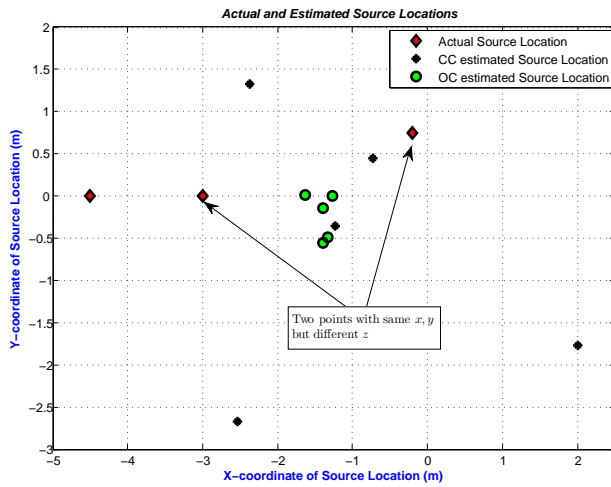
Table 6.28 shows the results for 4kHz when the array was placed in the center of the hall while Table 6.29 shows the results for 4kHz when the array was placed in a corner of the hall. Next we plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.21a and Fig. 6.21b plots the x, y coordinates for the more-reverberant case. Fig. 6.21c and Fig. 6.21d plots the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.

Table 6.28: Circular indoor results using VocoPro System at 4kHz for the case when the array was placed in the center of the hall. Slow Machine was used.

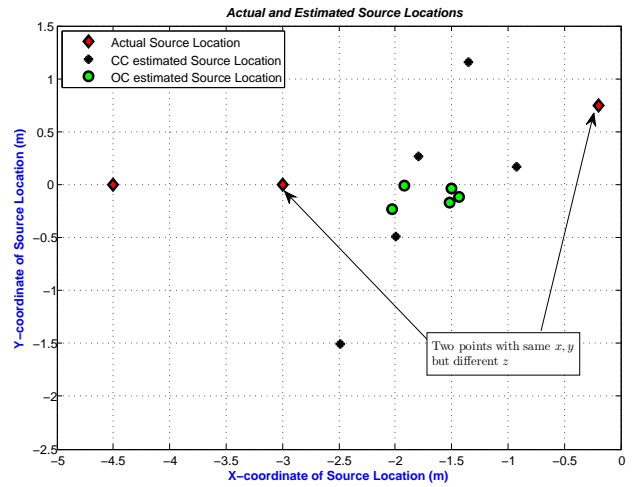
Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0138, -0.0005, 0.0275, 0.0143, 0.0135)	(0.0073, -0.0018, 0.0150, 0.0092, -0.0035)	(-0.0177, 0.0090, -0.0187, 0.0088, -0.0103)	(0.0027, -0.0165, -0.0013, 0.0020, -0.0155)	(-0.0018, -0.0063, -0.0040, -0.0210, -0.0190)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(-0.0065, -0.0103, 0, -0.0027, -0.0045)	(0.0060, -0.0053, 0.0005, -0.0035, -0.0035)	(0.0053, 0.0022, 0.0005, 0.0015, -0.0005)	(-0.0013, 0.0032, -0.0018, -0.0035, -0.0008)	(0.0040, -0.0020, -0.0025, 0.0020, -0.0010)
X_{CC} ($x_{10}^{oc}, y_{10}^{oc}, z_{10}^{oc}$)	(-2.5373, -2.6683, 17.1077)	(-0.7246, 0.4459, -2.5209)	(2.0021, -1.7679, -5.1883)	(-1.2263, -0.3539, -0.4450)	(-2.3751, 1.3301, 2.0626)
X_{OC} ($x_{10}^{oc}, y_{10}^{oc}, z_{10}^{oc}$)	(-1.3268, -0.4936, 1.7133)	(-1.3995, -0.5600, -1.4857)	(-1.6317, 0.0083, -0.2654)	(-1.2684, -0.0024, 0.7657)	(-1.3953, -0.1486, -0.3003)
$\epsilon_{cc,3D}$	17.6017	3.5728	7.4206	2.0407	3.2463
$\epsilon_{cc,2D}$	4.1410	0.6064	5.3053	1.8087	2.5069
$\epsilon_{oc,3D}$	2.3983	3.0551	1.3939	1.7518	3.1227
$\epsilon_{oc,2D}$	1.6781	1.7762	1.3684	1.7316	3.1082
Time for CC (Sec)	0.0469	0.0464	0.0544	0.0463	0.0464
Time for OC (Sec)	20.6177	20.5589	23.5458	19.9169	19.9744

Table 6.29: Circular indoor results using VocoPro System at 4kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.

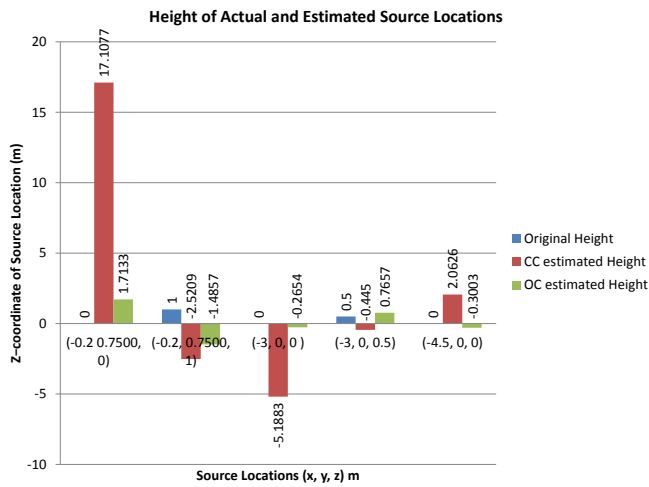
Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(-0.0003, 0.0010, 0.0053, 0.0133, 0.0083)	(0.0010, 0.0040, 0.0037, 0.0107, 0.0088)	(-0.0065, -0.0123, -0.0008, 0.0022, -0.0032)	(-0.0032, -0.0047, -0.0083, 0.0032, -0.0113)	(-0.0085, -0.0182, -0.0150, -0.0037, -0.0098)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0003, -0.0025, -0.0018, -0.0020, -0.0013)	(0.0105, 0.0067, 0.0050, 0.0080, 0.0065)	(-0.0003, -0.0060, -0.0005, -0.0032, -0.0015)	(-0.0103, -0.0083, -0.0075, -0.0107, -0.0060)	(-0.0025, -0.0065, -0.0050, -0.0047, -0.0030)
X_{CC} ($x_{10}^{oc}, y_{10}^{oc}, z_{10}^{oc}$)	(-2.4871, -1.5054, 2.8999)	(-1.7952, 0.2691, 1.2486)	(-0.9251, 0.1646, 4.2830)	(-1.3503, 1.1569, -0.2270)	(-1.9940, -0.4931, 8.0051)
X_{OC} ($x_{10}^{oc}, y_{10}^{oc}, z_{10}^{oc}$)	(-1.5003, -0.0329, 0.3026)	(-2.0237, -0.2316, 0.6986)	(-1.5202, -0.1695, 0.5537)	(-1.9196, -0.0118, 2.1073)	(-1.4343, -0.1125, 1.1040)
$\epsilon_{cc,3D}$	4.3275	1.6845	4.7620	2.1421	8.4026
$\epsilon_{cc,2D}$	3.2121	1.6661	2.0815	2.0149	2.5541
$\epsilon_{oc,3D}$	1.5476	2.0930	1.5891	1.9367	3.2604
$\epsilon_{oc,2D}$	1.5178	2.0711	1.4895	1.0804	3.0677
Time for CC (Sec)	1.5178	0.0468	0.0470	0.0473	0.0462
Time for OC (Sec)	18.2064	21.4984	21.2012	16.9659	26.7784



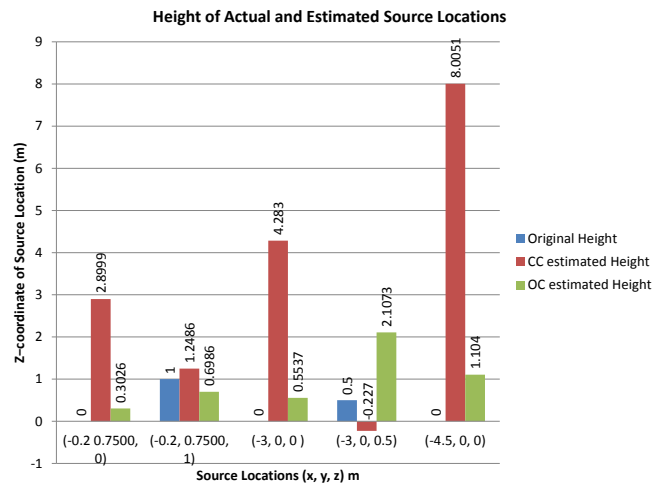
(a)



(b)



(c)

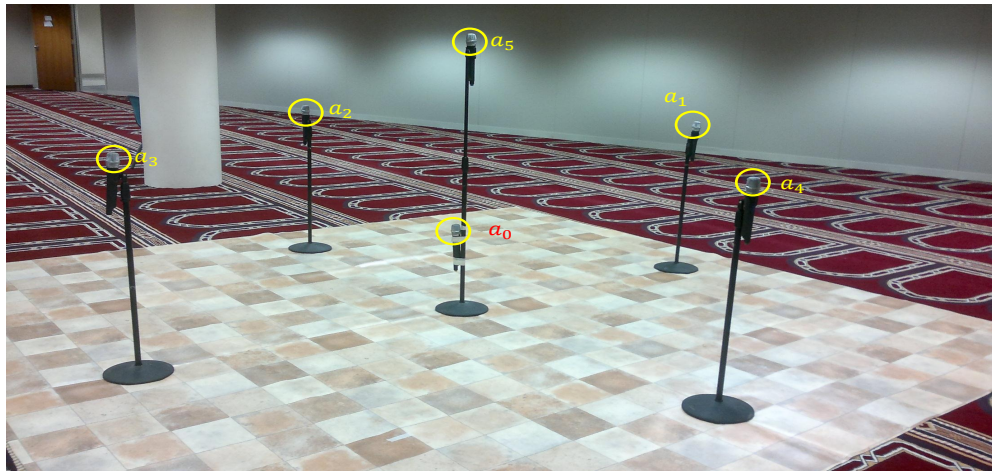


(d)

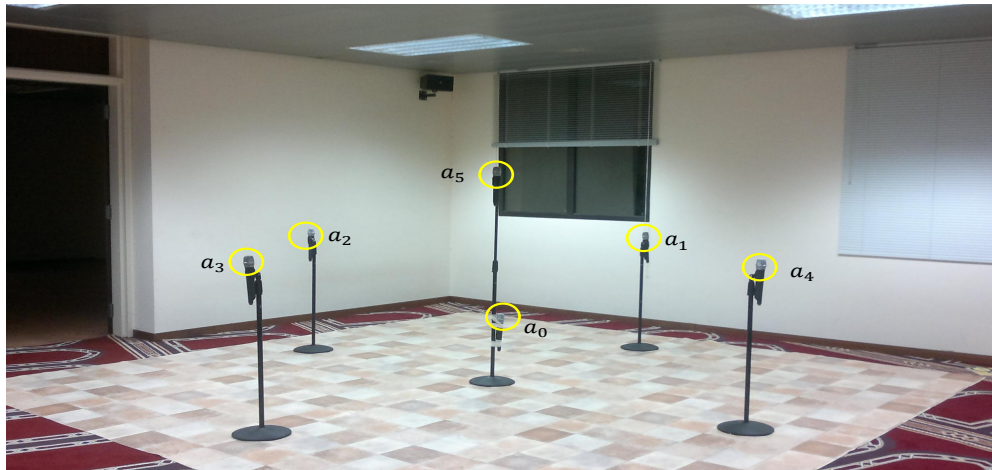
Figure 6.21: Circular Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case.

6.3.3 Rhombus Geometry Experiments and Results

In this section we will discuss the experiments we conducted and their results for the Rhombus geometry using VocoPro system. We conducted the experiments at 8kHz and 4kHz sampling rate. Fig. 6.22 shows the experimental setup for Rhombus geometry using VocoPro system indoor.



(a)



(b)

Figure 6.22: Indoor Rhombus Geometry Experimental Setup Using VocoPro System, (a) Indoor Rhombus Geometry in the Center of the Hall using VocoPro System, (b) Indoor Rhombus Geometry in a Corner of the Hall using VocoPro System.

Fig. 6.22a shows the experimental setup for the case of array placed in the center of the hall while Fig. 6.22b shows the experimental setup for the case of array placed in a corner of the hall.

For each sampling rate there are two different scenarios i) experimentation in the center of the hall and ii). experimentation in a corner of the hall. We divide the discussion further into two subsections w.r.t. to the sampling rate as follows.

Results for 8kHz Sampling Rate

Table 6.30 shows the results for 8kHz when the array was placed in the center of the hall while Table 6.31 shows the results for 8kHz when the array was placed in a corner of the hall.

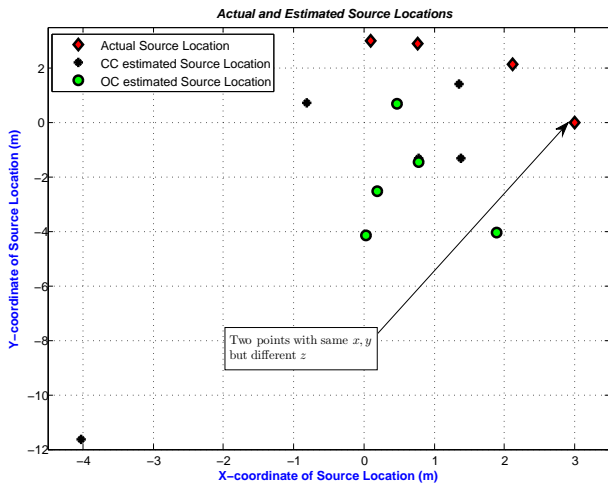
Table 6.30: Rhombus indoor results using VocoPro System at 8kHz for the case when the array was placed in the center of the hall. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0080, -0.0005, 0.0187, 0.0213, 0.0135)	(0.0415, 0.0001, 0.0150, -0.0065, -0.0050)	(-0.0076, -0.0049, 0.0030, 0.0051, 0.0001)	(0.0085, -0.0043, -0.0074, 0.0001, -0.0155)	(-0.0018, -0.0206, -0.0040, -0.0182, -0.0020)
TDE_{OC} $(\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc})$	(0.0139, 0.0039, 0.0226, 0.0279, 0.0119)	(0.0035, -0.0069, -0.0146, -0.0141, -0.0203)	(-0.0086, -0.0077, -0.0095, -0.0016, -0.0100)	(0.0073, 0.0081, 0.0018, 0.0070, 0.0090)	(-0.0009, -0.0098, -0.0037, -0.0053, 0.0035)
X_{CC} $(x_s^{cc}, y_s^{cc}, z_s^{cc})$	(-0.6085, 0.3906, 10.3074)	(-1.6646, 7.4002, 10.4321)	(-0.9197, 0.3618, 0.2642)	(-2.3683, 0.3564, -13.1643)	(-2.3994, -0.5120, 3.0382)
X_{OC} $(x_s^{oc}, y_s^{oc}, z_s^{oc})$	(-2.3367, 0.8083, 19.2068)	(-3.0968, 0.4411, -9.1498)	(-1.7063, 0.1253, -0.7380)	(-1.7280, 0.0656, 0.4371)	(-4.3159, 1.6332, -8.2502)
$\epsilon_{cc,3D}$	10.3217	11.6334	2.1280	13.6835	3.7290
$\epsilon_{cc,2D}$	0.5441	6.8095	2.1115	0.7253	2.1621
$\epsilon_{oc,3D}$	19.3253	10.5596	1.4947	1.2753	8.4124
$\epsilon_{oc,2D}$	2.1375	2.9133	1.2998	1.2737	1.6436
Time for CC (Sec)	0.1326	0.2568	0.1437	0.1364	0.1340
Time for OC (Sec)	34.5292	37.2538	35.9664	38.5995	39.6390

