

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2006

Directional speech acquisition using a MEMS cubic acoustical sensor microarray cluster.

Rongrong Hu
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Hu, Rongrong, "Directional speech acquisition using a MEMS cubic acoustical sensor microarray cluster." (2006). *Electronic Theses and Dissertations*. 1752.
<https://scholar.uwindsor.ca/etd/1752>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Directional Speech Acquisition Using A MEMS Cubic Acoustical Sensor Microarray Cluster

by

Rongrong Hu

A Thesis

Submitted to the Faculty of Graduate Studies and Research
through Electrical and Computer Engineering in Partial Fulfillment of the
Requirements for the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada
2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-17078-6
Our file *Notre référence*
ISBN: 978-0-494-17078-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

© 2006 Rongrong Hu

All rights reserved. No part of this document may be reproduced, stored or otherwise retained in a retrieval system or transmitted in any form, on any medium by any means without prior written permission of the author.

Abstract

This thesis presents the design of a directional speech acquisition system using a MEMS cubic acoustical sensor microarray cluster to improve speech intelligibility in a noisy reverberant acoustical environment. In the proposed system, five identical acoustical sensor arrays constitute the five sides of a cubic geometry whereas the other side of the cube is to be used for interconnection and packaging purposes. Each of the sensor microarrays is associated with two beam shapes: a main beam to acquire speech signal from a particular direction and a scanning beam to locate and track a potential speech source. A microelectronics based beam synthesis engine controls the selection of a main beam to acquire speech signals from a particular direction based on the output level of the five scanning beams. In this way the developed system provides an improved reduced noise dynamic directional speech acquisition system covering a 3-D space.

In the proposed system, Individual sensor microarrays—fabricated as a single die using Standard MEMS fabrication technology—are to be assembled in a cubic geometry using a lead-free solder alloy (94%Tin-3.2%Bismuth-2.8%Silver). After 3-D modeling, it has been determined that 3 solder balls each having a diameter of 0.2 mm are needed for 2 plates of the cubic structure to form a permanent 90 degree angle.

An algorithm has been developed based on delay phase-shift concept to realize the scanning beams associated with the sensor microarrays.

An acoustical beam synthesis engine has been developed to analyze the data provided by the scanning beams and steer one of the main beams to the direction of highest intensity level detected by the scanning beams. The

acoustical beam synthesis engine has been developed based on spatial Fourier transform method. Matlab/Simulink realization of the beam synthesis engine matches closely with the design objectives.

To My Parents

Acknowledgments

I would like to express my sincere thanks to Dr. Sazzadur Chowdhury, whose invaluable guidance and encouragement during this research was a source of enormous inspiration.

I would like to thank Mr. Matthew Meloche and Mr. Andrew Tam, graduate students in MEMS research group, University of Windsor, for many technical discussions and interactions in our office.

I would like to express my deepest gratitude to my dear parents, Ms. Shanfang Wang and Mr. Dayang Hu. It's all because of them.

TABLE OF CONTENTS

Abstract	iv
Dedication	vi
Acknowledgments	vii
List of Tables	xi
List of Figures	xii
1 Introduction	1
1.1 Thesis Introduction	1
1.2 Thesis Objectives	6
1.3 Thesis Goals	6
1.4 Thesis Organization	7
2. Challenges in Sound Source Localization and Speaker Tracking	8
2.1 Sound source localization (SSL)	8
2.1.1 Challenges in Microsoft's Approach	9
2.2 Speaker tracking	10
2.2.1 Challenges in speaker tracking	10
2.3 Cylindrical Platform Element Array	10
3. 3-D Cubic Geometry Assembling Method	12
3.1 Introduction	12
3.2 Two interlocking mechanisms: Micro Hinge Structure vs. Solder Joint Structure	13
3.2.1 Micro Hinge Structure	14

3.2.1.1 Discussion of existing hinges structures	14
3.2.2 Solder Ball Based Assembling Solution	16
3.2.2.1 Introduction of the Solder Ball Interconnection	16
3.2.2.2 Solder ball placement technology	16
3.2.2.3 The method of forming a 3-D Cubic Micro Array using Solder Ball based solution	17
3.3 Solder ball specifications	19
3.3.1 Lifting force calculation	19
3.3.1.1 Mathematical model for the lifting force	19
3.3.2 Determination of the composition of the solder ball alloy	21
3.3.2.1 Solder joint introduction	21
3.3.2.2 Solder alloy materials	21
3.3.2.3 Possible lead-free solder alloy choices	23
3.3.3 Solder ball melting simulation	24
3.3.4 Determination of the number of solder ball	25
3.3.4.1 Shear strain investigation	25
3.3.4.2 Determination of the Solder Ball Specifications	27
3.3.4.3 Conclusion	28
4. Beam forming, Beam steering and Beam scanning	29
4.1 Introduction	29
4.2 Beamforming	30
4.2.1 Introduction of beamforming	30
4.2.2 Mathematical model of 3X3 planar array beamforming	32
4.3 Beam steering	33
4.3.1 Introduction of beam steering	33
4.3.2 Mathematical model for beam steering	35
4.4 Beam scanning	37
4.4.1 Scanning beam concept	37
4.4.2 Mathematical Model of the Scanning Beam	37

4.4.3 Polar graphs of the scanning beam	39
5. Beam synthesis engine	46
5.1 Beam synthesis	46
5.1.1 Introduction	46
5.1.2 Simulation for beam synthesis engine in Matlab	47
5.1.2.1 Simulation Specifications	48
5.1.3 Simulation results	51
6. Conclusion	54
6.1 Conclusion	54
References	56
Appendix A: Program 1	60
Appendix B: Program 2	63
Appendix C: Program 3	65
Appendix D: Tensile properties of lead-free solders	113
Appendix E: Program 5	116
VITA AUCTORIS	119

List of Tables

3.1 Physical properties of common solder alloys	22
3.2 Lead-free Solder Alloys	24
3.3 Comparison of the maximum stress-induced strains in solder balls between CUEBGA package and the flip-chip assembly	26
3.4 Radius Vs Number of Solder ball needed	28

List of Figures

1.1 Cross section of the MEMS capacitive microphone	3
2.1 Omni-directional microphones pointing upwards	9
2.2 Unidirectional microphones pointing outwards	9
3.1 IntelliSuite generated 3-D model of the acoustical sensor microarray after simulating the fabrication process steps	13
3.2 Formation of a cubic cluster using planar acoustical sensor microarrays	13
3.3 Proposed hinge-based assembling solution	14
3.4 A MEMS micro hinge	15
3.5 A scissor hinge	15
3.6 Solder ball placement machine	17
3.7 Solder based connectivity approach	18
3.8 Side length of the micro array	19
3.9 Applying lifting force to rotate one plate	20
3.10 Solder ball melting simulation	25
3.11 CUEBGA package	27
4.1 Steerable beam shapes of individual planar sensor microarrays in a cubic geometry	30
4.2 The Delay-and-Sum Beamforming	31
4.3 Planar array in 3-D coordinate	32
4.4 Block Diagram of the developed beam synthesis engine	34
4.5 Scanning beam patterns in both polar graph and wavenumber form	36
4.6 Polar graph of the scanning beam (phase delay=0)	39
4.7 Polar graph of the scanning beam (phase delay=0.7854)	40
4.8 Polar graph of the scanning beam (phase delay=1.0472)	41
4.9 Polar graph of the scanning beam (phase delay=1.5708)	42
4.10 Polar graph of the scanning beam (phase delay=2.3562)	43
4.11 Polar graph of the scanning beam (phase delay=3.1416)	44