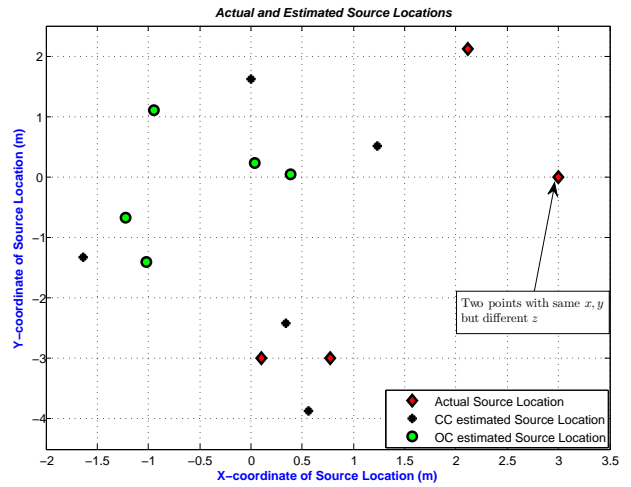
Table 6.31: Rhombus indoor results using VocoPro System at 8kHz for the case when the array was placed in a corner of the hall. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 on the circumference at height of 0m	In-between mic0 and 1 on the circumference at height of 1m	At mic3 with zero height	At mic3 with 0.5m height	Along the line (outside array) passing through mic3 and center of the array at height 0m
Exact TDs $(\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40})$	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0043, 0.0009, 0.0067, 0.0056, -0.0025)	(0.0005, 0.0025, 0.0031, 0.0077, 0.0005)	(-0.0014, -0.0054, -0.0008, 0.0010, -0.0037)	(0.0005, -0.0150, -0.0036, 0.0019, -0.0094)	(-0.0016, -0.0115, -0.0092, -0.0054, -0.0034)
TDE_{OC} $(\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc})$	(-0.0174, -0.0248, -0.0181, -0.0133, -0.0081)	(-0.0036, 0.0003, -0.0030, -0.0030, -0.0009)	(0.0022, -0.0011, 0.0027, -0.0031, 0.0043)	(-0.0049, -0.0027, -0.0018, 0.0010, -0.0005)	(-0.0064, -0.0145, -0.0120, -0.0105, -0.0050)
X_{CC} $(x_s^{cc}, y_s^{cc}, z_s^{cc})$	(-1.3727, 0.3006, -0.7569)	(-1.9563, 0.0145, 0.0753)	(-1.4084, 0.0421, 0.9923)	(-1.0281, -0.5055, 3.8458)	(-2.1488, -0.4130, 1.6808)
X_{OC} $(x_s^{oc}, y_s^{oc}, z_s^{oc})$	(-0.6978, -3.0374, 8.2110)	(-1.5110, -0.0580, 0.3634)	(-1.4462, 0.2392, -0.9877)	(-1.5661, 0.4255, 1.3173)	(-0.4923, -0.6927, 4.1383)
$\epsilon_{cc,3D}$	1.4663	2.1168	1.8760	3.9164	2.9195
$\epsilon_{cc,2D}$	1.2558	1.9041	1.5921	2.0356	2.3872
$\epsilon_{oc,3D}$	9.0561	1.6665	1.8566	1.7044	5.8023
$\epsilon_{oc,2D}$	3.8200	1.5401	1.5721	1.4957	4.0672
Time for CC (Sec)	0.1373	0.1368	0.1373	0.3102	0.6309
Time for OC (Sec)	41.3691	39.6792	47.3056	48.1760	44.0392

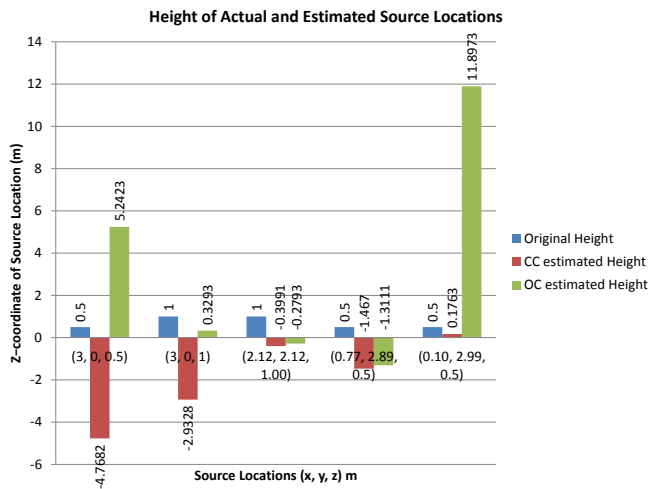
Next we plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.23a and Fig. 6.23b plots the x, y coordinates for the more-reverberant case. Fig. 6.23c and Fig. 6.23d plots the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.



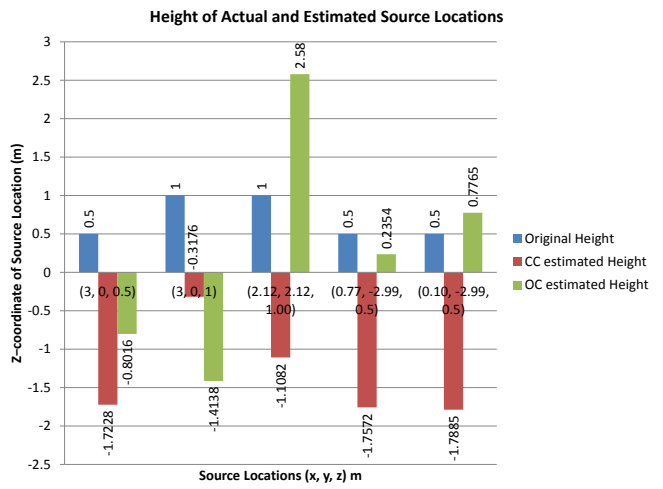
(a)



(b)



(c)



(d)

Figure 6.23: Rhombus Geometry Source Locations Estimated by CC and OC at 8 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case

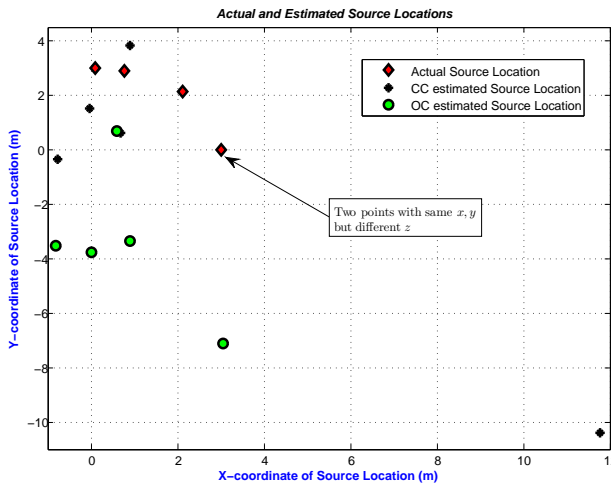
Results for 4kHz Sampling Rate

Table 6.32 shows the results for 4kHz when the array was placed in the center of the hall while Table 6.33 shows the results for 4kHz when the array was placed in a corner of the hall.

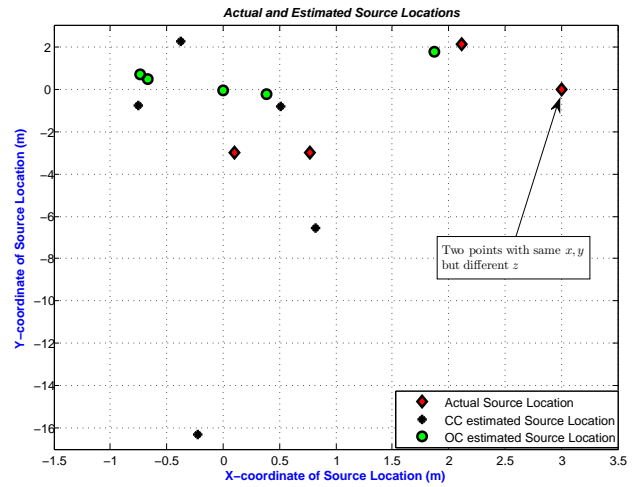
Next we plot the x, y coordinates of the actual and estimated source locations for the less-reverberant case in Fig. 6.24a and Fig. 6.24b plots the x, y coordinates for the more-reverberant case. Fig. 6.24c and Fig. 6.24d plots the bargraphs for the height of source locations for the case of less- and more-reverberant scenarios respectively.

Table 6.32: Rhombus indoor results using VocoPro System at 4kHz for the case when the array was placed in the center of the hall. Slow Machine was used.

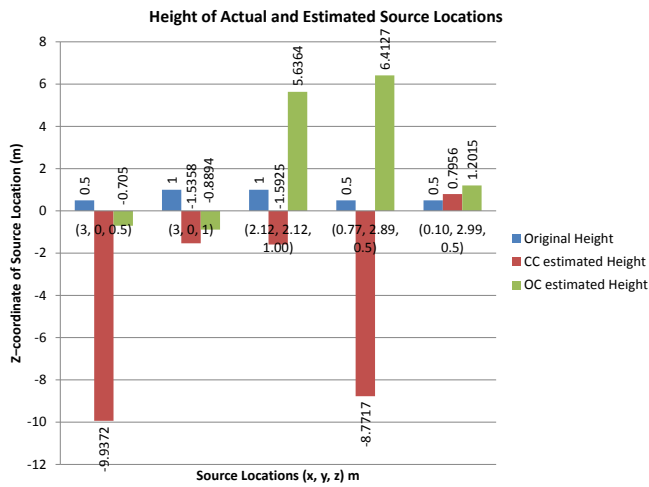
Source Location	(2r, 0, h)	(2r, 0, 2h)	$(2r \cos 45^\circ, 2r \sin 45^\circ, 2h)$	$(2r \cos 75^\circ, 2r \sin 75^\circ, h)$	$(2r \cos 88^\circ, 2r \sin 88^\circ, h)$
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, 2.99, 0.5)
Description	On the line passing through mic1 and center of the array at height of 0.5m	On the line passing through mic1 and center of the array at height of 1m	In-between mic1 and 2 outside the circumference on which mics lie.	At angle of 75° between mic1 and 2	At angle of 88° between mic1 and 2
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE_{CC} ($\epsilon_{10}^{cc}, \epsilon_{20}^{cc}, \epsilon_{30}^{cc}, \epsilon_{40}^{cc}$)	(-0.0018, -0.0071, 0.0061, 0.0011, 0.0016)	(-0.0003, -0.0065, -0.0005, 0.0057, -0.0103)	(0.0075, -0.0065, -0.0032, 0.0047, -0.0057)	(-0.0005, 0.0037, -0.0060, 0.0127, -0.0138)	(0.0008, 0.0003, -0.0037, 0.0053, 0.0035)
TDE_{OC} ($\epsilon_{10}^{oc}, \epsilon_{20}^{oc}, \epsilon_{30}^{oc}, \epsilon_{40}^{oc}$)	(0.0083, -0.0103, -0.0022, 0.0013, -0.0015)	(0.0008, -0.0080, -0.0057, 0.0175, 0.0013)	(0.0077, -0.0030, 0.0025, 0.0158, 0.0043)	(0.0030, -0.0030, 0.0030, 0.0035, 0.0138)	(0.0010, 0.0013, -0.0027, 0.0065, 0.0025)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(11.7638, -10.4075, -9.9372)	(-0.0224, 1.5024, -1.5358)	(-0.7715, -0.3331, -1.5925)	(0.9106, 3.8096, -8.7717)	(0.6780, 0.6066, 0.7956)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.8176, -3.5090, -0.7050)	(3.0570, -7.1134, -0.8894)	(0.9054, -3.3654, 5.6364)	(0, -3.7682, 6.4127)	(0.6080, 0.6820, 1.2015)
$\epsilon_{cc,3D}$	12.9342	4.2217	4.5934	9.3182	2.4703
$\epsilon_{cc,2D}$	15.2357	3.3752	3.7919	0.9303	2.4525
$\epsilon_{oc,3D}$	5.3234	7.3603	7.2843	8.9378	2.4651
$\epsilon_{oc,2D}$	5.1853	7.1137	5.6182	6.7026	2.3632
Time for CC (Sec)	0.3846	0.2279	0.0690	0.0451	0.0768
Time for OC (Sec)	63.4931	27.2423	24.4149	118.6396	27.8298



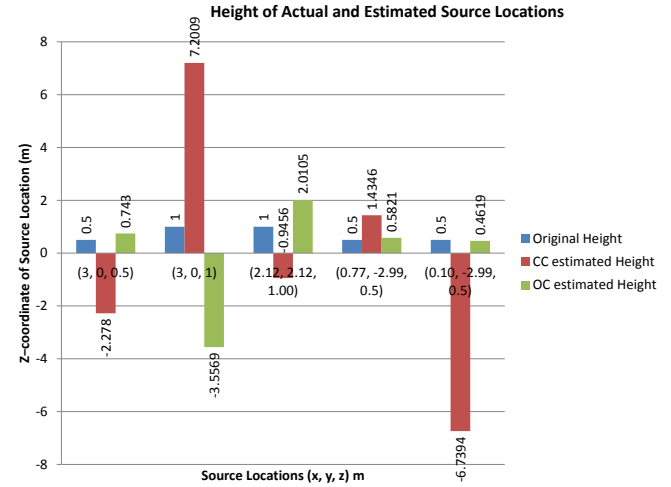
(a)



(b)



(c)



(d)

Figure 6.24: Rhombus Geometry Source Locations Estimated by CC and OC at 4 kHz Indoor using VocoPro system, (a) Source Locations estimated in the less-reverberant case, only the x, y coordinates are shown, (b) Source Locations estimated in the more-reverberant case, only the x, y coordinates are shown, (c) Height Estimate for the less-reverberant case, (d) Height Estimate for the more-reverberant case

Table 6.33: Rhombus indoor results using VocoPro System at 4kHz for the case when the array was placed in a corner of the hall. Slow Machine was used.

Source Location	$(2r, 0, h)$	$(2r, 0, 2h)$	$(2r \cos 45^\circ, 2r \sin 45^\circ, 2h)$	$(2r \cos 75^\circ, 2r \sin(\pi + 75^\circ), h)$	$(2r \cos 88^\circ, 2r \sin(\pi + 88^\circ), h)$
Value	$(3, 0, 0.5)$	$(3, 0, 1)$	$(2.12, 2.12, 1.00)$	$(0.77, -2.99, 0.5)$	$(0.10, -2.99, 0.5)$
Description	On the line passing through mic1 and center of the array at height of 0.5m	On the line passing through mic1 and center of the array at height of 1m	In-between mic1 and 2 outside the circumference on which mics lie.	At angle of 255° between mic3 and 4	At angle of 268° between mic3 and 4
Exact TDs $(\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40})$	$(-0.0045, 0.0009, 0.0043, 0)$	$(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)$	$(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)$	$(-0.0002, 0.0042, 0.0019, -0.0015, 0)$	$(0.0008, 0.0043, 0.0011, -0.0016, 0)$
TDE_{CC} $(\tau_{10}^{CC}, \tau_{20}^{CC}, \tau_{30}^{CC}, \tau_{40}^{CC})$	$(-0.0113, 0.0050, -0.0085, -0.0018, 0.0055)$	$(0.0055, 0.0125, 0.0095, 0.0190, 0.0060)$	$(-0.0027, -0.0110, 0.0135, 0.0022, 0.0043)$	$(0.0013, 0.0110, -0.0018, 0.0032, 0.0015)$	$(-0.0027, -0.0150, -0.0022, 0.0055, -0.0060)$
TDE_{OC} $(\tau_{10}^{OC}, \tau_{20}^{OC}, \tau_{30}^{OC}, \tau_{40}^{OC})$	$(0.0043, -0.0018, -0.0015, 0.0005, -0.0013)$	$(0.0013, 0.0020, -0.0113, 0.0065, 0.0070)$	$(0.0035, -0.0012, 0.0050, 0.0090, 0.0030)$	$(0, -0.0053, -0.0047, -0.0025, 0.0003)$	$(0.0043, 0.0008, -0.0005, -0.0003, 0.0013)$
X_{CC} $(x_s^{CC}, y_s^{CC}, z_s^{CC})$	$(-0.7541, -0.7443, -2.2780)$	$(-0.3730, 2.2572, 7.2009)$	$(0.8201, -6.5767, -0.9456)$	$(0.5112, -0.7912, 1.4346)$	$(-0.2282, -16.3172, -6.7394)$
X_{OC} $(x_s^{OC}, y_s^{OC}, z_s^{OC})$	$(-0.7384, 0.7084, 0.7430)$	$(1.8774, 1.7742, -3.5569)$	$(0.0001, -0.0609, 2.0105)$	$(0.3849, -0.2167, 0.5821)$	$(-0.6696, 0.4858, 0.4619)$
$\epsilon_{cc,3D}$	4.7291	7.4110	9.0060	2.3120	15.1701
$\epsilon_{cc,2D}$	3.8272	4.0586	8.7934	2.1146	13.3313
$\epsilon_{oc,3D}$	3.8127	5.0173	3.2049	2.7022	3.5602
$\epsilon_{oc,2D}$	3.8050	2.0995	3.0414	2.7009	3.5599
Time for CC (Sec)	0.0448	0.0438	0.0365	0.0354	0.0364
Time for OC (Sec)	24.3726	25.3826	26.8228	33.1654	25.2192

6.3.4 Observations for the Experiments using VocoPro System

Looking at the results in subsections 6.3.1 to 6.3.3 we can find the numerical facts about the results for the VocoPro system. We present these numerical facts in subsection 6.3.4 and then based on these numerical facts we deduce the observations and present them in subsection 6.3.4

Numerical Facts

1. For the Pyramid 8kHz, less-reverberant case: 80% of the time CC produces more correct results than OC by a factor of at least 1.27 i.e. CC 3D-MSE (ϵ_{CC_3D}) is at least 1.27 times less than OC 3D-MSE (ϵ_{OC_3D}) for 80% of the source locations. The runtime for OC is at least 1160 times more than the runtime for CC.
2. For the Pyramid 8kHz, more-reverberant case: 80% of the time CC produces more correct results than OC by a factor of at least 1.09. The runtime for OC is at least 750 times more than the runtime for CC.
3. For the Pyramid 4kHz, less-reverberant case: 60% of the time OC produces more correct results than CC by a factor of at least 1.04. The runtime for OC is at least 200 times more than the runtime for CC.
4. For the Pyramid 4kHz, more-reverberant case: 80% of the time CC produces more correct results than OC by a factor of at least 1.13. The runtime for OC is at least 300 times more than the runtime for CC.
5. For the Circular 8kHz, less-reverberant case: 60% of the time OC produces more correct results than CC by a factor of at least 1.102. The runtime for OC is at least 260 times more than the runtime for CC.
6. For the Circular 8kHz, more-reverberant case: 60% of the time OC produces more correct results than CC by a factor of at least 1.01. The runtime for OC is at least 280 times more than the runtime for CC.

7. For the Circular 4kHz, less-reverberant case: 100% of the time OC produces more correct results than CC by a factor of at least 1.04. The runtime for OC is at least 430 times more than the runtime for CC.
8. For the Circular 4kHz, more-reverberant case: 80% of the time OC produces more correct results than CC by a factor of at least 1.106. The runtime for OC is at least 350 times more than the runtime for CC.
9. For the Rhombus 8kHz, less-reverberant case: 60% of the time OC produces more correct results than CC by a factor of at least 1.102. The runtime for OC is at least 260 times more than the runtime for CC.
10. For the Rhombus 8kHz, more-reverberant case: 60% of the time OC produces more correct results than CC by a factor of at least 1.01. The runtime for OC is at least 300 times more than the runtime for CC.
11. For the Rhombus 4kHz, less-reverberant case: 60% of the time CC produces more correct results than OC by a factor of at least 1.004. The runtime for OC is at least 350 times more than the runtime for CC.
12. For the Rhombus 4kHz, more-reverberant case: 80% of the time OC produces more correct results than CC by a factor of at least 1.24. The runtime for OC is at least 540 times more than the runtime for CC.

Indoor VocoPro Experiments Observations

Looking at the numerical facts in subsection 6.3.4 we can make the following observations for the experiments carried out indoor using VocoPro system.

1. 70% of the time CC produces better results for the Pyramid geometry for the less reverberant case by a factor of at least 1.04 while 80% times for the more-reverberant case by a factor of at least 1.09 than the OC. The OC still takes more time than CC, at least 200 times more than CC both for less- and more-reverberant cases.
2. 80% of the time OC produces better results for the Circular geometry for the less reverberant case by a factor of at least 1.04 while 70% times for the more-reverberant case by a factor of at least 1.01 than the CC. But this accuracy comes at the cost of at least 260 times more runtime taken by OC than CC both for the less- and more-reverberant environments.
3. 60% of the time OC produces better results for the Rhombus geometry for the less reverberant case by a factor of at least 1.004 while at least 70% of the time for the more-reverberant case by a factor of at least 1.01 than the CC. But this accuracy comes at the cost of at least 260 times more runtime taken by OC than CC for the less-reverberant case and at least 300 times more runtime by OC than CC for the more-reverberant case.
4. The Rhombus geometry produces the least accurate results with a minimum $\epsilon_{OC,3D}$ of value 1.27m among the three geometries while the Pyramid ge-

ometry produces the most accurate results with a minimum $\epsilon_{OC.3D}$ of value 0.68m. The Circular is in-between Rhombus and Pyramid, accuracy-wise, with a minimum $\epsilon_{OC.3D}$ of value 1.25m.

5. The 8kHz case is taking at least 2 times more runtime for OC than 4kHz case for all the geometries
6. The OC algorithm is taking at least 200 times more runtime than CC for all the cases of indoor experimentation using VocoPro.
7. At least 70% of the time the OC produces more accurate results than CC in the less- and more-reverberant environment for all the indoor experimental scenarios using VocoPro system.

6.4 Conclusion

An extensive experimentation and analysis was carried out for the indoor 3D geometries using two different hardware systems in this chapter. Two different sampling rate i.e. 4 kHz and 8 kHz were used during experimentation to compare the results of the OC algorithm with the classical CC TDE techniques. Several scenarios for the experimentation were considered which may be encountered during practical situations. Three different geometries were considered to see the effect of the geometry on the performance of the OC algorithm. Further more, in Pyramid geometry five sensors were used while in the remaining two geometries six sensors were used. A parametric analysis of the OC algorithm was also

conducted to see the variation in the results of the algorithm due to variation in various parameters. Specifically it was shown that the OC algorithm is sensitive to the signal shape.

It was observed that the Pyramid geometry produces best results among the three geometries with a minimum 3D-MSE OC error of 0.21m when using the RevoLabs system . This verifies that the structure of the geometry is more important than the number of sensors. The Rhombus geometry provides the least accurate results with a minimum 3D-MSE OC error of 1.25m. The OC algorithm was producing at least 70% of the times better results than CC for the RevoLabs system with a runtime tradeoff of up to 90 times more than that of CC. With VocoPro system the OC algorithm was producing at least 70% of the times better results than CC with a runtime tradeoff of up to 200 times more than that of CC.

The VocoPro system produces at least four times erroneous results compared to the RevoLabs system. Thus the signal acquisition hardware plays an important role in the source localization system.

CHAPTER 7

OUTDOOR EXPERIMENTS AND RESULTS

In this chapter we discuss in detail the experiments we conducted in an outdoor environment and their results and analysis. Section 7.1 gives the details of the experimental scenarios we are considering and the environment details. Section 7.2 discuss in detail the experiments conducted using the RevoLabs Wireless Microphone System and their results while section 7.3 discuss the experiments and their results conducted using VocoPro Wireless Microphone System. Section 7.5 concludes this chapter.

7.1 Experimental Scenarios

Fig. 7.1 shows the satellite image of the outdoor location where we conducted all our outdoor experiments. This image was taken using GoogleTM Earth [43] software, version 6.1.0 in April 2012. The red solid line in Fig. 7.1 provides the

distances of the nearby buildings and trees from the center of the spot where experiments took place. These distances are long enough such that there is no reverberation observed in the signals we captured. However, the reflections from the walls of the nearby buildings still exist, specifically in the case of loaded gunfire.



Figure 7.1: Satellite image of the outdoor experimental environment

Fig. 7.2 shows all the experimental scenarios in the form of block diagram. Note that there is no center- or wall-version of the experiment as we did in the indoor experiments. Also Fig. 7.2 shows only two sampling rates i.e. 4kHz and 8kHz although we acquired the signals at 16kHz as well. This is because the OC algorithm could not be run on the PCs we have because of the large amount of memory required for the case of 16kHz, hence we did not include it in Fig. 7.2.

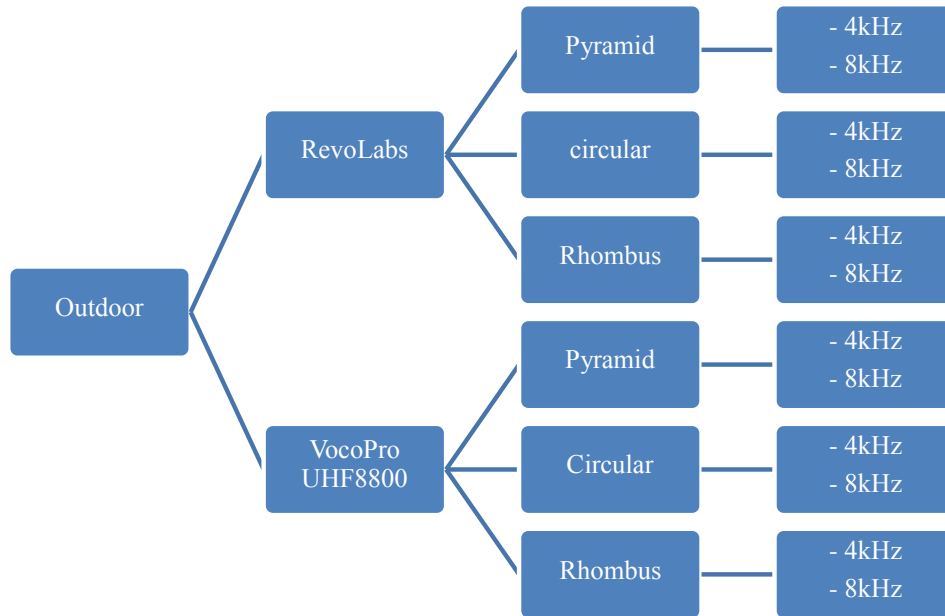


Figure 7.2: Outdoor experimental scenarios

7.2 Results using RevoLabs System

In this section we discuss in detail the outdoor experiments we conducted using the RevoLabs system and their results and analysis. As there are three different geometries and each geometry has different scenarios we need to organize the discussion in the following subsections. Subsections 7.2.1, 7.2.2 and 7.2.3 discuss the results for Pyramid, Circular and Rhombus geometries in outdoor environment using RevoLabs System respectively. The observations and discussion are given in subsection 7.2.4. It is to be noted that we used an unloaded toy gunshot for all the experiments using the RevoLabs system.

7.2.1 Pyramid Geometry Experiments and Results

In this section we discuss the results for the Pyramid geometry obtained using the RevoLabs wireless microphone system. We are using the geometry with the source locations discussed in Section 5.2.1. Fig. 7.3 shows the experimental setup of the Pyramid geometry using the RevoLabs system. Note that we carried out all our experiments during night to avoid the daily life noise and disturbances although a practical system would require to work in any situation. This is because we wanted to assess the OC algorithm performance and compare it with CC results without the results being affected by the environmental factors.



Figure 7.3: Pyramid geometry using RevoLabs system outdoor

There are several experimental scenarios as was shown in Fig. 7.2. The results

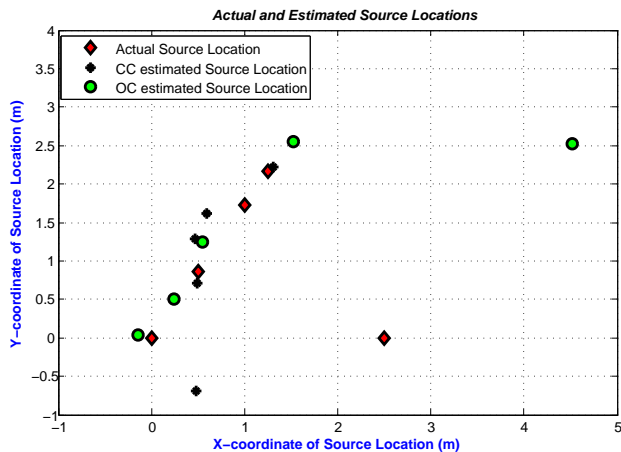
are organized with respect to the sampling rate and follow next.

Results for the 8kHz Case

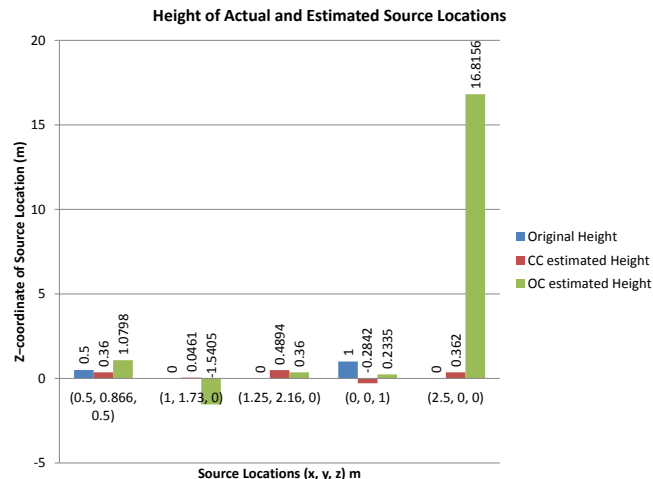
Table 7.1 shows the results we obtained for the case of 8kHz. Using the results from Table 7.1 we plot the x, y coordinates of the estimated source locations in Fig. 7.5a while Fig. 7.4b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.1: Pyramid Outdoor results using RevoLabs at 8kHz. Fast Machine was used.

Source Location	$(\cos 60^\circ, \sin 60^\circ, 0.5)$ $= (0.5, 0.866, 0.5)$	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	$(2.5\cos 60^\circ, 2.5\sin 60^\circ, 0) = (1.25, 2.16, 0)$	$(0, 0, 1)$	$(2.5, 0, 0)$
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	In-between a_1 & a_2 , outside the array	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array
Exact TDs $(\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40})$	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0037, 0.0015, 0.0084, 0)	(0.0019, -0.0008, 0.0073, 0.0006)	(-0.0009, -0.0008, 0.0064, -0.0001)	(0.0047, 0.0085, 0.0053, -0.0024)	(-0.0040, 0.0049, 0.0083, 0.0004)
TDE_{OC} $(\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc})$	(0.0026, 0.0024, 0.0045, -0.0015)	(0.0011, 0.0005, 0.0046, 0.0016)	(-0.0018, -0.0015, 0.0072, 0)	(0.0026, 0.0021, 0.0022, 0.0001)	(-0.0013, 0.0004, 0.0019, -0.0021)
X_{CC} $(x_s^{cc}, y_s^{cc}, z_s^{cc})$	(0.4683, 1.2863, 0.3600)	(0.5932, 1.6194, 0.0461)	(1.3071, 2.2164, 0.4894)	(0.4857, -0.6912, -0.2842)	(0.4866, 0.7143, 0.3620)
X_{OC} $(x_s^{oc}, y_s^{oc}, z_s^{oc})$	(0.2428, 0.5038, 1.0798)	(0.5426, 1.2436, -1.5405)	(1.5263, 2.5573, 0.3600)	(-0.1457, 0.0351, 0.2335)	(4.5163, 2.5238, 16.8156)
$\epsilon_{cc,3D}$	0.4441	0.4241	0.4959	1.5371	2.1668
$\epsilon_{cc,2D}$	0.4215	0.4215	0.0802	0.8448	2.1363
$\epsilon_{oc,3D}$	0.7304	1.6790	0.6032	0.7810	17.1230
$\epsilon_{oc,2D}$	0.4443	0.6677	0.4840	0.1498	3.2303
Time for CC (Sec)	0.0345	0.0351	0.0350	0.0351	0.0343
Time for OC (Sec)	61.4610	66.5716	62.8896	45.6164	38.9513



(a)



(b)

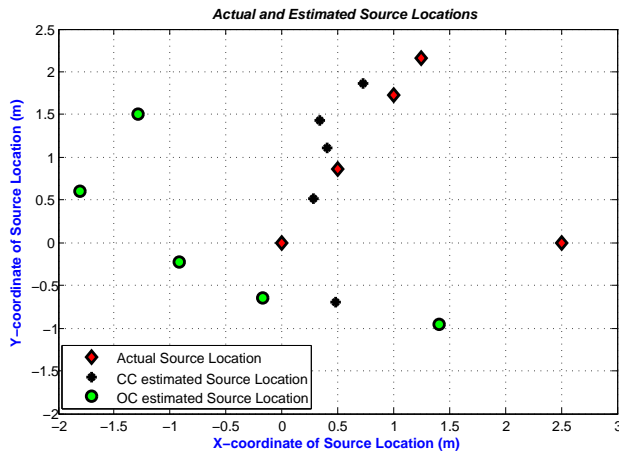
Figure 7.4: Pyramid Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

Results for the 4kHz Case

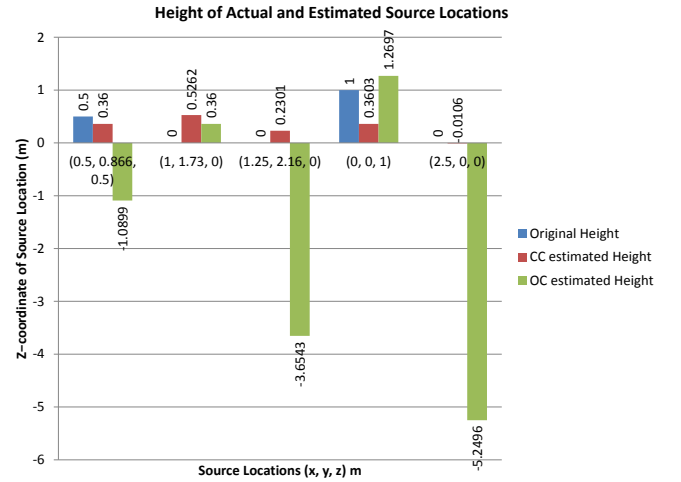
Table 7.2 shows the results we obtained for the case of 4kHz. Using the results from Table 7.2 we plot the x, y coordinates of the estimated source locations in Fig. 7.5a while Fig. 7.5b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.2: Pyramid Outdoor results using RevoLabs at 4kHz. Slow Machine was used.