4.12 Polar graph of the scanning beam (phase delay=4.7124)	45
5.1 Simulation for beam synthesis engine in Matlab/Simulink	47
5.2 Beam scanning part	48
5.3 Beam synthesis part	49
5.4a Potential Sound Source Waterfall Scope (Input)	51
5.4b Spectrum Waterfall Scope (Output)	51
5.4c Potential Sound Source Waterfall Scope (Input)	52
5.4d Spectrum Waterfall Scope (Output)	52
5.4e Potential Sound Source Waterfall Scope (Input)	53
5.4f Spectrum Waterfall Scope (Output)	53

Chapter 1

Introduction

1.1 Thesis introduction

Hearing aid instruments help to improve the hearing condition of hearing challenged persons. The main causes of hearing loss are: excessive noise (i.e. gun shot, construction), aging, infections, injury to the head or ear, birth defects or genetics, cancer treatment and negative reaction to drugs [1]. A conventional hearing instrument is comprised of a microphone that converts the external sound in to an electrical signal, an amplifier that amplifies the electrical signal and then feeds the amplified signal to a loudspeaker to channel a higher intensity sound to the inner ear. A battery is required to provide the power to the hearing instrument and make amplification process possible [2].

In today's market, there are fundamentally three different types of hearing instruments. The Completely-In-The-Canal (CIC) type of hearing aid instrument fits deeply within the ear canal and is barely visible. The In-The-Ear (ITE) model fits within the outer ear, extending to the ear canal. And the Behind-The-Ear (BTE) model is a small hearing aid instrument that fits behind the ear [3].

The CIC instrument is not recommended for children, because CIC instruments are custom fit, and a child's ear is continually growing. CIC instrument is more expensive than conventional ITE instrument. Besides, the plastic case of the CIC instrument is uncomfortable to wear.

The ITE type can be more easily seen by other people and it can be damaged by earwax and organic contamination.

The BTE instrument is larger than other hearing aids and can be more noticeable. And the ear mold may need to be reconfigured periodically to preserve acoustical seal.

However, these conventional type hearing aid instruments, also amplify the background noise and noise due to multipath distortion associated with an acoustic environment, and ultimately limits the speech intelligibility to a hearing aid instrument user.

Several methods have been explored to eliminate background noise and improve speech intelligibility in noisy and reverberant environments [1][2][5]. A microphone array with directional sensitivity that can be steered electronically depending on suitable speaker tracking or other information-rich algorithms may provide increased speech intelligibility to hearing challenged persons. Single cardioid or supercardioid type microphone [6], combination of one or more directional and omnidirectional microphone in the form of an array where individual microphones are arranged at different locations around the head sphere[7] or in the form of a wearable necklace where the signal from the microphones is being picked up by telecoils [8] have been reported. These systems have been found to be effective in eliminating background noise and improving speech intelligibility to some extent. Single directional microphones cannot provide adequate directionality whereas external connectivity in the cases of wearable necklace is vulnerable to damage and inconvenience to the users. Furthermore, there's a mismatch in the frequency response and sensitivity of the standalone microphones used to construct an array that severely degrades the array response [7].

MEMS technology allows one to fabricate planar acoustical sensor microarray to provide beamforming and beamsteering capability. Such an array can provide variable directional sensitivity to improve speech intelligibility in a noisy reverberant environment. Due to batch fabrication, the possibility of the mismatch in frequency response and sensitivity will be greatly diminished.

There are mainly three types of MEMS acoustical sensors: Capacitive, piezoresistive and piezoelectric. It has been established that capacitive type

acoustical sensors offer high sensitivity to acoustical pressure variations, are not effected greatly by temperature variations, have lower power requirements, and failure modes are easier to detect [7].

A typical MEMS capacitive acoustical sensor is constructed using a low stress thin film diaphragm, an air gap and a backplate. The diaphragm and the backplate form the plates of a parallel type capacitor. The acoustical waves cause the diaphragm to vibrate which results in the changes in capacitance between the diaphragm and the backplate that is converted in to a useful voltage signal using a suitable electronic circuit. A conceptual cross section of a MEMS capacitive type acoustical sensor is shown as figure 1.1.

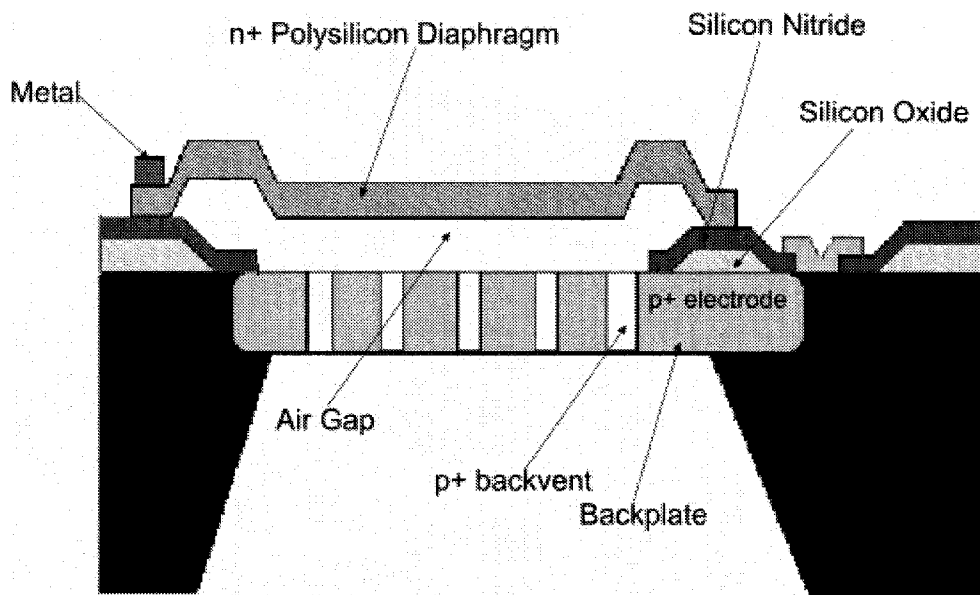


Figure 1.1 Cross section of the MEMS capacitive microphone

Beamforming is a signal processing technique used with arrays of sensors that controls the directionality and the sensitivity of a specific pattern. When receiving acoustical signal, beamforming can increase the gain in the direction of speaker and decrease the gain in the direction of interferences and noise. It is also possible to manipulate the magnitude and phase of the planar acoustical

microphone array to steer the beam to a desired direction. This technique is known as beamsteering technology [7].

In [29-30], two MEMS based acoustical sensor array designs are presented that exploits the techniques of beamforming and beamsteering to improve the directional gain of an acoustical sensor microarray. However, due to large sizes, these arrays are not suitable for ear canal mounted applications. In [31], the design of a MEMS based acoustical sensor microarray has been presented with an appropriate beamforming and beamsteering algorithm pertaining to closely spaced array elements. The $4.6 \times 4.6 \text{ mm}^2$ array is comprised of nine capacitive type acoustical sensors in a 3×3 planar layout. The array is small enough to be placed in the ear canal. However, when mounted in the ear canal, such a planar array is capable of acquiring speech only from the broadside region and the dynamic beam steering capability (look angle range) is limited. Besides, the associated microelectronics circuitry to provide the beamforming and beamsteering engine will occupy a considerable ear canal space and will consume significant amount of battery power comparing to conventional hearing instruments. As a result, a highly crowded ear canal will add more discomfort to a hearing instrument user, shorter battery life, and may give rise to an unhealthy side effect known as the occlusion effect.