Source Location	$(\cos 60^\circ, \sin 60^\circ, 0.5)$ $= (0.5, 0.866, 0.5)$	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	$(2.5\cos 60^\circ, 2.5\sin 60^\circ, 0) = (1.25, 2.16, 0)$	$(0, 0, 1)$	$(2.5, 0, 0)$
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	In-between a_1 & a_2 , outside the array	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array
Exact TDs $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0035, -0.0015, 0.0077, 0)	(0.0032, 0.0018, 0.0073, -0.0010)	(0.0013, -0.0020, 0.0075, 0.0003)	(0.0047, 0.0085, 0.0053, 0.0010)	(-0.0043, 0.0057, 0.0083, -0.0015)
TDE_{OC} $(\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc})$	(0.0030, -0.0015, 0.0022, 0.0008)	(-0.0022, -0.0027, -0.0008, 0)	(-0.0035, -0.0015, -0.0008, -0.0027)	(-0.0010, 0.0045, 0.0018, -0.0008)	(-0.0040, 0.0008, -0.0010, -0.0030)
X_{CC} $(x_s^{cc}, y_s^{cc}, z_s^{cc})$	(0.3426, 1.4275, 0.3600)	(0.4063, 1.1077, 0.5262)	(0.7301, 1.8580, 0.2301)	(0.4857, -0.6912, 0.3603)	(0.2842, 0.5182, -0.0106)
X_{OC} $(x_s^{oc}, y_s^{oc}, z_s^{oc})$	(-1.2830, 1.5068, -1.0899)	(-0.1711, -0.6427, 0.3600)	(-0.9151, -0.2213, -3.6543)	(1.4036, -0.9580, 1.2697)	(-1.8082, 0.5998, -5.2496)
$\epsilon_{cc,3D}$	0.5998	1.0083	0.6438	1.0597	2.2756
$\epsilon_{cc,2D}$	0.5832	0.8600	0.6013	0.8448	2.2756
$\epsilon_{oc,3D}$	2.4733	2.6704	4.8695	1.7207	6.8176
$\epsilon_{oc,2D}$	1.8946	2.6460	3.2184	1.6994	4.3498
Time for CC (Sec)	0.4294	0.0427	0.0433	0.0429	0.0426
Time for OC (Sec)	16.2785	19.5867	20.3712	22.0094	19.8440



(a)



(b)

Figure 7.5: Pyramid Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

7.2.2 Circular Geometry Experiments and Results

In this section we discuss the results for the Circular geometry obtained using RevoLabs wireless microphone system. We are using the geometry with the source locations discussed in Section 5.2.1. Fig. 7.6 shows the experimental setup of the Circular geometry using RevoLabs system which was also conducted during night.

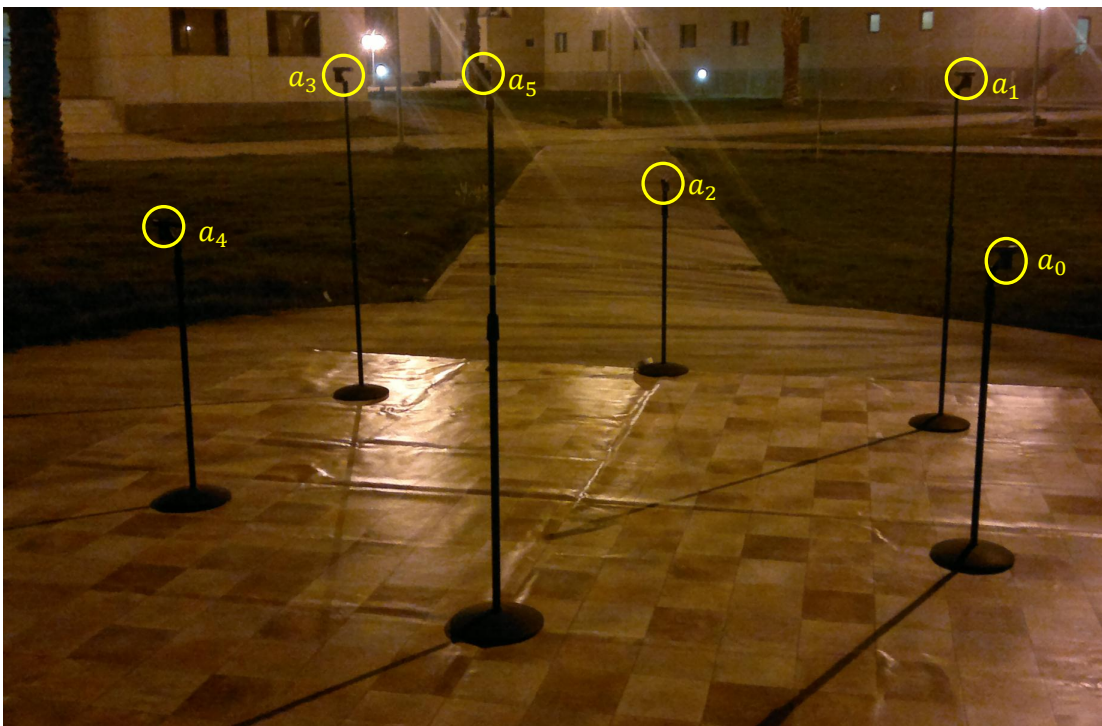


Figure 7.6: Circular geometry using RevoLabs system outdoor

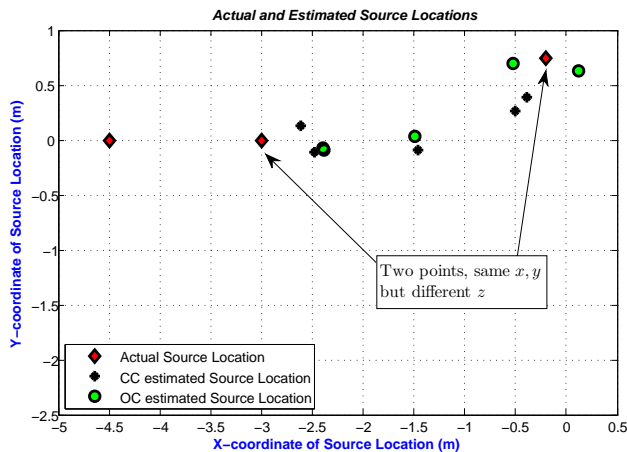
There are several experimental scenarios as was shown in Fig. 7.2. The results are organized with respect to the sampling rate and follows next.

Results for the 8kHz Case

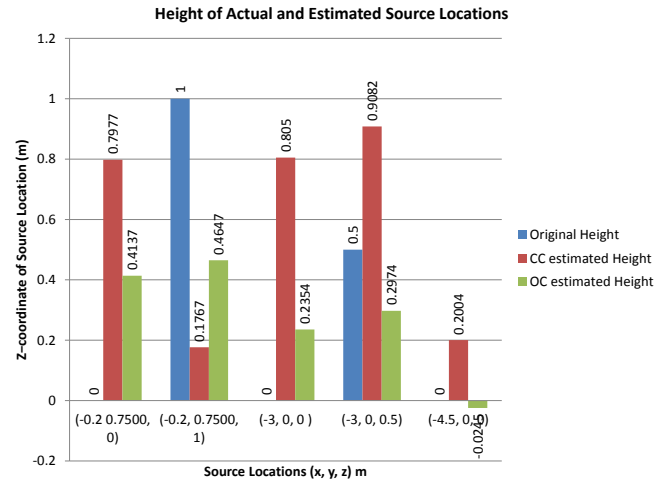
Table 7.3 shows the results we obtained for the case of 8kHz. Using the results from Table 7.3 we plot the x, y coordinates of the estimated source locations in Fig. 7.7a while Fig. 7.7b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.3: Circular Outdoor results using RevoLabs at 8kHz. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 0m	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 1m	At mic3 (a_3) with zero height	At mic3 (a_3) with 0.5m height	Along the line (outside array) passing through mic3 (a_3) and center of the array at height 0m
Exact TDs ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0014, 0.0067, 0.0079, 0.0074, 0.0048)	(-0.0033, 0.0050, 0.0070, 0.0064, 0.0040)	(-0.0014, -0.0038, -0.0057, -0.0029, -0.0012)	(-0.0010, -0.0033, -0.0069, -0.0033, -0.0018)	(0.0006, -0.0043, -0.0044, -0.0034, -0.0006)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0009, 0.0049, 0.0069, 0.0066, 0.0038)	(-0.0029, 0.0031, 0.0050, 0.0055, 0.0035)	(-0.0006, -0.0035, -0.0062, -0.0035, -0.0013)	(-0.0009, -0.0030, -0.0075, -0.0038, -0.0010)	(0.0008, -0.0010, -0.0019, 0, 0.0019)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-0.5055, 0.2660, 0.7977)	(-0.3900, 0.3906, 0.1767)	(-2.6153, 0.1293, 0.8050)	(-2.4742, -0.1079, 0.9082)	(-1.4648, -0.0894, 0.2004)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(0.1202, 0.6295, 0.4137)	(-0.5272, 0.7006, 0.4647)	(-2.3847, -0.0900, 0.2354)	(-2.3921, -0.0745, 0.2974)	(-1.4908, 0.0295, -0.0245)
$\epsilon_{cc,3D}$	0.9818	0.9181	0.9015	0.6743	3.0431
$\epsilon_{cc,2D}$	0.5723	0.4065	0.4059	0.5367	3.0365
$\epsilon_{oc,3D}$	0.5368	0.6293	0.6649	0.6451	3.0095
$\epsilon_{oc,2D}$	0.3421	0.3309	0.6218	0.6125	3.0094
Time for CC (Sec)	0.2126	0.2356	0.1413	0.1381	0.1543
Time for OC (Sec)	56.1228	55.1528	65.5428	77.7755	62.7603



(a)



(b)

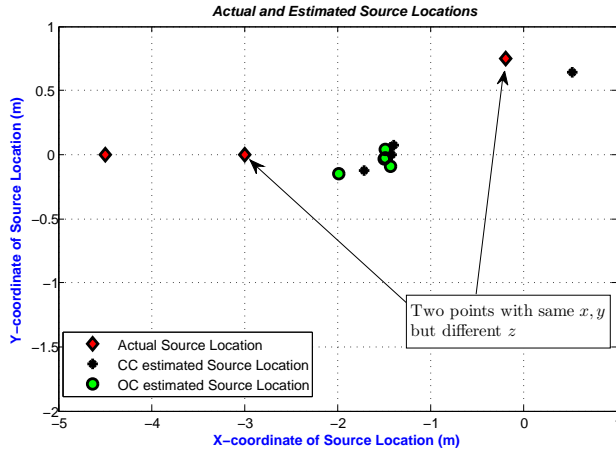
Figure 7.7: Circular Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

Results for the 4kHz Case

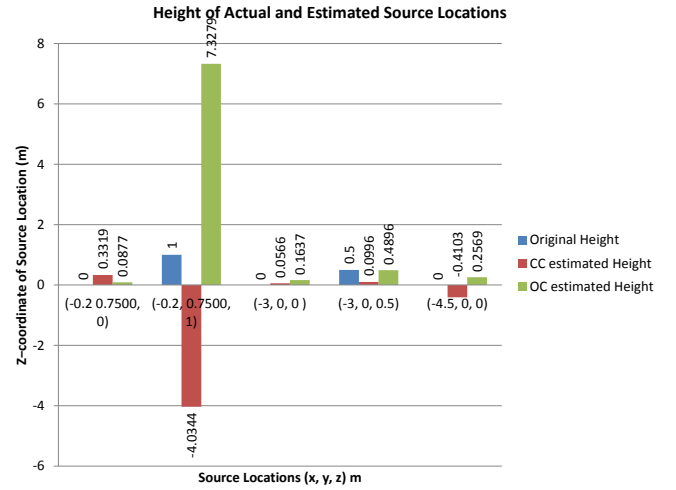
Table 7.4 shows the results we obtained for the case of 4kHz. Using the results from Table 7.4 we plot the x, y coordinates of the estimated source locations in Fig. 7.8a while Fig. 7.8b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.4: Circular Outdoor results using RevoLabs at 4kHz. Slow Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 0m	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 1m	At mic3 (a_3) with zero height	At mic3 (a_3) with 0.5m height	Along the line (outside array) passing through mic3 (a_3) and center of the array at height 0m
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{CC}, \tau_{20}^{CC}, \tau_{30}^{CC}, \tau_{40}^{CC}$)	(0.0035, 0.0057, 0.0107, 0.0065, 0.0057)	(0.0037, 0.0057, 0.0098, 0.0065, 0.0065)	(-0.0003, 0.0047, 0.0070, 0.0020, 0.0020)	(-0.0013, 0.0008, 0.0020, -0.0022, -0.0020)	(0.0008, -0.0050, -0.0037, -0.0008, 0.0030)
TDE_{OC} ($\tau_{10}^{OC}, \tau_{20}^{OC}, \tau_{30}^{OC}, \tau_{40}^{OC}$)	(-0.0020, 0.0010, -0.0003, -0.0008, 0.0010)	(0.0032, 0.0008, 0.0047, 0.0010, 0.0037)	(0.0005, -0.0008, -0.0003, 0.0010, -0.0020)	(0.0027, -0.0003, 0.0005, 0, 0.0018)	(0.0027, -0.0020, -0.0005, 0.0005, 0.0030)
X_{CC} ($x_s^{CC}, y_s^{CC}, z_s^{CC}$)	(-1.3972, 0.0751, 0.3319)	(0.5197, 0.6436, -4.0344)	(-1.7170, -0.1214, 0.0566)	(-1.4342, 0.0593, 0.0996)	(-1.4167, -0.0022, -0.4103)
X_{OC} ($x_s^{OC}, y_s^{OC}, z_s^{OC}$)	(-1.4983, -0.0303, 0.0877)	(-1.9887, -0.1471, 7.3279)	(-1.4932, 0.0392, 0.1637)	(-1.4878, -0.0238, 0.4896)	(-1.4308, -0.0933, 0.2569)
$\epsilon_{CC,3D}$	1.4139	5.0867	1.2900	1.6173	3.1105
$\epsilon_{CC,2D}$	1.3744	0.7276	1.2888	1.5669	3.0833
$\epsilon_{OC,3D}$	1.5173	6.6367	1.5161	1.5124	3.0814
$\epsilon_{OC,2D}$	1.5147	2.0011	1.5073	1.5124	3.0706
Time for CC (Sec)	0.0462	0.0462	0.0474	0.0455	0.0452
Time for OC (Sec)	26.6654	21.1501	23.5461	23.4298	29.4686



(a)



(b)

Figure 7.8: Circular Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

7.2.3 Rhombus Geometry Experiments and Results

In this section we discuss the results for the Rhombus geometry obtained using the RevoLabs wireless microphone system. We are using the geometry with the source locations discussed in Section 5.2.1.

There are several experimental scenarios as was shown in Fig. 7.2. The results are organized with respect to the sampling rate and follow next.

Results for the 8kHz Case

Table 7.5 shows the results we obtained for the case of 8kHz. Using the results from Table 7.5 we plot the x, y coordinates of the estimated source locations in Fig. 7.9a while Fig. 7.9b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

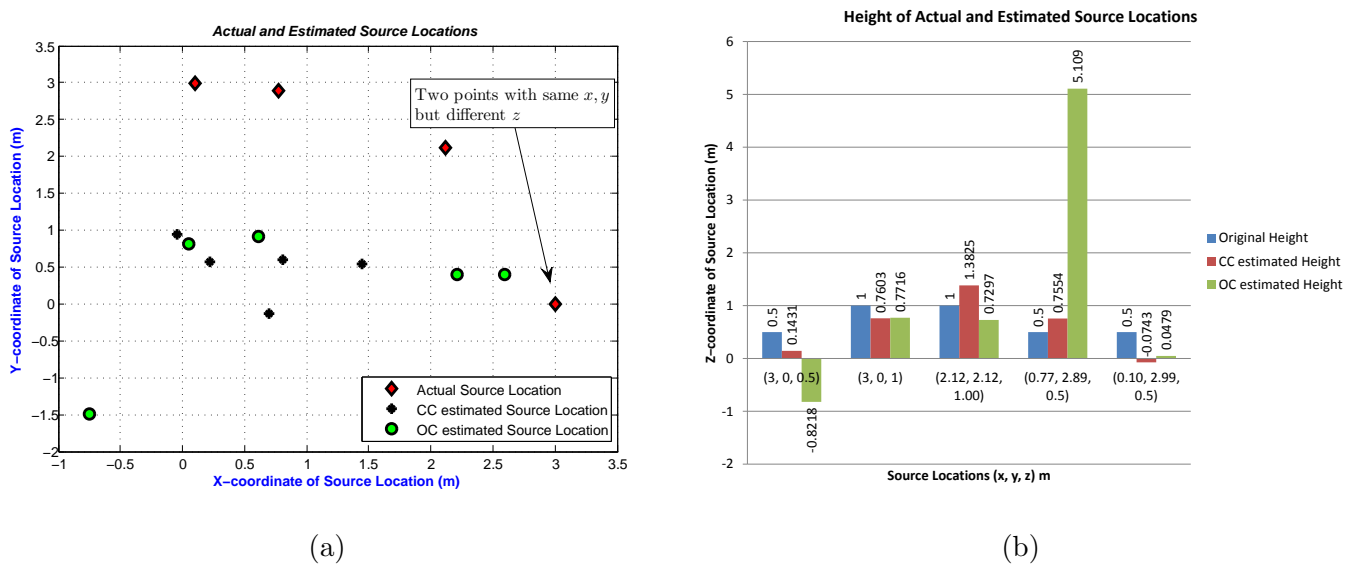


Figure 7.9: Rhombus Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

Table 7.5: Rhombus Outdoor results using RevoLabs at 8kHz. Fast Machine was used.

Source Location	(2r, 0, h)	(2r, 0, 2h)	(2r cos 45°, 2r sin 45°, 2h)	(2r cos 75°, 2r sin 75°, h)	(2r cos 88°, 2r sin 88°, h)
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, 2.99, 0.5)
Description	On the line passing through mic1 (a_1) and center of the array at height of 0.5m	On the line passing through mic1 (a_1) and center of the array at height of 1m	In-between mic1 (a_1) and 2 (a_2) outside the circumference on which mics lie.	At angle of 75° between mic1 (a_1) and 2 (a_2)	At angle of 88° between mic1 (a_1) and 2 (a_2)
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(-0.0039, 0.0006, 0.0049, 0.0005, 0.0007)	(-0.0018, 0.0001, 0.0035, 0.0009, -0.0009)	(0.0032, -0.0014, -0.0024, 0.0025, 0.0015)	(0.0008, -0.0011, 0.0045, 0.0050, 0.0015)	(0.0039, -0.0001, 0.0036, 0.0040, 0.0024)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(-0.0041, 0.0013, 0.0054, 0.0011, 0.0014)	(-0.0038, 0.0003, 0.0038, 0.0001, -0.0003)	(-0.0020, -0.0016, 0.0030, 0.0019, -0.0047)	(0.0003, -0.0004, 0.0014, 0.0018, 0.0020)	(0.0015, 0.0004, 0.0018, 0.0020, 0.0009)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(1.4432, 0.5489, 0.1431)	(0.8066, 0.6000, 0.7603)	(0.6966, -0.1342, 1.3825)	(0.2166, 0.5760, 0.7554)	(-0.0419, 0.9498, -0.0743)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(2.5922, 0.4034, -0.8218)	(2.2121, 0.4030, 0.7716)	(0.6110, 0.9208, 0.7297)	(-0.7522, -1.4986, 5.1090)	(0.0536, 0.8116, 0.0479)
$\epsilon_{cc,3D}$	1.6888	2.2866	2.6933	2.3930	2.1242
$\epsilon_{cc,2D}$	1.6507	2.2740	2.6660	2.3793	2.0451
$\epsilon_{oc,3D}$	1.4409	0.9139	1.9464	6.5436	2.2253
$\epsilon_{oc,2D}$	0.5736	0.8849	1.9275	4.6451	2.1789
Time for CC (Sec)	0.0372	0.0376	0.0372	0.0374	0.0373
Time for OC (Sec)	56.8121	49.8612	54.3802	48.8363	48.1199

Results for the 4kHz Case

Table 7.6 shows the results we obtained for the case of 4kHz. Using the results from Table 7.6 we plot the x, y coordinates of the estimated source locations in Fig. 7.10a while Fig. 7.10b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.6: Rhombus Outdoor results using RevoLabs at 4kHz. Slow Machine was used.

Source Location	(2r, 0, h)	(2r, 0, 2h)	$(2r \cos 45^\circ, 2r \sin 45^\circ, 2h)$	$(2r \cos 75^\circ, 2r \sin 75^\circ, h)$	$(2r \cos 88^\circ, 2r \sin 88^\circ, h)$
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, 2.99, 0.5)
Description	On the line passing through mic1 (a_1) and center of the array at height of 0.5m	On the line passing through mic1 (a_1) and center of the array at height of 1m	In-between mic1 (a_1) and 2 (a_2) outside the circumference on which mics lie.	At angle of 75° between mic1 (a_1) and 2 (a_2)	At angle of 88° between mic1 (a_1) and 2 (a_2)
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(-0.0032, 0.0045, 0.0065, 0.0030, 0.0022)	(-0.0013, 0.0050, 0.0067, 0.0035, 0.0018)	(-0.0032, -0.1645, -0.0015, 0.0008, 0.0015)	(0.0045, -0.0010, 0.0045, 0.0037, 0.0015)	(0.0027, -0.0003, 0.0037, 0.0040, 0.0013)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(-0.0005, 0.0010, 0.0022, 0.0013, 0.0020)	(-0.0045, -0.0003, -0.0010, -0.0018, -0.0010)	(0.0010, 0.0020, 0.0013, 0.0008, 0.0013)	(0.0015, -0.0003, 0.0027, 0.0018, 0.0005)	(-0.0025, -0.0003, -0.0013, -0.0025, -0.0037)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(0.2340, 0.2875, 0.2877)	(0.3801, 0.2485, 0.6474)	(5.4664, 14.6437, 5.3962)	(0, 0.7082, 0.3440)	(0.1391, 0.8598, 0.3126)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-1.7069, 0.4012, 4.8467)	(-0.5855, 0.7196, 0.2520)	(0.1301, -0.6510, -1.4182)	(0.2525, 0.8673, 0.2308)	(-0.0305, 0.5173, 1.0128)
$\epsilon_{cc,3D}$	2.7890	2.6552	13.6882	2.3189	2.1388
$\epsilon_{cc,2D}$	2.7809	2.6317	12.9631	2.3137	2.1305
$\epsilon_{oc,3D}$	6.4195	3.7327	4.1816	2.1052	2.5287
$\epsilon_{oc,2D}$	4.7239	3.6570	3.4115	2.0879	2.4762
Time for CC (Sec)	0.0506	0.0522	0.0456	0.0452	0.0450
Time for OC (Sec)	25.4932	24.1869	23.6965	23.7874	24.1993

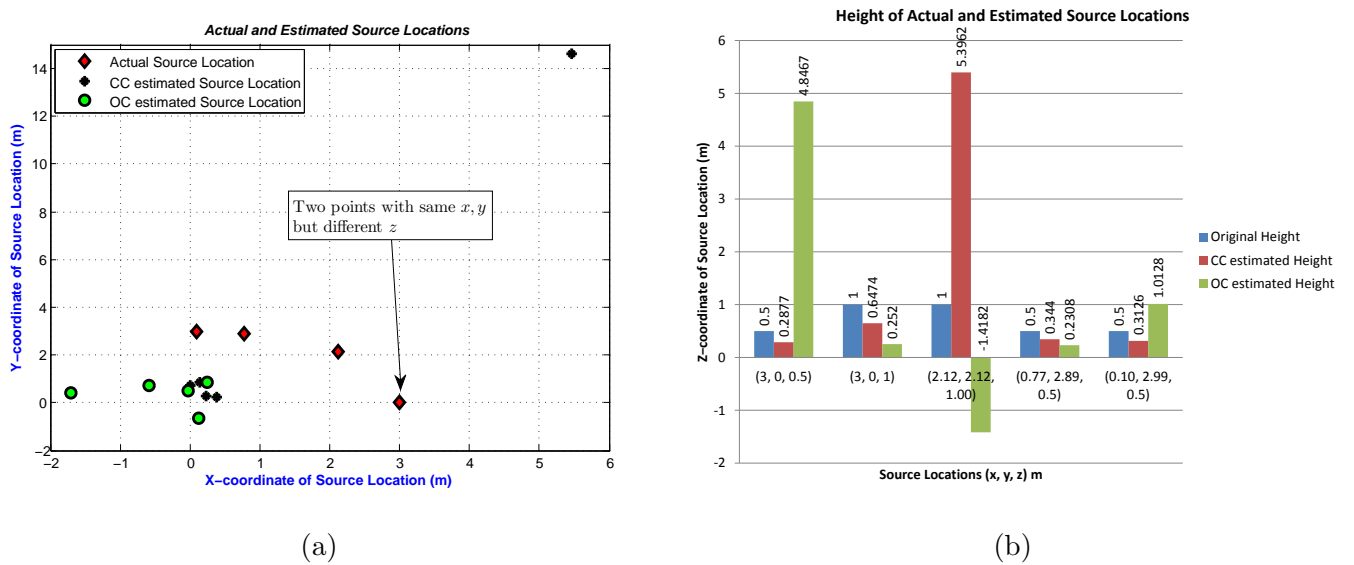


Figure 7.10: Rhombus Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using RevoLabs system, (a) The x, y coordinates of estimated source Locations, (b) The z -coordinate of the estimated source locations.

7.2.4 Observation for RevoLabs Outdoor Experiments

Looking at the results in subsections 7.2.1 to 7.2.3 we can find the numerical facts about the results. We present these numerical facts in subsection 7.2.4 and then based on these numerical facts we deduce the observations and present them in subsection 7.2.4

Numerical Facts

1. In outdoor experiments using RevoLabs system the CC produces more accurate results than OC for the Pyramid Geometry. 80% of the time the CC produce more accurate results for the Pyramid geometry by a factor of at least 1.21 than OC i.e. CC 3D-MSE ($\epsilon_{CC.3D}$) is at least 1.21 times less than OC 3D-MSE ($\epsilon_{OC.3D}$) at 8kHz. For 4kHz case 100% of the times CC produce more accurate results than OC by a minimum factor of 1.63. The runtime for OC is at least 37 times more than the runtime for CC.
2. For Circular geometry 8kHz case , the OC produces more accurate results than CC 100% of the times by a minimum factor of 1.003 while for the 4kHz case CC produces better results than OC by a minimum factor of 1.007 60% of the times. The minimum runtime for OC is at least 230 times more than CC runtime.
3. For Rhombus geometry 8kHz case , the OC produces more accurate results than CC 60% of the times by a minimum factor of 1.16 while for the 4kHz case CC produces better results than OC by a minimum factor of 1.18 60%

of the times. The minimum runtime for OC is at least 230 times more than CC runtime.

Outdoor RevoLabs Experiments Observations

Looking at the numerical facts in subsection 7.2.4 we can make the following observations for the experiments carried out outdoor using RevoLabs system.

1. The CC is producing the best results for Pyramid geometry in outdoor using RevoLabs system with a minimum 3D-MSE ϵ_{CC_3D} of 0.4215m and minimum 2D error of 0.08m.
2. The OC produces better results for the Circular in 8kHz case 100% of the times and for the Pyramid 60% of the times respectively. However for the case of 4kHz CC produces better results than OC 60% of the times. It means the OC efficiency decreases as the sampling rate is decreased.
3. Overall the CC produces better results 66% of the times in all the outdoor cases using RevoLabs and the OC is better only 34% of the times. The accuracy of CC in outdoors is due to the fact that because of the absence of reverberations and nearby reflections there is no signal shape distortion among the signals received at all microphones. Thus their correlation with each other is better than the indoor environment.

7.3 Results using VocoPro System

In this section we discuss in detail the outdoor experiments we conducted using the VocoPro system and their results and analysis. As there are three different geometries and each geometry has different scenarios we need to organize the discussion in subsections. Subsections 7.3.1, 7.3.2 and 7.3.3 discuss the results for Pyramid, Circular and Rhombus geometries in outdoor environment using the VocoPro System. The observations are given in subsection 7.3.4. It is to be noted that we used a loaded toy gunshot for all the experiments using the VocoPro system.

7.3.1 Pyramid Geometry Experiments and Results

In this section we discuss the results for the Pyramid geometry obtained using the VocoPro wireless microphone system. We are using the geometry with the source locations discussed in Section 5.2.1. Fig. 7.11 shows the experimental setup of the Pyramid geometry using the VocoPro system. Like for the case of the RevoLabs system we conducted all outdoor experiments during night.

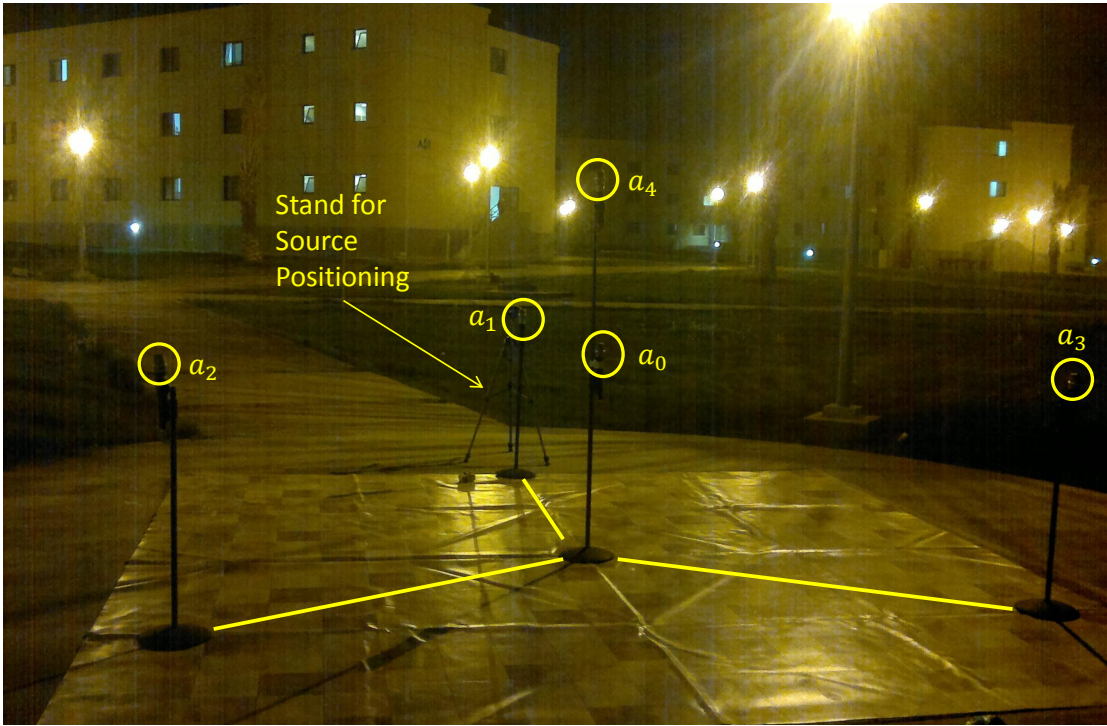


Figure 7.11: Pyramid geometry using VocoPro system outdoor

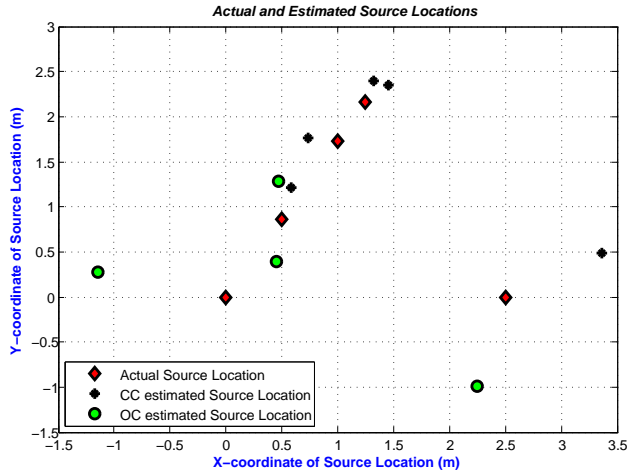
There are several experimental scenarios as was shown in Fig. 7.2. The results are organized with respect to the sampling rate and follow next.

Results for the 8kHz Case

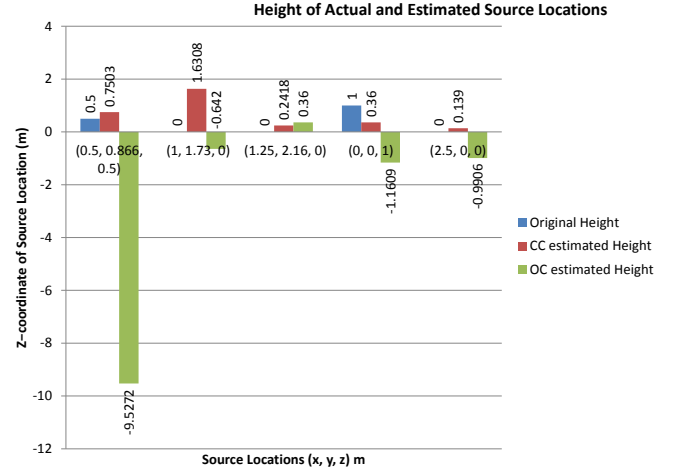
Table 7.7 shows the results we obtained for the case of 8kHz. Using the results from Table 7.7 we plot the x, y coordinates of the estimated source locations in Fig. 7.12a while Fig. 7.12b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.7: Pyramid Outdoor results using VocoPro at 8kHz. Fast Machine was used.