Instead of using a single planar microarray, if a number of microarrays, in the formation of a three dimensional cluster have been used, it is possible to extend the acoustical visibility and steerability of the beam over the entire three-dimensional region surrounding a person. The 3-D microarray cluster geometry can be of any suitable geometrical shape, such as a hemispherical dome, a 3-D hexagonal or a cube depending on the permissible MEMS fabrication and assembly technology. Such a cluster of MEMS-based microarrays with all the necessary beamforming and beamsteering microelectronic engine can be body worn and the final signal can be fed to a wireless receiver mounted inside the ear canal using a UHF link. In this way, the ear canal can be freed from crowding whereas advance functionality can be incorporated in a hearing instrument.

In this thesis, the design of a 3-D cubic cluster, consisting of five MEMS-based planar acoustical sensor microarrays has been presented. Such a 3-D sensor microarray cluster can extend the dynamic steerability over the entire region surrounding a person when used in conjunction with appropriate beamforming, beamsteering, and beam synthesis engines.

Although, the 6 faces of the cubic geometry can be fabricated on the same planar substrate, assembling the 3-D cubic geometry from the planar layout is a challenge. The state-of-the-art technology for out-of-the-plane MEMS assembly uses micro-hinges to create joint structure and MEMS fabricated latches can be used to secure them in place. Investigation shows that micro-hinged based assembling approach requires additional MEMS fabrication steps that will make the fabrication process very complex and expensive too.

In this thesis, it has been determined that a reduced complexity 3-D MEMS cubic structure can be realized by using high-performance solder balls. With the development of the microelectronics industry, the solder-ball technology has been matured enough and the reliability and durability issues have been addressed through extensive research and development. Considering the fabrication cost associated with the micro-hinge structure, the solder-ball approach will be low-cost, and thus will be more affordable.

To enable a speaker tracking capability, it is necessary to locate a potential sound source based on, for example, the threshold signal strength criteria. A simple compare and choose circuit can be used for this purpose. It is established that by applying delaying techniques, it is possible to form multiple beams using the same array geometry. Thus, two concurrent beams can be formed for each of the planar array—one of which is the main beam and the other is a scanning beam. The scanning beams from each of the planar array will scan for a potential speech source. After identifying a potential speech source, a beam synthesis engine will determine the possible deployment of a main beam out of the five main beams associated with the five planar arrays, will estimate the direction of arrival, and will signal the beamsteering engine to steer the respective main beam towards that direction. In this way, it will be possible to track a speaker in a

3-D region. The acoustical signal will then be filtered and amplified to a proper range that suitable for the human hearing range.

1.2 Thesis Objectives

Objective of this thesis is to design a 3-D cubic acoustical sensor microarray cluster using 5 planar MEMS-base acoustical sensor microarrays. The cubic MEMS acoustical sensor microarray cluster is to be used for directional speech acquisition in a 3-D space with a speaker-tracking capability. When used in conjunction with appropriate microelectronic circuitry, the device is to provide improved speech intelligibility to a hearing challenged person.

1.3 Thesis Goals

More specific thesis goals are as follows:

1. To develop a technique to assemble and secure five $6 \times 6 \text{ mm}^2$ MEMS beamforming acoustical sensor microarray to realize a cubic shaped cluster geometry.
2. To develop the necessary algorithms to realize five scanning beams associated with each of the sensor microarray to locate a potential speech source in a 3-D space.
3. To develop a beam synthesis engine to focus the main beam associated with respective sensor microarray towards the direction of the speech source dictated by the scanning beam.

1.4 Thesis Organization

This thesis consists of 6 chapters.

Chapter 1 provides an introduction to the area of research carried out in this thesis. It provides the general ideas of the target application—MEMS based hearing instruments.

Chapter 2 explores the state-of-the-art in current acoustical sensor array technology. Both MEMS-and non-MEMS based designs were investigated to determine the scopes of improvements for MEMS based acoustical sensor design technologies.

Chapter 3 begins with 2 possible methods for assembling the 3-D cubic geometry. One is a hinge structure and the other is a solder ball based method. After comparing, solder ball assembling method is chosen. In this chapter, the material and composition of the solder ball and the number of solder balls needed in order to rotate one plate to form a 90 degree angle are also presented.

Chapter 4 provides the mathematical models of beamforming, beamsteering and beam scanning for each planar array, which can provide controlled directional sensitivity and constant beamwidth over the audio frequency range to improve speech intelligibility in a noisy environment.

Chapter 5 investigates the beam synthesis engine. The mathematical model of beam synthesis engine is given. The simulation results generated by Matlab software are also presented.

Chapter 6 presents a concluding summary of the research work carried out, future works.

Chapter 2

Challenges in Sound Source Localization and Speaker Tracking

To track a speaker using a beamforming acoustical sensor array in a 2-D or 3-D space requires the localization of a potential speech source in a particular direction and then steer the beam in that direction to acquire speech from only that direction where as the background noise or multipath distortion coming from other directions are automatically cancels out due to the inherent spatial filtering of the beamformer. Typical sound source localization techniques can be categorized in three groups. One is to maximizing the steered response power (SRP) of a beamformer. The location estimate is derived from a filtered weighted solution of signal. The estimator steers the beam to various locations and search for a peak in output.

The second approach is spectral estimation. It estimates the location of the sound source from spectral correlation matrix of the received signals.

The third one is time-delay estimation (TDE). The location estimation derived from delay estimate. Majority of practical localization systems are TDE-based, Because of its moderate computational cost. However, this method is not robust in reverberate environment, and weak in multipath distortion.

2.1 Sound source localization (SSL)

Microsoft™ Corporation [2] is doing research in sound source localization for circular arrays of directional microphones. They proposed that, for 360° sound

source localization, circular arrays as shown in Figure 2.1 are more appropriate compared to conventional linear arrays, which can only resolve sound location within a small range.

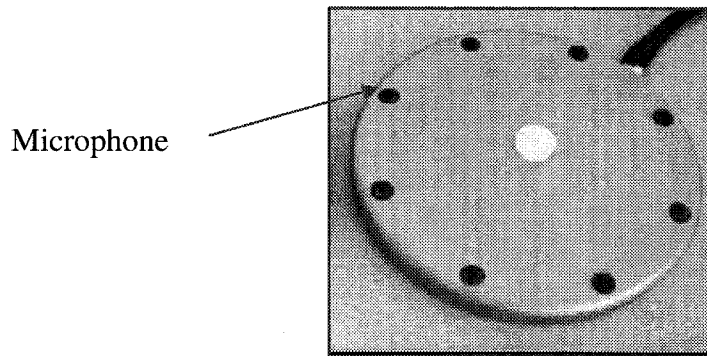


Figure 2.1 Omnidirectional microphones pointing upwards [11]

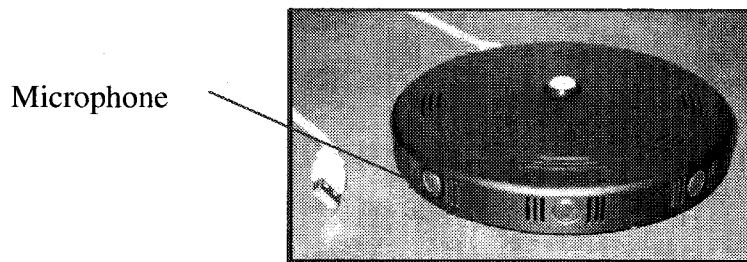


Figure 2.2 Unidirectional microphones pointing outwards [11]

2.1.1 Challenges in Microsoft's Approach

For a typical unidirectional microphone, the phase has reasonably flat response for lower angle, but it becomes highly variable as the angle of arrival increases [2]. Moreover, this phase response is frequency dependent, which is a big problem for circular arrays. For a circular array where the microphones point outward as shown in Figure 2.2, sound arrives at the microphones from directions essentially spaced uniformly over the 360° span.