Source Location	$(\cos 60^\circ, \sin 60^\circ, 0.5)$ $= (0.5, 0.866, 0.5)$	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	$(2.5\cos 60^\circ, 2.5\sin 60^\circ, 0) = (1.25, 2.16, 0)$	$(0, 0, 1)$	$(2.5, 0, 0)$
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	In-between a_1 & a_2 , outside the array	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array
Exact TDs ($\tau_{10}^c, \tau_{20}^c, \tau_{30}^c, \tau_{40}^c$)	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0013, 0.0008, 0.0053, -0.0005)	(-0.0008, -0.0005, 0.0049, -0.0008)	(0.0008, -0.0005, 0.0057, 0.0001)	(0.0084, 0.0018, -0.0075, 0)	(-0.0171, 0.0031, 0.0041, 0.0001)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0049, 0.0006, 0.0019, 0.0011)	(0.0049, 0.0006, 0.0019, 0.0011)	(0.0037, 0.0015, 0.0084, 0)	(0.0005, 0.0008, 0.0014, 0.0005)	(-0.0029, 0.0046, 0.0022, 0.0009)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(0.5855, 1.2181, 0.7503)	(1.4575, 2.3525, 1.6308)	(0.7311, 1.7629, 0.2418)	(1.3204, 2.3928, 0.3600)	(3.3621, 0.4863, 0.1390)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-1.1449, 0.2765, -9.5272)	(-1.1449, 0.2765, -0.6420)	(0.4683, 1.2863, 0.3600)	(0.4525, 0.3964, -1.1609)	(2.2483, -0.9849, -0.9906)
$\epsilon_{cc,3D}$	0.4404	1.8045	0.6967	2.8069	0.9995
$\epsilon_{cc,2D}$	0.3624	0.7725	0.6534	2.7329	0.9898
$\epsilon_{oc,3D}$	2.0875	2.6694	1.2264	2.2431	1.4194
$\epsilon_{oc,2D}$	1.7474	2.5910	1.1724	0.6015	1.0165
Time for CC (Sec)	0.0350	0.0349	0.0338	0.0348	0.0352
Time for OC (Sec)	36.9219	40.2083	45.6537	38.7338	39.9076



(a)



(b)

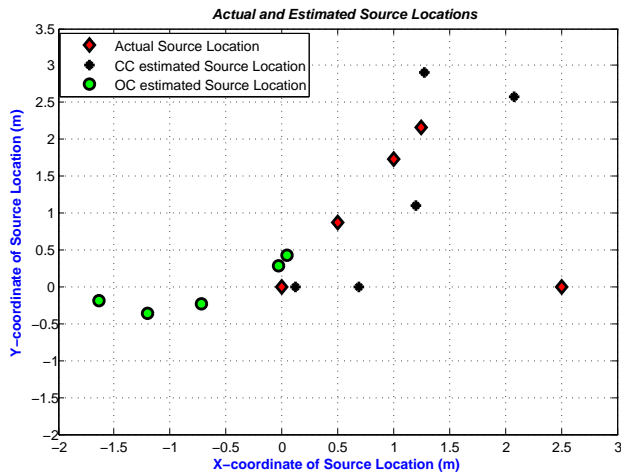
Figure 7.12: Pyramid Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

Results for the 4kHz Case

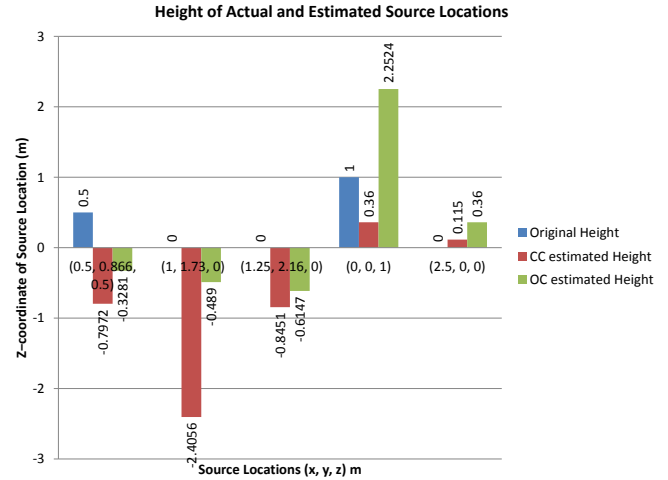
Table 7.8 shows the results we obtained for the case of 4kHz. Using the results from Table 7.8 we plot the x, y coordinates of the estimated source locations in Fig. 7.13a while Fig. 7.13b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.8: Pyramid Outdoor results using VocoPro at 4kHz. Slow Machine was used.

Source Location	$(\cos 60^\circ, \sin 60^\circ, 0.5)$ $= (0.5, 0.866, 0.5)$	$(2\cos 60^\circ, 2\sin 60^\circ, 0) = (1, 1.73, 0)$	$(2.5\cos 60^\circ, 2.5\sin 60^\circ, 0) = (1.25, 2.16, 0)$	$(0, 0, 1)$	$(2.5, 0, 0)$
Description	In-between a_1 and a_2 inside the array at height of 0.5m at 60°	In-between a_1 & a_2 on the circle inscribing the array.	In-between a_1 & a_2 , outside the array	Above the center of the array	Along the line (outside array) passing through a_1 and center of the array
Exact TDs $(\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40})$	(0.0020, 0.0020, 0.0057, -0.0003)	(0, 0.0000, 0.0059, 0.0004)	(-0.0006, -0.0006, 0.0059, 0.0003)	(0.0036, 0.0036, 0.0036, -0.0021)	(-0.0059, 0.0041, 0.0041, 0.0003)
TDE_{CC} $(\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc})$	(0.0010, 0.0035, 0.0035, 0.0013)	(-0.0015, -0.0003, 0.0047, 0.0013)	(-0.0005, -0.0018, 0.0057, 0.0008)	(-0.0035, 0.0018, 0.0018, 0)	(-0.0018, 0.0027, 0.0070, 0.0005)
TDE_{OC} $(\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc})$	(0.0035, 0.0027, 0.0040, 0.0011)	(0.0093, 0.0032, 0.0005, 0.0030)	(0.0030, 0.0022, 0.0040, 0.0013)	(0.0050, 0.0020, 0.0003, -0.0035)	(0.0057, 0.0035, 0.0022, 0)
X_{CC} $(x_s^{cc}, y_s^{cc}, z_s^{cc})$	(0.6882, 0, -0.7972)	(2.0755, 2.5704, -2.4056)	(1.2712, 2.9115, -0.8451)	(0.1250, 0, 0.3600)	(1.1979, 1.0981, 0.1150)
X_{OC} $(x_s^{oc}, y_s^{oc}, z_s^{oc})$	(-0.0310, 0.2905, -0.3281)	(-1.6407, -0.1903, -0.4890)	(0.0428, 0.4277, -0.6147)	(-1.2015, -0.3556, 2.2524)	(-0.7225, -0.2268, 0.3600)
$\epsilon_{cc,3D}$	1.5710	2.7659	1.1311	0.6521	1.7072
$\epsilon_{cc,2D}$	0.8862	1.3649	0.7518	0.1250	1.7033
$\epsilon_{oc,3D}$	1.1397	3.3015	2.1991	1.7716	3.2505
$\epsilon_{oc,2D}$	0.7830	3.2651	2.1114	1.2530	3.2305
Time for CC (Sec)	0.0496	0.0428	0.0433	0.0424	0.0435
Time for OC (Sec)	22.3595	21.4214	23.2750	21.1390	20.1693



(a)



(b)

Figure 7.13: Pyramid Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

7.3.2 Circular Geometry Experiments and Results

In this section we discuss the results for the Circular geometry obtained using the VocoPro wireless microphone system. We are using the geometry with the source locations discussed in Section 5.2.1. Fig. 7.14 shows the experimental setup of the Circular geometry using the VocoPro system.

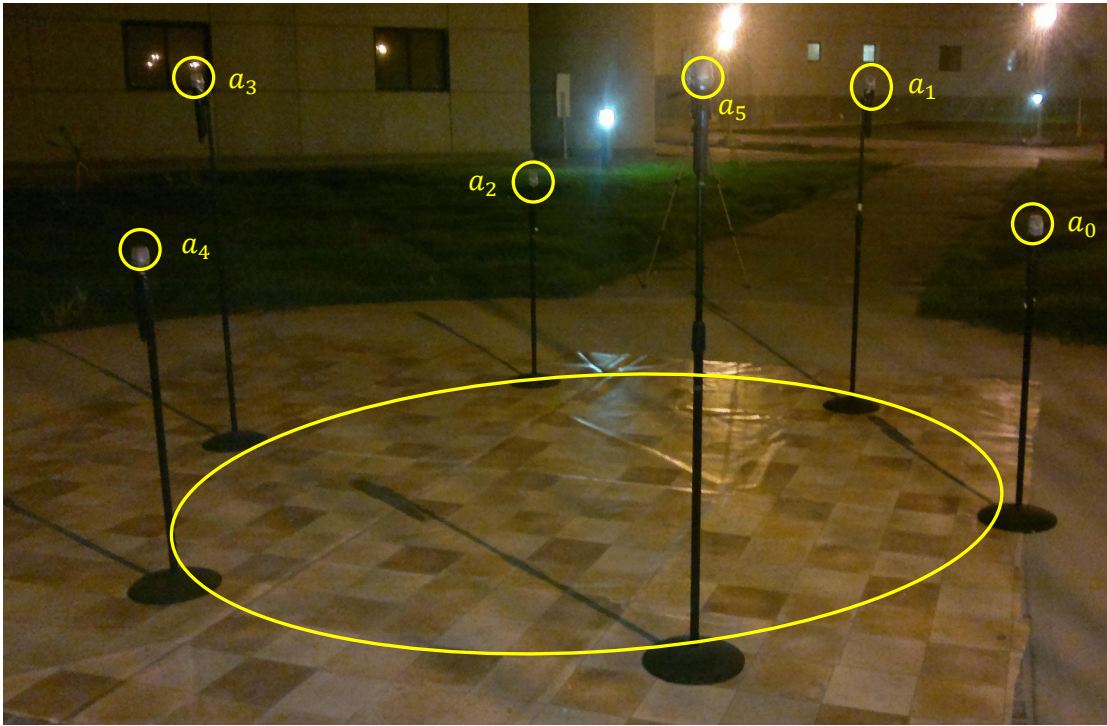


Figure 7.14: Circular geometry using VocoPro system outdoor

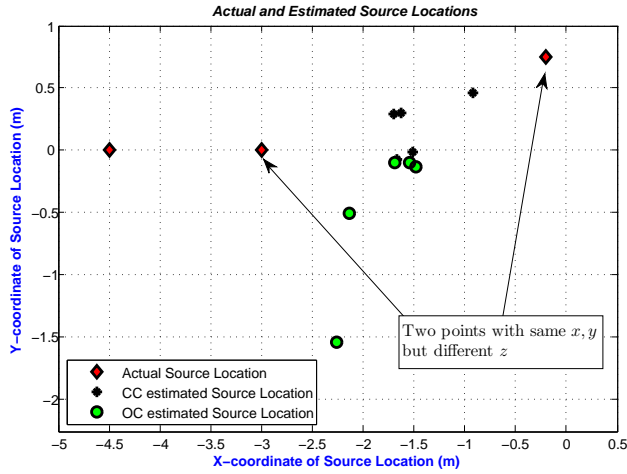
There are several experimental scenarios as was shown in Fig. 7.2. The results are organized with respect to the sampling rate and follow next.

Results for the 8kHz Case

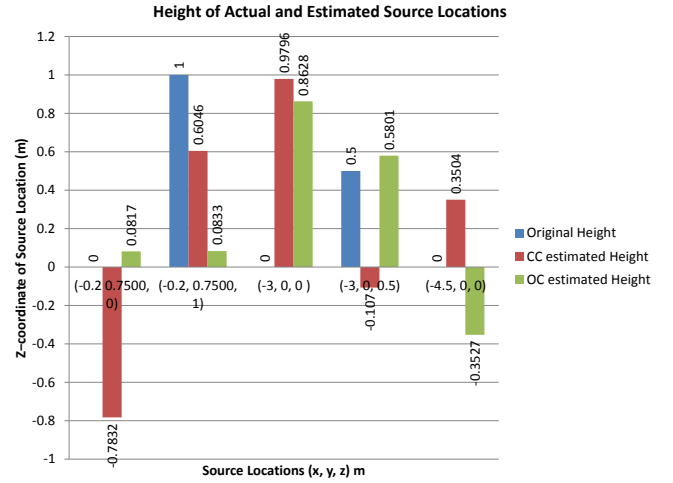
Table 7.9 shows the results we obtained for the case of 8kHz. Using the results from Table 7.9 we plot the x, y coordinates of the estimated source locations in Fig. 7.15a while Fig. 7.15b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.9: Circular Outdoor results using VocoPro at 8kHz. Fast Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 0m	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 1m	At mic3 (a_3) with zero height	At mic3 (a_3) with 0.5m height	Along the line (outside array) passing through mic3 (a_3) and center of the array at height 0m
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0019, 0.0044, 0.0073, 0.0060, 0.0060)	(0.0019, 0.0020, 0.0064, 0.0081, 0.0081)	(0.0036, 0.0025, 0.0046, 0.0003, 0.0003)	(0.0045, 0.0016, 0.0047, 0.0035, 0.0035)	(-0.0009, -0.0044, -0.0075, -0.0044, -0.0044)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(-0.0009, 0.0004, 0.0035, 0.0018, 0.0018)	(-0.0014, 0.0037, -0.0003, 0.0005, 0.0005)	(0.0080, 0.0004, 0.0047, 0.0011, 0.0011)	(-0.0014, 0.0275, 0.0025, 0.0018, 0.0018)	(0.0011, -0.0015, -0.0020, -0.0153, -0.0153)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-0.9180, 0.4587, -0.7832)	(-1.6279, 0.2979, 0.6046)	(-1.6998, 0.2942, 0.9796)	(-1.5159, -0.0180, -0.1070)	(-1.6632, -0.0749, 0.3504)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-1.5437, -0.0977, 0.0817)	(-1.4843, -0.1294, 0.0833)	(-1.6858, -0.0987, 0.8628)	(-2.2585, -1.5381, 0.5801)	(-2.1311, -0.5037, -0.3527)
$\epsilon_{cc,3D}$	1.1017	1.5491	1.6543	1.6035	2.8593
$\epsilon_{cc,2D}$	0.7748	1.4978	1.3331	1.4842	2.8378
$\epsilon_{oc,3D}$	1.5909	1.8064	1.5752	1.7094	2.4474
$\epsilon_{oc,2D}$	1.5888	1.5565	1.3180	1.7076	2.4218
Time for CC (Sec)	0.1320	0.1410	0.1944	0.1324	0.1387
Time for OC (Sec)	47.1253	48.7252	55.1040	48.7728	47.0343



(a)



(b)

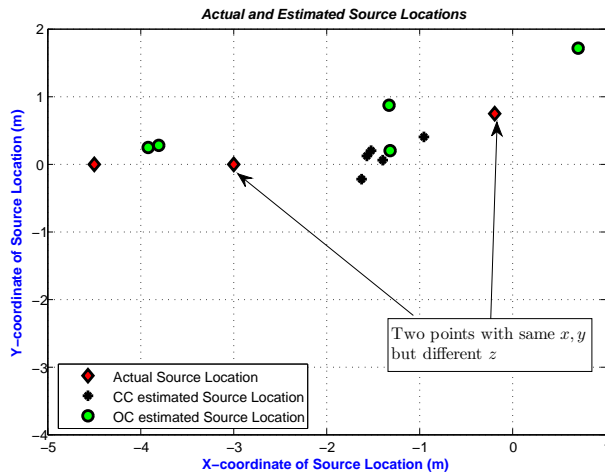
Figure 7.15: Circular Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

Results for the 4kHz Case

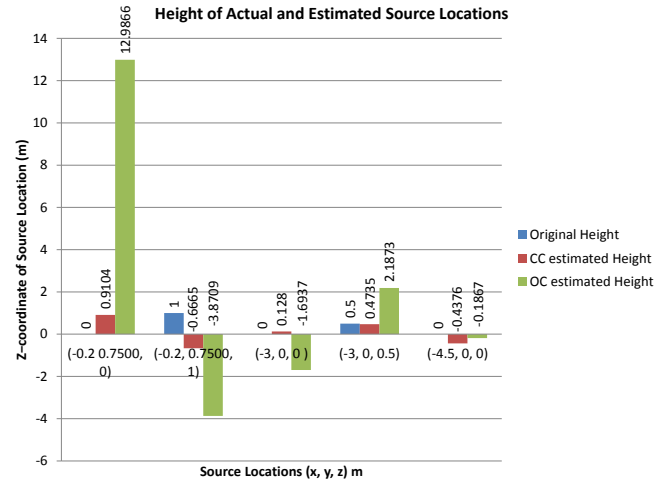
Table 7.10 shows the results we obtained for the case of 4kHz. Using the results from Table 7.10 we plot the x, y coordinates of the estimated source locations in Fig. 7.16a while Fig. 7.16b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.10: Circular Outdoor results using VocoPro at 4kHz. Slow Machine was used.

Source Location	$(r \cos 30^\circ - r, r \sin 30^\circ, 0)$	$(r \cos 30^\circ - r, r \sin 30^\circ, 1)$	$(-2r, 0, 0)$	$(-2r, 0, 0.5)$	$(-3r, 0, 0)$
Value	$(-0.2, 0.7500, 0)$	$(-0.2, 0.7500, 1)$	$(-3, 0, 0)$	$(-3, 0, 0.5)$	$(-4.5, 0, 0)$
Description	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 0m	In-between mic0 and 1 (a_0, a_1) on the circumference at height of 1m	At mic3 (a_3) with zero height	At mic3 (a_3) with 0.5m height	Along the line (outside array) passing through mic3 (a_3) and center of the array at height 0m
Exact TDs ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0004, 0.0040, 0.0064, 0.0062, 0.0041)	(-0.0010, 0.0032, 0.0049, 0.0053, 0.0027)	(-0.0010, -0.0044, -0.0074, -0.0044, -0.0010)	(-0.0013, -0.0043, -0.0089, -0.0043, -0.0013)	(-0.0015, -0.0056, -0.0086, -0.0056, -0.0015)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(0.0045, 0.0070, 0.0010, 0.0075, 0.0075)	(0.0018, 0.0040, -0.0060, 0.0053, 0.0053)	(0.0008, 0.0025, 0.0037, 0.0027, 0.0027)	(0.0040, 0.0013, 0.0043, 0, 0)	(0.0013, -0.0043, -0.0075, -0.0032, -0.0032)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0035, 0.0015, 0.0475, 0.0022, 0.0022)	(-0.0020, 0.0260, 0, -0.0022, -0.0022)	(-0.0698, 0.0013, 0.0013, 0.0018, 0.0018)	(0.0010, 0.0040, 0.0290, 0.0020, 0.0020)	(-0.0005, 0.0022, -0.0032, -0.0027, -0.0027)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(-1.5237, 0.1922, 0.9104)	(-0.9517, 0.3933, -0.6665)	(-1.4031, 0.0528, 0.1280)	(-1.5644, 0.1223, 0.4735)	(-1.6256, -0.2305, -0.4376)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-3.9176, 0.2430, 12.9866)	(-1.3257, 0.8619, -3.8709)	(0.6988, 1.7034, -1.6937)	(-3.7990, 0.2737, 2.1873)	(-1.3191, 0.1965, -0.1867)
$\epsilon_{cc,3D}$	1.7006	1.8627	1.6029	1.4410	2.9166
$\epsilon_{cc,2D}$	1.4364	0.8320	1.5977	1.4408	2.8836
$\epsilon_{oc,3D}$	13.5177	5.0005	4.4104	1.8869	3.1924
$\epsilon_{oc,2D}$	3.7520	1.1313	4.0722	0.8445	3.1870
Time for CC (Sec)	0.0533	0.0468	0.0454	0.0448	0.0464
Time for OC (Sec)	25.2732	24.8884	26.7861	24.1191	30.3864



(a)



(b)

Figure 7.16: Circular Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

7.3.3 Rhombus Geometry Experiments and Results

In this section we discuss the results for the Rhombus geometry obtained using the VocoPro wireless microphone system. We are using the geometry with the source locations discussed in Section 5.2.1. Fig. 7.17 shows the experimental setup of the Rhombus geometry using the VocoPro system.

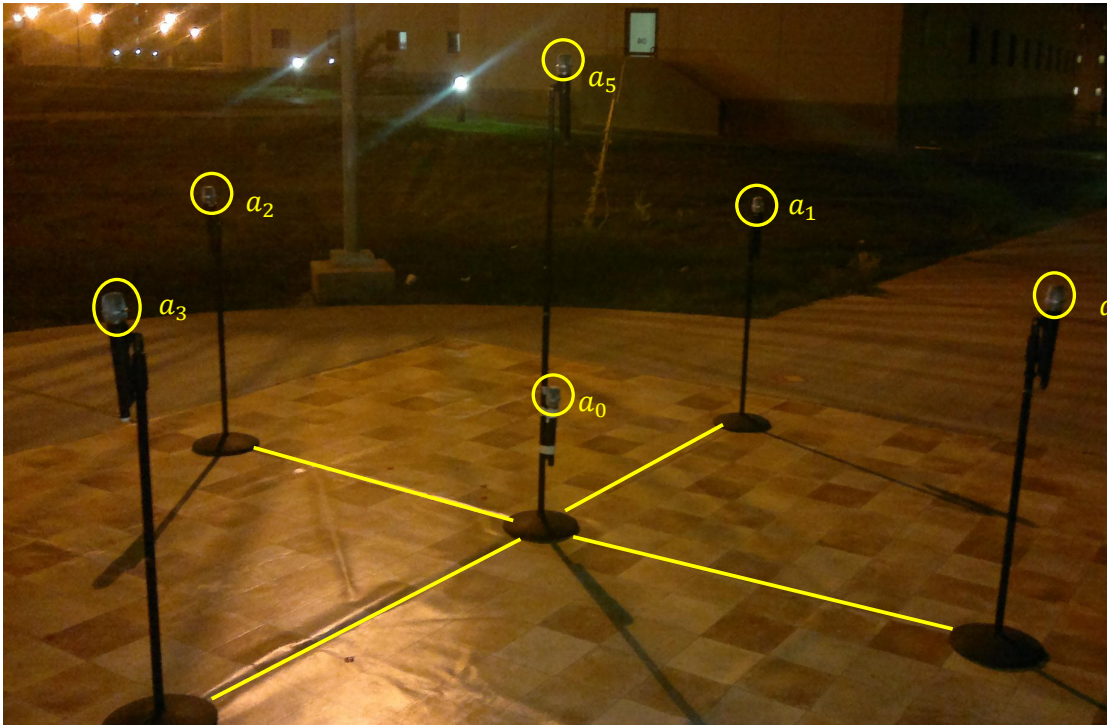


Figure 7.17: Rhombus geometry using VocoPro system outdoor

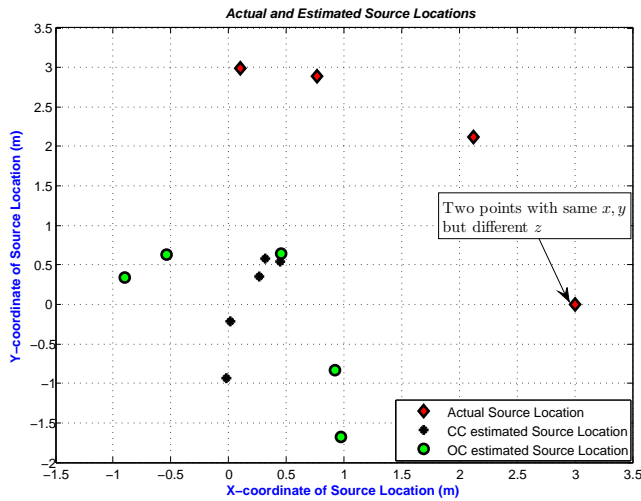
There are several experimental scenarios as was shown in Fig. 7.2. The results are organized with respect to the sampling rate and follow next.

Results for the 8kHz Case

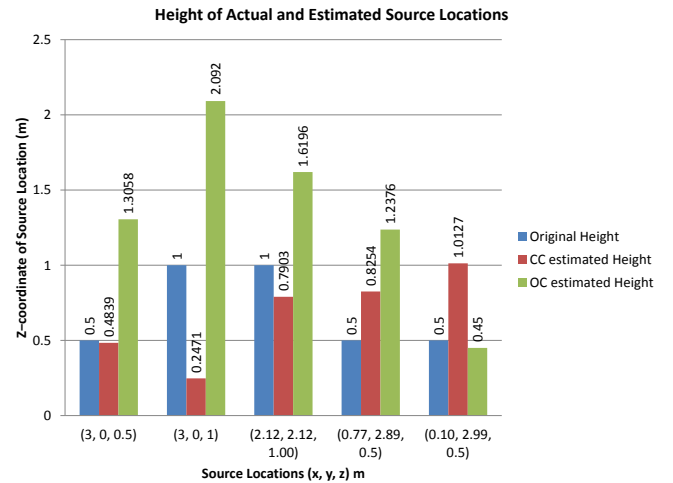
Table 7.11 shows the results we obtained for the case of 8kHz. Using the results from Table 7.11 we plot the x, y coordinates of the estimated source locations in Fig. 7.18a while Fig. 7.18b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.11: Rhombus Outdoor results using VocoPro at 8kHz. Fast Machine was used.

Source Location	(2r, 0, h)	(2r, 0, 2h)	$(2r \cos 45^\circ, 2r \sin 45^\circ, 2h)$	$(2r \cos 75^\circ, 2r \sin 75^\circ, h)$	$(2r \cos 88^\circ, 2r \sin 88^\circ, h)$
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, 2.99, 0.5)
Description	On the line passing through mic1 (a_1) and center of the array at height of 0.5m	On the line passing through mic1 (a_1) and center of the array at height of 1m	In-between mic1 (a_1) and 2 (a_2) outside the circumference on which mics lie.	At angle of 75° between mic1 (a_1) and 2 (a_2)	At angle of 88° between mic1 (a_1) and 2 (a_2)
Exact TDs ($\tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}$)	(-0.0045, 0.0009, 0.0043, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE_{CC} ($\tau_{10}^{CC}, \tau_{20}^{CC}, \tau_{30}^{CC}, \tau_{40}^{CC}$)	(-0.0027, 0.0014, 0.0045, 0.0021, 0.0015)	(-0.0035, 0.0016, -0.0063, 0.0016, 0.0019)	(-0.0011, -0.0009, -0.0029, 0.0021, 0)	(-0.0001, -0.0044, 0.0018, 0.0034, -0.0001)	(0.0008, -0.0045, 0.0011, 0.0043, 0)
TDE_{OC} ($\tau_{10}^{OC}, \tau_{20}^{OC}, \tau_{30}^{OC}, \tau_{40}^{OC}$)	(0.0027, -0.0025, -0.0011, -0.0026, -0.0035)	(-0.0018, 0.0006, 0.0021, 0.0016, 0.0020)	(0.0061, -0.0039, -0.0024, 0.0030, 0.0070)	(0.0025, -0.0030, -0.0032, 0.0045, 0.0055)	(0.0011, 0.0008, 0.0041, 0.0021, 0)
X_{CC} ($x_s^{CC}, y_s^{CC}, z_s^{CC}$)	(0.3173, 0.5778, 0.4839)	(0.4477, 0.5407, 0.2471)	(0.2694, 0.3486, 0.7903)	(0.0164, -0.2205, 0.8254)	(-0.0197, -0.9391, 1.0127)
X_{OC} ($x_s^{OC}, y_s^{OC}, z_s^{OC}$)	(-0.5406, 0.6279, 1.3058)	(-0.9036, 0.3346, 2.0920)	(0.9698, -1.6825, 1.6196)	(0.9241, -0.8295, 1.2376)	(0.4525, 0.6390, 0.4500)
$\epsilon_{cc,3D}$	2.7443	2.7154	2.5704	3.2170	3.9642
$\epsilon_{cc,2D}$	2.7443	2.6089	2.5618	3.2005	3.9309
$\epsilon_{oc,3D}$	3.6850	4.0673	4.0206	3.7950	2.3778
$\epsilon_{oc,2D}$	3.5958	3.9179	3.9726	3.7226	2.3773
Time for CC (Sec)	0.0373	0.0377	0.0375	0.0377	0.0375
Time for OC (Sec)	44.7077	49.3199	48.9376	46.3949	50.2759



(a)



(b)

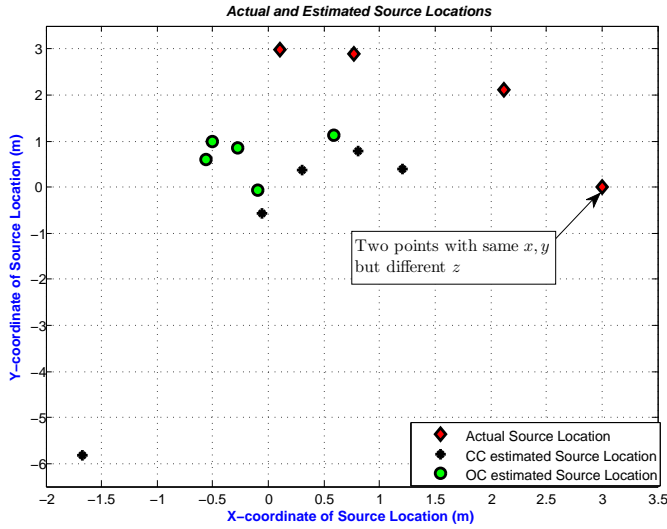
Figure 7.18: Rhombus Geometry Source Locations Estimated by CC and OC at 8kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

Results for the 4kHz Case

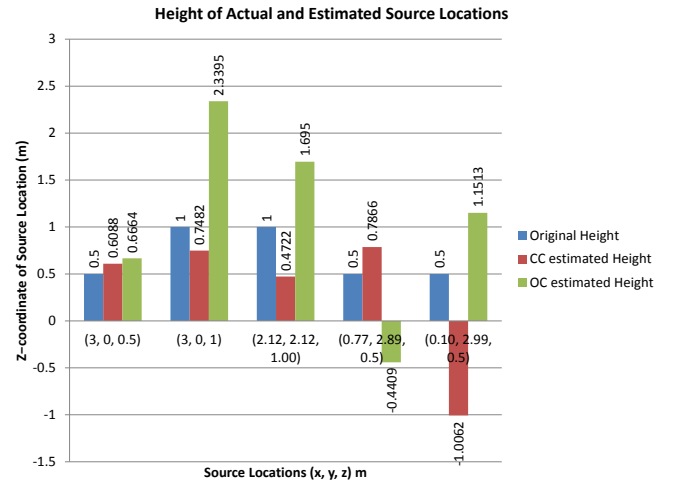
Table 7.12 shows the results we obtained for the case of 4kHz. Using the results from Table 7.12 we plot the x, y coordinates of the estimated source locations in Fig. 7.19a while Fig. 7.19b plots the bargraph for the z -coordinate of the estimated source locations along with the actual source locations.

Table 7.12: Rhombus Outdoor results using the VocoPro at 4kHz. Slow Machine was used.