It may also have problems of the sensitivity mismatch and frequency response mismatch. So the modeling and compensation for the phase shift is not easy.

2.2 Speaker tracking

The goal of a speaker tracking system (speech source localization) is to estimate the target location based on sensor outputs. In the recent research by Palo Alto Research Center [4] and the Microsoft research, they proposed a new way to efficiently organize and utilize network resources for target localization. They separate sensor nodes into different groups in such a way that sensors which jointly provide the best information about a phenomenon should form a group. Sensors which are less informative, or whose data are redundant could be left out. By limiting the collaboration to a small number of sensors in a limited area, communication and computation are made independent of the size of network.

2.2.1 Challenges in speaker tracking

In theory, the algorithm is scalable since all sensors except the leader ones are in the idle state, and free to pick up other tasks. As a consequence, the number of active nodes is proportional only to the number of targets. However, without proper ways of initiating new target tracks and maintaining local collaboration groups, scalability cannot be achieved in practice.

2.3 Cylindrical Platform Element Array

If an array of elements uniformly spaced in a straight line along the surface of a cylindrical platform along the z axis. The radiation pattern produced by such an array can be expressed as:

$$E(\theta, \phi) = \sum_{n=0}^{N-1} EP_n(\theta, \phi) e^{(jnk d \cos \theta + \beta_n)} \quad (2.1)$$

Where $EP_n(\theta, \phi)$ is the individual array element pattern, d is the separation distance between elements, and k is the wavenumber. Assuming that all of the elements have identical element patterns and β_n is the phasing parameter, then we have:

$$E(\theta, \phi) = EP(\theta, \phi)AF(\theta, \phi) \quad (2.2)$$

Where $EP(\theta, \phi)$ represents the element pattern and

$$AF(\theta, \phi) = \left(\frac{\sin\left(\frac{M}{2}\psi_x\right)}{\frac{M}{2}\psi_x} \right) \left(\frac{\sin\left(\frac{N}{2}\psi_y\right)}{\frac{N}{2}\psi_y} \right) \quad (2.3)$$

$AF(\theta, \phi)$ is the array factor. Because in our design elements are identical, the phasing parameter β_n may be used to steer the main beam by selecting its value accordingly.

$$\beta_n = -nkd \cos \theta_0 \quad (2.4)$$

in which θ_0 is the desired steering angle.

Chapter 3

3-D Cubic Geometry Assembling Method

3.1 Introduction

A 3-D model of a MEMS acoustical sensor microarray consisting of 3 x 3 capacitive type acoustical sensors generated by IntelliSuite™ is shown in the Figure 3.1. Detailed design procedures of the planar acoustical sensor microarray consisting of nine capacitive type MEMS-based acoustical sensors are available in [11]. Figure 3.2 shows how a 3-D cubic geometry sensor microarray can be formed by using five of planar arrays as shown in figure 3.1 and the bottom face of the cube could be used for interconnection and packaging purposes. Major challenge here is how to realize (assemble) the 3-D geometry, establish electrical connection for the sides and top arrays. Since MEMS is a planar process, a number of microarrays can be fabricated in the same batch process. Micro hinges having different geometries and force characteristics have already been fabricated successfully. On the other hand, solder balls are in wide application to establish electrical connectivity in microelectronics and the technology is well established and readily available. In this chapter, the possibility of using either the MEMS fabricated micro hinges or a solder ball based assembly method to realize the 3-D cubic geometry has been investigated. Main considerations are: ease of fabrication, ease of assembling, durability, and associated costs.

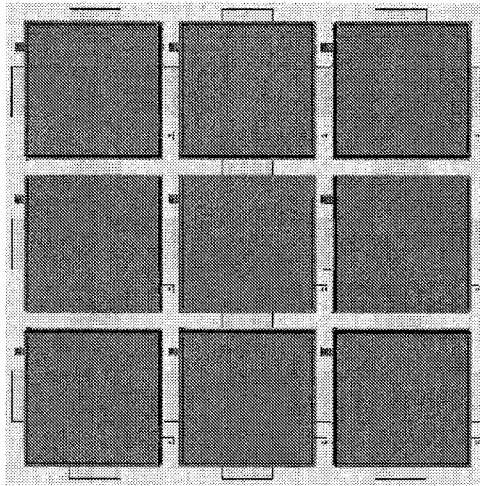


Figure 3.1 IntelliSuite generated 3-D model of the acoustical sensor microarray after simulating the fabrication process steps [6]

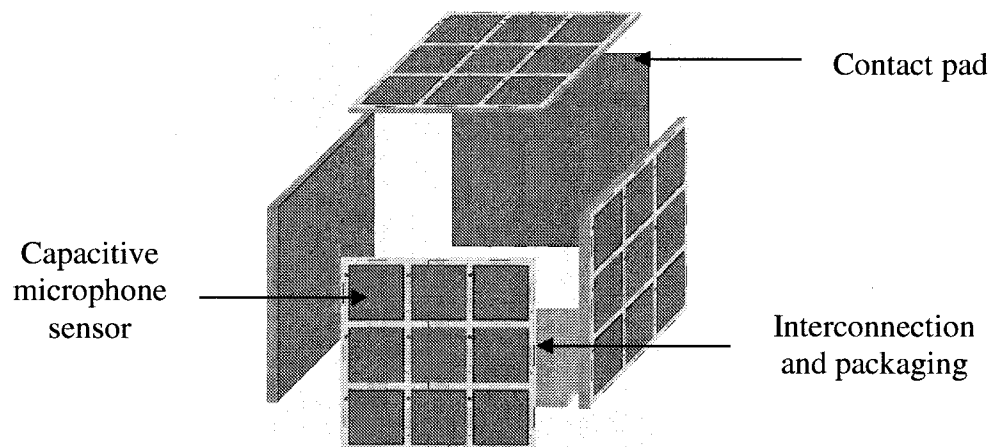


Figure 3.2. Formation of a Cubic Cluster Using Planar Acoustical Sensor Microarrays [6]

3.2 Two interlocking mechanisms: Micro Hinge Structure vs. Solder Joint Structure

A possible assembling method for the cubic geometry composed five acoustical sensor microarrays and an interconnection base is shown in figure 3.3.

Microhinges can be fabricated on the top plane of the wafer after fabrication of the acoustical sensors constituting the arrays. Microhinges will enable to move individual sensor microarrays out of their plane and form a cubic geometry.

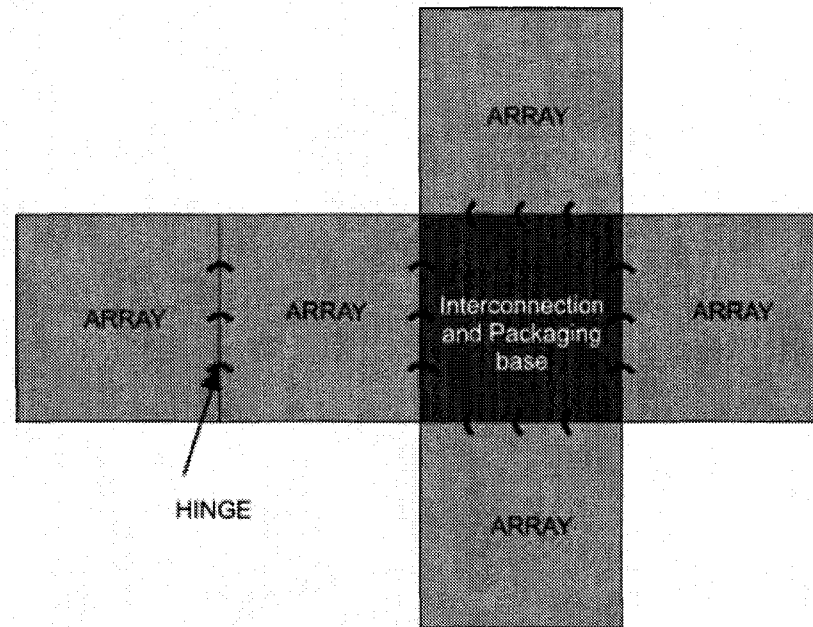


Figure 3.3 Proposed hinge-based assembling solution [1]

3.2.1 Micro Hinge Structure

3.2.1.1 Discussion of existing hinges structures

Micro hinge is a device that can produce out of plan motion without the limitations of a torsion spring. The surface micro-machined non-planar hinges can perform multitask. One common implementation is to use hinge to hold structures that are fabricated in a planar position in a 3-D space. Another important use of hinge is to bind non-planar structures together to form a desired geometry. A common MEMS fabricated micro hinge by Very Small Technologies, Inc. is shown in figure 3.4.

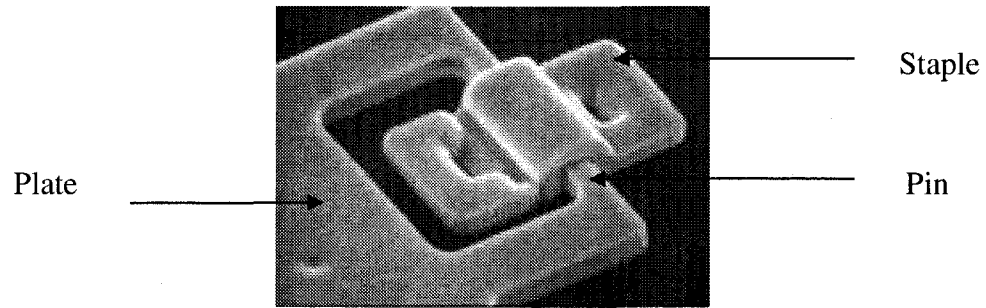


Figure 3.4 A MEMS micro hinge

Two major types of micro hinge structures are: (1) substrate hinge, (2) Scissor hinge. Substrate hinge is constructed out of a pin and a staple. This hinge is often used to support non-planar structures and is commonly used on optical MEMS technologies. Scissor hinge is constructed of interlocking beams, which can have a wider range of motion. Scissor hinges give the flexibility of the hinge released structures. In figure 3.5, the two piece interlock in such a way that the connection can be folded, but in only in one direction (Courtesy to institute of micromachine and microfabrication research, Simon Fraser University.)

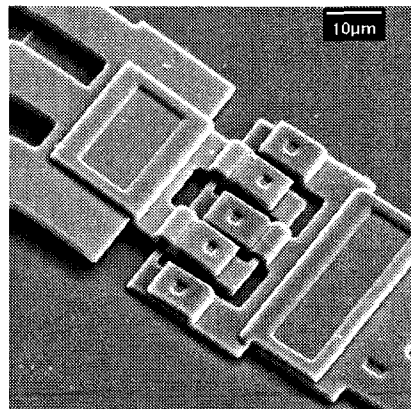


Figure 3.5 A scissor hinge

A major limitation in surface-micro-machining is the limited height of the structure that can be monolithically fabricated. Hinge structure can overcome the limitation by creating moveable components that are nevertheless still attached to the

substrate. However, this will introduce new steps into the fabrication process, which is time-consuming and expensive.

Based on these considerations, the micro hinge based approach has been discarded.

3.2.2 Solder Ball Based Assembling Solution

3.2.2.1 Introduction of the Solder Ball Interconnection

The idea of using solder-balls to assemble a MEMS hearing instrument is introduced from the Ball Grid Array (BGA) package technology [27]. BGA is a surface-mount package that utilizes an array of metal spheres or balls as the method of providing external electrical interconnection.

BGA package is an IC package for active devices, for which the interconnections are made of balls (spheres) of a certain solder alloy. Assembling of BGA onto a specific substrate is more manageable, mainly because the solder structure is economically preferable. Also solder structure is less fragile and easier to handle both before and during assembly. The flexibility in solder ball size and the alloy composition gives MEMS designer more options to solve assembling problem.

3.2.2.2 Solder ball placement technology

As figure 3.6 shows, the solder ball placement machine can be used for special three dimensional packages assembling. The system can place various solder balls from 100 μm up to 760 μm in diameter. It is also possible to place the solders of different solder alloys as well as very high temperature solders. The solder ball bumper uses special bond-heads to single the solder balls out and place them on a specific substrate using a short laser pulse.