Source Location	$(2r, 0, h)$	$(2r, 0, 2h)$	$(2r \cos 45^\circ, 2r \sin 45^\circ, 2h)$	$(2r \cos 75^\circ, 2r \sin 75^\circ, h)$	$(2r \cos 88^\circ, 2r \sin 88^\circ, h)$
Value	(3, 0, 0.5)	(3, 0, 1)	(2.12, 2.12, 1.00)	(0.77, 2.89, 0.5)	(0.10, 2.99, 0.5)
Description	On the line passing through mic1 (a_1) and center of the array at height of 0.5m	On the line passing through mic1 (a_1) and center of the array at height of 1m	In-between mic1 (a_1) and 2 (a_2) outside the circumference on which mics lie.	At angle of 75° between mic1 (a_1) and 2 (a_2)	At angle of 88° between mic1 (a_1) and 2 (a_2)
Exact TDs ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(-0.0045, 0.0009, 0.0043, 0, 0)	(-0.0047, 0.0007, 0.0040, -0.0002, -0.0005)	(-0.0026, -0.0026, 0.0031, 0.0007, -0.0005)	(-0.0002, -0.0042, 0.0019, 0.0013, 0)	(0.0008, -0.0045, 0.0011, 0.0013, 0)
TDE_{CC} ($\tau_{10}^{cc}, \tau_{20}^{cc}, \tau_{30}^{cc}, \tau_{40}^{cc}$)	(-0.0032, 0.0020, 0.0040, 0.0013, 0)	(-0.0022, 0.0005, 0.0035, 0.0003, -0.0005)	(-0.0018, -0.0010, -0.0083, 0.0015, 0)	(-0.0003, -0.0045, 0.0018, 0.0040, -0.0008)	(0.0008, -0.0260, -0.0103, 0.0043, 0)
TDE_{OC} ($\tau_{10}^{oc}, \tau_{20}^{oc}, \tau_{30}^{oc}, \tau_{40}^{oc}$)	(0.0270, 0, 0.0010, 0.0283, -0.0003)	(0.0260, 0.0255, 0.0008, 0.0262, 0.0010)	(-0.0098, -0.0082, 0.0018, 0.0005, -0.0035)	(-0.0053, -0.0020, -0.0018, -0.0013, -0.0047)	(-0.0015, -0.0015, -0.0010, 0.0008, 0.0020)
X_{CC} ($x_s^{cc}, y_s^{cc}, z_s^{cc}$)	(0.3033, 0.3796, 0.6088)	(1.2047, 0.4033, 0.7482)	(0.8028, 0.7836, 0.4722)	(-0.0561, -0.5727, 0.7866)	(-1.6690, -5.8132, -1.0062)
X_{OC} ($x_s^{oc}, y_s^{oc}, z_s^{oc}$)	(-0.5057, 0.9780, 0.6664)	(-0.2734, 0.8594, 2.3395)	(0.5888, 1.1176, 1.6950)	(-0.5587, 0.5970, -0.4409)	(-0.0996, -0.0778, 1.1513)
$\epsilon_{cc,3D}$	2.7255	1.8572	1.9492	3.5714	9.1046
$\epsilon_{cc,2D}$	2.7233	1.8401	1.8764	3.5599	8.9792
$\epsilon_{oc,3D}$	3.6434	3.6397	1.9576	2.8123	3.1425
$\epsilon_{oc,2D}$	3.6396	3.3843	1.8301	2.6502	3.0743
Time for CC (Sec)	0.0446	0.0466	0.0450	0.0448	0.0440
Time for OC (Sec)	24.2646	29.0311	27.3696	25.5099	32.8518



(a)



(b)

Figure 7.19: Rhombus Geometry Source Locations Estimated by CC and OC at 4kHz Outdoor using VocoPro system, (a) The x, y coordinates of estimated source Locations, (b) The z coordinate of the estimated source locations.

7.3.4 Observations for the VocoPro Outdoor Experiments

Looking at the results in subsections 7.3.1 to 7.3.3 we can find the numerical facts about the results of VocoPro system outdoor. We present these numerical facts in subsection 7.3.4 and then based on these numerical facts we deduce the observations and present them in subsection 7.3.4

Numerical Facts

1. In outdoor experiments using VocoPro system the CC produces more accurate results than OC for all the geometries.
2. 80% of the time the CC produces more accurate results for the Pyramid geometry by a factor of at least 1.21 than CC i.e. CC 3D-MSE ($\epsilon_{CC,3D}$) is

at least 1.21 times less than OC 3D-MSE ($\epsilon_{OC.3D}$) at 8kHz. For 4kHz case 80% of the times CC produce more accurate results than OC by a minimum factor of 1.19. The runtime for CC is at least 460 times less than the runtime for OC.

3. For Circular geometry 8kHz case , the CC produces more accurate results than OC 60% of the times by a minimum factor of 1.066 while for the 4kHz case CC produces better results than OC by a minimum factor of 1.3 100% of the times. The minimum runtime for OC is at least 340 times more than CC runtime.
4. For Rhombus geometry 8kHz case , the CC produces more accurate results than CC 80% of the times by a minimum factor of 1.18 while for the 4kHz case CC produces better results than OC by a minimum factor of 1.005 60% of the times. The minimum runtime for OC is at least 540 times more than CC runtime.
5. In specific cases, the OC produces better results than CC. However, on average CC is better than OC accuracy-wise in outdoor using VocoPro system. Especially the runtime for CC is much less than the OC runtime.

Outdoor VocoPro Experiments Observations

Looking at the numerical facts in subsection 7.3.4 we can make the following observations for the experiments carried out outdoor using VocoPro system.

1. The CC is producing the best results for all the geometries in outdoor using

VocoPro system with a minimum 3D-MSE ϵ_{CC3D} of 1.85m for Rhombus, 1.10m for Circular and 0.44m for Pyramid Geometry. The reason of which is that there is less distortion in the signal shape because of the absence of reverberations and nearby reflections which is favorable for the CC algorithm.

2. The Pyramid geometry is still the best accuracy-wise and the Rhombus the worst geometry among all the three geometries.
3. Overall, about 75% of the times CC produces more accurate results than OC in outdoor environment using VocoPro system.
4. Overall, the minimum runtime of the OC in any case of the outdoor experiments using VocoPro system is 340 times more than the CC runtime.
5. Overall, the CC algorithm is better both accuracy-wise and runtime than OC in outdoor using VocoPro system.

7.4 An Abstract Observation of All Experimental Results

In this section we provide an abstract observation of all the experiments that we conducted both indoor and outdoor in a tabular form. Table 7.13 summarizes the observations for the indoor experiments while Table 7.14 summarizes those of the outdoor experiments. An entry of OC (or CC) in both Table 7.13 and Table 7.14

means that OC (or CC) is better accuracy-wise than the CC (or OC) algorithm. The runtime for CC is always less than half a second while that for OC is at least ten seconds.

Table 7.13: Abstract Observations (accuracy-wise) of Indoor Results

	RevoLabs System		VocoPro System	
	Less-Reverberant	More-Reverberant	Less-Reverberant	More-Reverberant
Pyramid	OC	OC	CC	CC
Circular	OC	OC	OC	OC
Rhombus	OC	CC	OC	OC

Table 7.14: Abstract Observations (accuracy-wise) of Outdoor Results

	RevoLabs System	VocoPro System
Pyramid	CC	CC
Circular	OC	CC
Rhombus	CC	CC

7.5 Conclusions

This chapter provided in detail the results and analysis of the experiments carried out in an outdoor environment using RevoLabs and VocoPro wireless microphone

systems. The RevoLabs system provides better accuracy than the VocoPro system. It was observed that outdoor experiments produce better results than the indoor. Especially, the CC method provides better accuracy than the OC algorithm. The OC algorithm produces good results in moderately reverberant environment such as the center of the hall than CC. But in dense reverberant environments its accuracy deteriorates, though its accuracy is still better than the CC.

The results of the VocoPro system are better in an outdoor environment than indoor. The limited sensitivity of the VocoPro system limits its applicability for less loud events such as a clap or unloaded gunshot while RevoLabs system provides better results in such situations.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

This thesis focused on the implementation of an impulsive acoustic source localization (IASL) system. The system was implemented using the classical Cross-Correlation (CC) method and a new time delay estimation algorithm called Orthogonal Clustering (OC). To properly analyze, assess and compare the results of these two methods the system implementation was carried out for several practical scenarios.

The system was implemented both for an indoor and outdoor environment in three-dimensions (3D) using two different wireless microphone systems. An attempt was also made to implement the system on a Wireless Sensor Network (WSN) but due to hardware limitations and issues (discussed in chapter 4) we

could not proceed with the implementation. Two different sampling rates of 4kHz and 8kHz were considered for system implementation to check the performance of both of the algorithms at reduced rates. Considering the situation that may arise in practical applications in an indoor environment, experiments were performed both in a less and a more reverberant environment to analyze the effects of the environmental factors on the performance of the system.

Through several experimentations it was found that the Pyramid geometry was producing the best results, accuracy-wise, among all the presented geometries in all the indoor and outdoor scenarios except few special cases. The Rhombus geometry was producing the most erroneous results among the presented geometries. The RevoLabs microphone system was producing relatively accurate results than the VocoPro both in the indoor and outdoor environments.

The OC algorithm was producing relatively accurate results than CC in a reverberant environment. However, it was observed that the OC accuracy deteriorates as the reverberation increases. The CC algorithm outperforms, on the average, the OC algorithm in an outdoor environment. Furthermore, the runtime for OC algorithm is much higher than the CC algorithm. In all the scenarios the CC runtime was always less than a second while the minimum runtime observed for the OC algorithm was more than 10 seconds.

8.2 Future Work and Recommendation

Possible future directions are as follows.

1. The OC algorithm was compared only with the CC algorithm. A comparison is desired between OC and the algorithms developed for indoor environments like Adaptive Eigenvalue Decomposition.
2. Simple Least Squares (LS) method was used to estimate the source locations from over-determined system which does not take into account the uncertainty of the design matrix. Advanced mathematical tools such as Total Least Squares (TLS) can be applied to reduce the effect of uncertainty of both of the design and observation matrix on the estimation process.
3. Furthermore, the OC algorithm uses Minimum Mean Square Estimation (MMSE) or Maximum A Priori (MAP) estimation methods for the channel impulse response (CIR) estimation. TLS can be applied at this stage of the algorithm to reduce the effect of uncertainty of the design matrix on the results.
4. No post processing was carried out on the signals captured through microphones. A noise reducing process may be desired to apply to the acquired signals to improve the results.

REFERENCES

- [1] Frederick V. Hunt, *Origins in Acoustics: The Science of Sound from Antiquity to the Age of Newton*, Yale University Press, 1978.
- [2] G. Becker, “Passive sensing with acoustics on the battlefield,” *Applied Acoustics*, vol. 59, no. 2, pp. 149–178, 2000.
- [3] Davif Muñoz, Frantz Bouchereau, César Vargas, and Rogerio Enriquez-Caldera, *Position Location Techniques and Applications*, Academic Press, 2009.
- [4] T. D. Rossing, *Springer Handbook of Acoustics*, Springer, 2007.
- [5] Tareq Y. Al-Naffouri, Ahmed A. Quadeer, and Giuseppe Caire, “Impulsive noise estimation and cancellation in DSL using orthogonal clustering,” in *IEEE International Symposium on Information Theory Proceedings*, 2011, pp. 2841–2845.
- [6] M. Omer, *Indoor Impulsive Acoustic Source Localization Methods Utilizing Sparse Signal Reconstruction Algorithms*, Master thesis, King Fahd University of Petroleum and Minerals, 2012.

- [7] H E Bass, “Atmospheric absorption of sound: Further developments,” *Journal of the Acoustical Society of America*, vol. 97, no. 1, pp. 680, 1995.
- [8] Jacob Benesty, Jingdong Chen, and Yiteng Huang, *Microphone Array Signal Processing*, vol. 125 of *Springer Topics in Signal Processing*, Springer Berlin Heidelberg, 2008.
- [9] Baruch Berdugo, Miriam A Doron, Judith Rosenhouse, and Haim Azhari, “On direction finding of an emitting source from time delays,” *Journal of the Acoustical Society of America*, vol. 105, no. 6, pp. 3355–3363, 1999.
- [10] A Gelb, *Applied Optimal Estimation*, vol. 64, MIT Press, 1974.
- [11] P. Stoica and Jian Li, “Lecture Notes - Source Localization from Range-Difference Measurements,” *IEEE Signal Processing Magazine*, vol. 23, no. 6, pp. 63 – 66, Nov. 2006.
- [12] Jacob Benesty, Mohan M Sondhi, Yiteng Huang, and Steven Greenberg, “Springer Handbook of Speech Processing,” *Journal of the Acoustical Society of America*, vol. 126, no. 4, pp. 2130, 2009.
- [13] C Knapp and G Carter, “The generalized correlation method for estimation of time delay,” *Analysis*, vol. 24, no. 4, pp. 320–327, 1976.
- [14] Jingjing Jia, Mingjie Liu, and Xiaofeng Li, “Acoustic Localization Algorithm Using Wireless Sensor Networks,” in *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, 2009, pp. 434 – 437.

- [15] Z M Saric, D D Kukolj, and N D Teslic, "Acoustic source localization in wireless sensor network," *Circuits Systems and Signal Processing*, vol. 29, no. 5, pp. 837–856, 2010.
- [16] Akos Ledeczi, Gergely Kiss, Bela Feher, Peter Volgyesi, and Gyorgy Balogh, "Acoustic Source Localization Fusing Sparse Direction of Arrival Estimates," *2006 International Workshop on Intelligent Solutions in Embedded Systems*, pp. 1–13, 2006.
- [17] Lei Liu, Jin-Song Chong, Xiao-Qing Wang, and Wen Hong, "Adaptive Source Location Estimation Based on Compressed Sensing in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 15, 2012.
- [18] I. Bilik, "Spatial Compressive Sensing for Direction-of-Arrival Estimation of Multiple Sources using Dynamic Sensor Arrays," *Ieee Transactions On Aerospace And Electronic Systems*, vol. 47, no. 3, pp. 1754–1769, 2011.
- [19] Volkan Cevher, Richard G Baraniuk, Anna C Gilbert, and Martin J Strauss, "Near-Optimal Bayesian Localization via Incoherence and Sparsity," *Processing*, pp. 205–216, 2009.
- [20] Philip Levis and David Gay, "TinyOS Programming," *ReVision*, vol. 28, pp. 2006, 2009.
- [21] "Latest News," <http://www.tinyos.net/>, 2011.

- [22] Haider Ali, M. S. Sharawi, and T. Y. Al-Naffouri, “Error Sources in COTS WSN Platforms for Impulsive Signal Acquisition Applications,” in *19th International Conference on Telecommunications*, Jounieh, 2012.
- [23] *MPR-MIB Users Manual*, Crossbow Technology, Inc., 2007.
- [24] *MTS/MDA Sensor Board Users Manual*, Crossbow Technology, Inc., 2007.
- [25] *4M-bit AT45DB041B DataFlash Datasheet*, ATMEL Corporation, San Jose, 2005.
- [26] M. R. Azimi-Sadjadi, G Kiss, B Feher, S Srinivasan, and A Ledeczki, “Acoustic source localization with high performance sensor nodes,” *Proceedings of SPIE*, vol. 6562, pp. 65620Y–65620Y–10, 2007.
- [27] *Mica2 Datasheet*, Crossbow Technology, Inc., 2011.
- [28] M Maroti, G Simon, A Ledeczki, and J Sztipanovits, “Shooter localization in urban terrain,” *Computer*, vol. 37, no. 8, pp. 60–61, 2004.
- [29] Murat Arabaci and Robin N Strickland, “Direction of Arrival Estimation in Reverberant Rooms Using a Resource-Constrained Wireless Sensor Network,” *IEEE International Conference on Pervasive Services*, pp. 29–38, 2007.
- [30] Jeremy Elson, Lewis Girod, and Deborah Estrin, “Fine-grained network time synchronization using reference broadcasts,” *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147, 2002.

- [31] J. E. Elson, *Synchronization in Wireless Sensor Networks*, Phd thesis, University of California, 2003.
- [32] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava, “Timing-sync protocol for sensor networks,” *Proceedings of the first international conference on Embedded networked sensor systems SenSys 03*, pp. 138–149, 2003.
- [33] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi, “The flooding time synchronization protocol,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems SenSys 04*. ACM New York, NY, USA, 2004, SenSys ’04, p. 39, ACM Press.
- [34] “Active TinyOS development source tree on Google code,” <http://code.google.com/p/tinyos-main/source/browse/\#svn/trunk/tos/lib\2Fftsp>, 2012.
- [35] *WM-62A Omnidirectional Back Electret Condenser Microphone Cartridge Datasheet*, Panasonic Corporation, New Jersey, 2006.
- [36] *ATmega128/L 8-bit AVR Microcontroller Datasheet*, ATMEL Corporation, San Jose, 2006.
- [37] “TinyOS online documentation,” <http://docs.tinyos.net/tinywiki/index.php/Storage>, 2011.
- [38] *TEP 103: Permanent Data Storage*, TinyOS standard documentation, tinyos-2.x/doc/txt/tep103.txt.

- [39] *RevoLabs Executive HD 8 Channel Wireless Microphone System Data Sheet*, Revolabs Inc., Massachusetts, 2012.
- [40] *VocoPro UHF8800 Owner Manual*, VocoPro Manufactures, La Verne, 2011.
- [41] *DaqBoard/2000 Series User's Manual*, Measurement Computing Corporation, Massachusetts, 2010.
- [42] *USB1608FS User's Guide*, Measurement Computing Corporation, Massachusetts, Feb 2012.
- [43] *Google Earth Software*, Google Inc., Mountain View, California, April 2012.

Vitae

- Name: Haider Ali
- Nationality: Pakistani
- Current Address: P.O. Box 8654, K.F.U.P.M., Dhahran 31261, Saudi Arabia
- Permanent Address: Tairwattoo, Garhi Ismailzay, Garhi Kapura, Mardan 23200, Pakistan
- Telephone: (+966) 531 576 475
- Email: *enrhaiderali@gmail.com*
- Born in Mardan, Pakistan on January 01, 1986
- Received Bachelor of Engineering (B.E.) in Telecommunication Engineering from N-W.F.P University of Engineering & Technology, Peshawar, Pakistan in 2008.
- Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia as a Research Assistant in February 2010.
- Completed Masters in Electrical Engineering in May 2012.