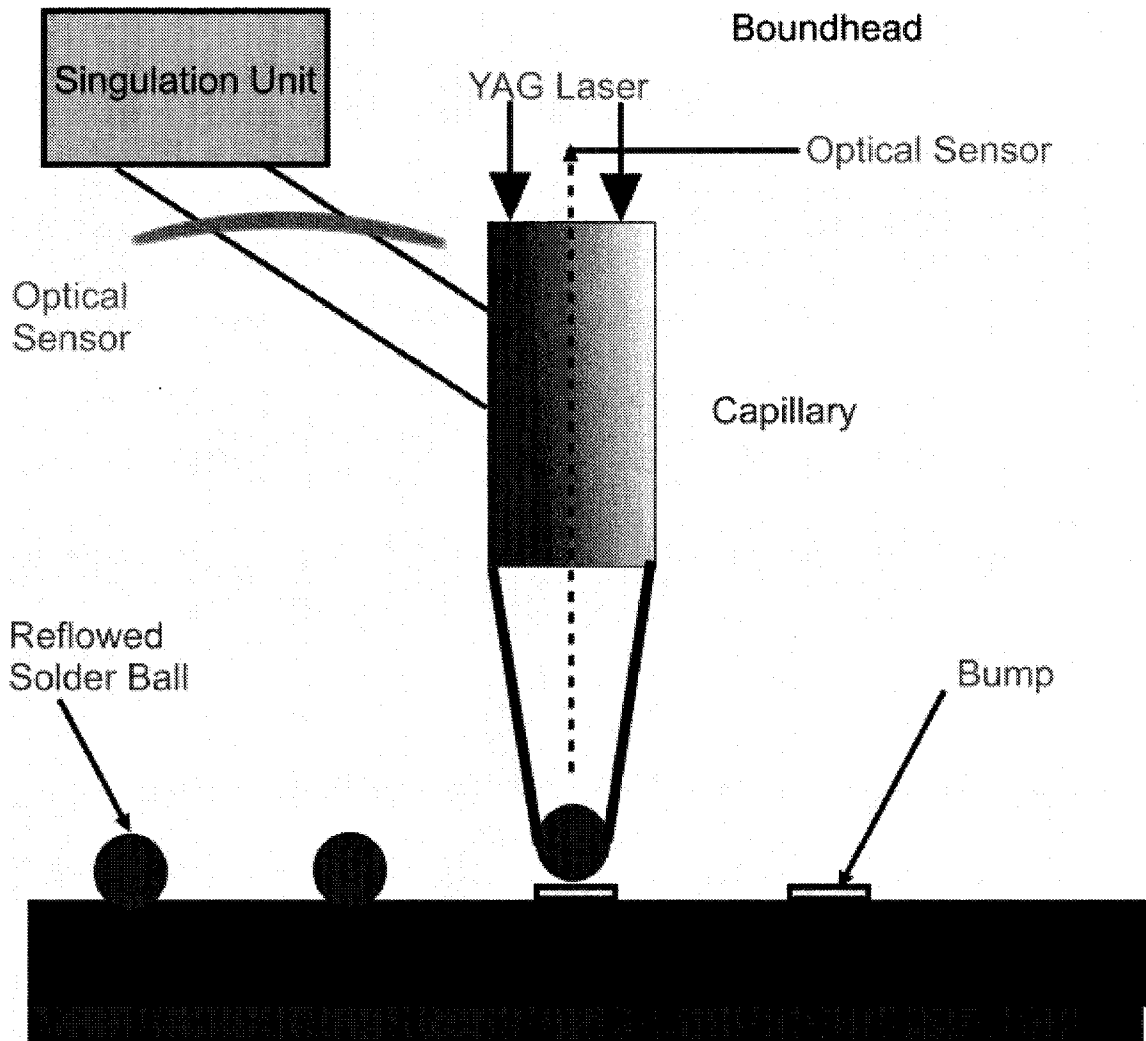
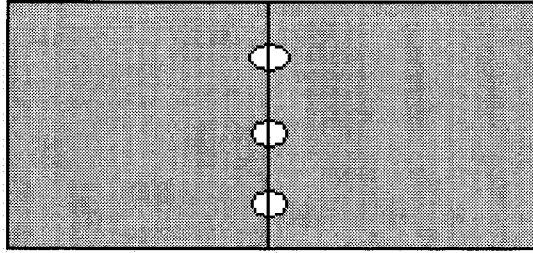


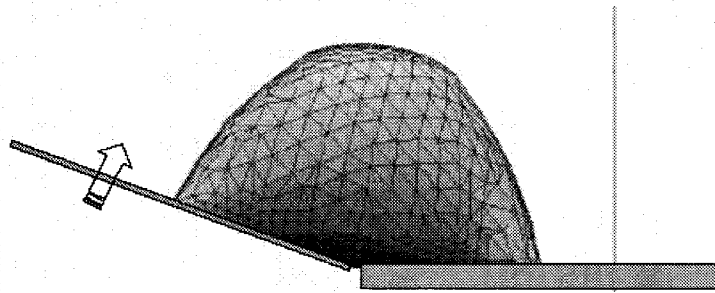
Figure 3.6 Solder Ball Placement Machine

3.2.2.3 The method of forming a 3-D Cubic Micro Array using Solder Ball based solution

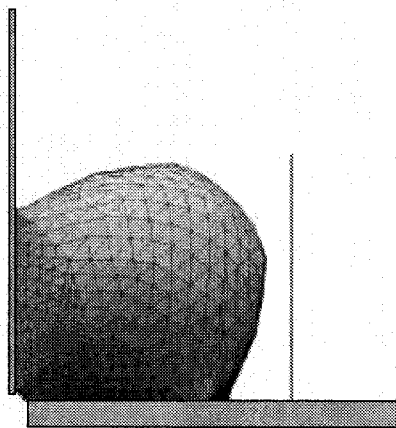
When two plates are in the same planar level, the solder will be put in place of the connecting line in the middle, as shown in the figure 3.7a. Then, the solder will be heated to its melting point. At this time, one of the plates will be lifted up using external force until two plates form a 90 degree angle, as shown in the figure 3.7b. The solder is then let to cool down to form a permanent joint, as shown in the figure 3.7c.



(a)



(b)



(c)

Figure 3.7 Solder Based Connectivity Approach

3.3 Solder ball specifications

3.3.1 Lifting force calculation

3.3.1.1 Mathematical model for the lifting force

Materials: Silicon

Density at 20°C : $2.33 \times 10^3 \text{ Kg} / \text{m}^3$

Thickness: $300 \mu\text{m}$

Side length: $a = 6 \text{ mm}$

Mass of one plate (Acoustical sensor microarray): $M = 2.5164 \times 10^{-5} \text{ Kg}$

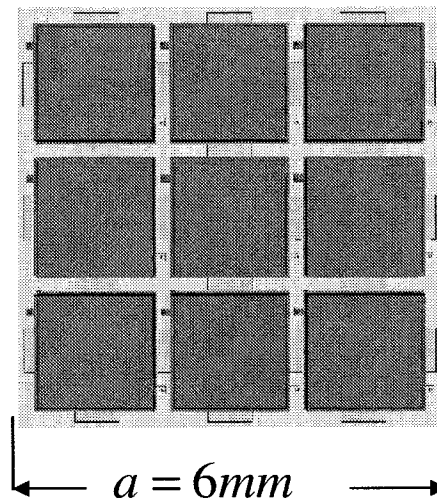


Figure 3.8 Side length of the micro array [1]

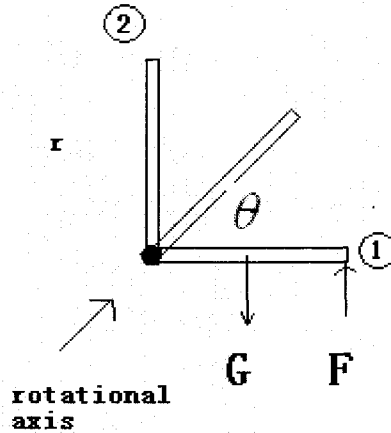


Figure 3.9 Applying lifting force to rotate one plate

As we can see in figure 3.9, τ is the torque; F is the lifting force; θ is the rotational angle; r is the radius of rotation; W is the kinetic energy of rotation; G is the gravity.

The torque applied on the plate is a constant value, and the direction never changes.

Assuming that from position 1 to position 2, the plate moves very slowly, which means the plate's velocity at position 2 equals to zero.

Rotational kinetic energy of the plate:

$$W = \int_{\theta_i}^{\theta_f} \tau d\theta \quad (3.1)$$

$$W = \int_{0^\circ}^{90^\circ} rF \sin(90^\circ) d\theta \quad (3.2)$$

$$W = 6 \times 10^{-3} \times F \Big|_0^{1.57}$$

The change of potential energy from position 1 to position 2:

$$mg\Delta h = 2.5164 \times 10^{-5} \times 9.8 \times 3 \times 10^{-3} = 7.398216 \times 10^{-7} J$$

Due to the Conservation of Energy:

$$mg\Delta h = W \quad (3.3)$$

$$7.398216 \times 10^{-7} = 6 \times 10^{-3} \times F \cdot 0.57$$

Thus, the required magnetic force to pop up one plate to form a 90 degree angle is:

$$F = 78.537 \mu N$$

3.3.2 Determination of the composition of the solder ball alloy

3.3.2.1 Solder joint introduction

Since IBM first commercially introduced flip chip technology in 1960's, solder ball has been used to connect the bond pads to realize flip chip assembling. Assembly operations include handling, placing, fluxing and solder joining [xx]. When introduced as an assembling method in MEMS, the solder joint is evaluated thoroughly in various perspectives. Solder Joint Reliability (SJR) is the ability of solder joints to remain in conformance to their visual, mechanical and electrical specifications over a given period of time, under a specified set of operation conditions. At the same time, we have to put the solder joint failures into consideration, when we choose solder ball alloy materials. Solder joint failures occur for 4 main reasons: 1. poor solder joint design; 2. poor solder joint processing; 3. solder material choosing; 4. excessive stresses applied to the solder joints.

3.3.2.2 Solder alloy materials

Lead and its alloys, due to their low melting temperatures and wide availability, are the most commonly used solder materials. However, with the growing concern regarding environmental pollutants, lead free solder alloys are preferred.

We need to find lead alternatives which meet the following criteria: 1.The element should have no negative impact on both environment and human beings; 2. Sufficient quantities of such materials must be available; 3. Melting temperatures preferably below 200°C; 4.Alloy materials should have similar thermal and electrical conductivity; 5.Adequate joint strength and thermal fatigue resistance; 6. Low cost.

Table 3.1 shows the physical properties of Sn-Pb alloys and Sn/Ag/Cu alloys. And Appendix D shows the tensile properties of lead-free solders.

Table 3.1 Physical properties of common solder alloys

Alloy	Sn/Ag/Cu Lead Free	63Sn95Pb Standard	5Sn95Pb High Lead	63Sn37Pb Low Alpha	63Sn37Pb Ultra low Alpha
Alloy Composition	Sn 95.5% Ag 3.5% Cu 1.0%	Sn 63% Pb 37%	Sn 5% Pb 95%	Sn 63% Pb 37%	Sn 63% Pb 37%
Melting	217°C	183°C	300°C-314°C	183°C	183°C
Re-flow temperature	235°C-255°C	215°C-225°C	330°C-340°C	215°C-225°C	215°C-225°C

In [27]. it has been stated that the wettability of conventional eutectic tin-lead solder is much higher than that of solders based on tin-bismuth system. The authors suggested that the composition of the solder alloy is Sn-58.8%Bi-2%Ag. In their experiment, they proved that H-flux has a large effect in making the solder more spreadable on a copper plate. However, the H-flux contains a large amount of chlorine, which is avoided often recently in electronic fabrication industry. So FR-type flux came to our attention. By varying the composition of silver into the alloy, the spreading area of the solder can be controlled. Also, the strength of a solder ball depends on the quantity of Bismuth. Bismuth separates the joints from to the surface of the joint, which weakens the joint. However, for small amount of Bismuth, the effect will be weak, and will not lead to a dramatic loss of strength for the joint.

3.3.2.3 Possible lead-free solder alloy choices

A solder composition of Tin-antimony—95% by weight tin and 5% by weight antimony—shows a higher melting points, approximately 232 degree C., than that of the traditional tin-lead solder compositions that can be altered in the range around 182 degree C.

Another choice is a composition of tin-antimony, zinc and silver, typically 95% by weight tin, 3% by weight antimony, 1.5% by weight zinc and 0.5% by weight silver. This composition results in a lower melting temperature around 215 degree C., which is better than the previous one; however, as zinc tends to create air pockets, an unstable solder results.

Another choice is a solder composition of tin-copper and silver, 95.5% by weight tin, 4% by weight copper, and 0.5% by weight silver. But this solder composition has an undesirable high melting point around 226 degree C. It's hard to realize the high temperature to assemble our MEMS cubic geometry.

It's been experimentally proved that, a lead-free solder composition of 94% by weight tin, 3.2% by weight bismuth, and 2.8% by weight silver has advantages over the other considerations. When the solder composition was heated and reached a melting transition from solid state to a plastic-like state at a temperature of 217 degree C. It passed from a soft plastic-like state at 225 degree C, and to a liquid state at 235 degree C. The tensile strength was found to be 7,400 psi, and a wire of the composition 0.124 inches in diameter can sustain a single shear load of 110 pounds.

Tin base lead-free solder composition containing bismuth, silver is basically our choices for the solder alloy materials.

A lead-free solder composition of 94% by weight tin, 3.2% by weight bismuth, and 2.8% by weight silver has been chosen.

Table 3.2 shows the most popular lead-free alloys practically used by various companies. Lead-free alloys have higher melting point than eutectic lead-tin solders.

Table 3.2 Lead-free Solder Alloys

Alloys	Melting Temp(°C)	Industry	Company
Sn-Ag(tin-silver)	221-226	Automobile	Visteon
Sn-Ag-Bi(tin-silver-bismuth)	206-213	Military/Aerospace	Panasonic
		Consumer	Hitachi
Sn-Ag-Bi-Cu(tin-silver-bismuth-copper)		Military/Aerospace	Panasonic
Sn-Ag-Bi-Cu-Ge(tin-silver-bismuth-copper-germanium)		Consumer	Panasonic
Sn-Ag-Bi-X(tin-silver-bismuth-X)	206-213	Consumer	Panasonic
Sn-Ag-Cu(tin-silver-copper)	217	Automobile	Panasonic
		Telecom	Nokia
			Nortel
			Panasonic
		Toshiba	
Sn-Bi(tin-bismuth)	138	Consumer	Panasonic
Sn-Cu(tin-copper)	227	Consumer	Panasonic
		Telecom	Nortel
Sn-Zn(tin-zinc)	198.5	Telecom	NEC
		Consumer	Panasonic
			Toshiba

3.3.3 Solder ball melting simulation

Ansys™ finite element modeling tool has been used to simulate the melting of a solder ball of desired composition. The simulation result is shown in Figure 3.10.

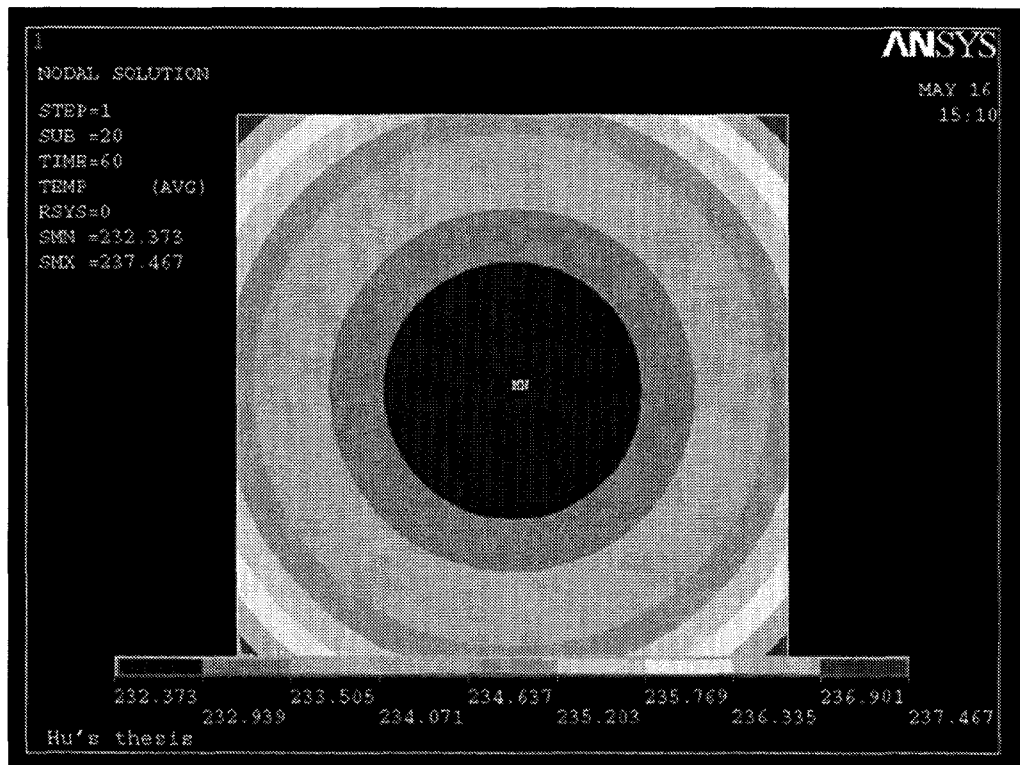


Figure 3.10 Solder ball melting simulation

Element Death is used to model melting of a material. The model is an infinite long rectangular block of material. It will be subject to convection heating which will cause the block to melt.

Boundary conditions: For thermal problems, constraints can be in the form of temperature, heat flow, convection, heat flux, heat Generation, or radiation.

In this model, all external surfaces of the material will be subject to convection with surrounding temperature of 328K (bulk temperature).

3.3.4 Determination of the number of solder ball

3.3.4.1 Shear strain investigation

The following general formula approximates the maximum possible shear strain in the flip-chip-on-board construction:

$$\text{ShearStrain, } \gamma = \Delta T (CTE_{die} - CTE_{board}) \frac{DNP}{H} \quad (3.4)$$

CTE_i= coefficient of thermal expansion for the die

ΔT=temperature range of the stress cycle

DNP= distance from the neutral point of the solder joints

H=die-to-board standoff height

In a ball grid array (BGA) package, the diameter of the solder ball normally range from 0.10 mm to 0.85mm.

Table 3.3 shows the comparison of the maximum stress-induced strains in solder balls between cavity-up enhanced ball grid array package and the flip-chip assembly.

Table 3.3 Comparison of the maximum stress-induced strains in solder balls between CUEBGA package and the flip-chip assembly

(MPa)	ε11 (x0.001)	ε22 (x0.001)	ε33 (x0.001)	ε12 (x0.001)	ε13 (x0.001)	ε23 (x0.001)	Plastic Equivalent Strain
CUEBGA (Polyimide Adhesive)	0.92	0.93	2.17	1.55	3.45	3.31	0.66
CUEBGA (3M Adhesive)	1.91	1.73	1.18	2.92	2.52	2.45	0.47
Flip-Chip Assembly	2.81	2.82	3.65	3.69	12.45	12.88	7.14

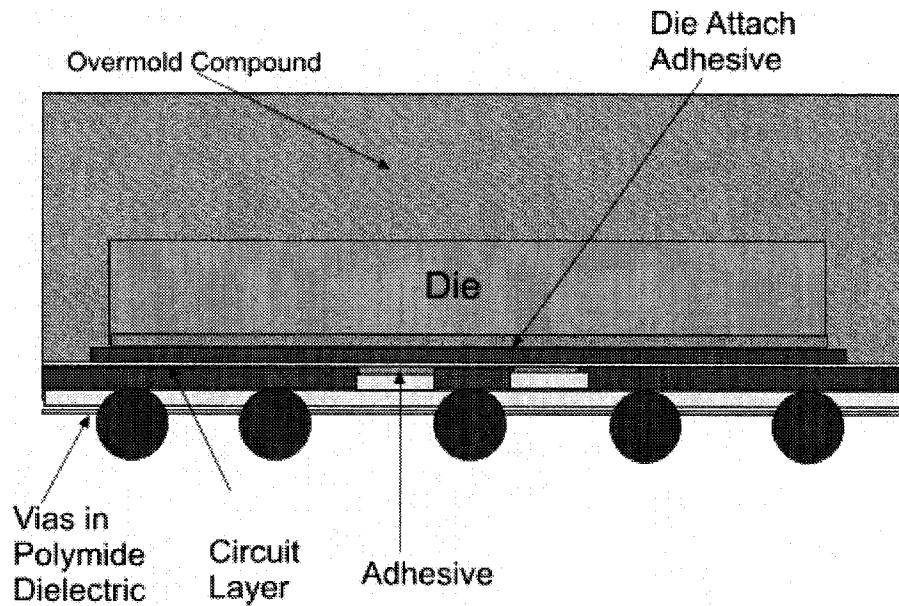


Figure 3.11 CUEBGA package

Overmold (above die): 500um

Die: 250um

Die Attach Adhesive: 25um

Copper Stiffener: 125um

Flex Substrate: 50um

Lamination Adhesive: 50um

Solder Ball Diameter: 800um

3.3.4.2 Determination of the Solder Ball Specifications

First, we choose a diameter of 0.2mm for the solder ball (side length of one plate is 6 mm);

Next, we use the minimum Stress-Induced Strains $0.92 \times 0.001 \text{ MP}$

Thus, the minimum supporting force that one solder ball can provide is approximately:

$$\pi \times (0.1 \times 0.001)^2 \times 0.001 \times 1000000 = 28.888 \mu\text{N}$$

Compared with the determined lifting force of $78.537 \mu N$, 3 solder balls are enough for two plates to form a 90 degree angel.

Table 3.4 Radius Vs number of solder ball needed

	Radius of solder ball (unit: mm)							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Minimum Number of Solder ball	12	3	2	1	1	1	1	1
Supporting Force (unit:)	7.222	28.888	64.998	115.552	180.55	259.992	353.878	462.208

3.3.4.3 Conclusion

We designed a solder ball based assembling method to realize the joint connectivity. From our mathematical model, we chose 94%Tin-3.2%Bismuth-2.8%Silver as the solder material, and 0.2mm as the radius of each solder ball, then for every joint connection, 3 solder balls are needed to form a 90 degree angle.

Chapter 4

Beamforming, Beam steering and Beam scanning

4.1 Introduction

The target application of the developed cubic sensor microarray cluster is a hearing instrument to improve speech Intelligibility in a noisy reverberant environment to a hearing challenged person. The beam scanning sub-system will localize a potential sound source. Then the specific main beam will be steered to that position by the beam steering techniques. By applying delaying techniques, it's possible to form multiple beams using the same array geometry. In our design, two beams are formed for each of the planar array. One of them is a main beam; the other is a scanning beam. As a consequence, we have five pairs of main beam and scanning beam located on the top and the four side plates of the cubic geometry.

Then the acoustical signal will be modified using advanced DSP algorithms to be suitable for a human hearing range.

Figure 4.1 provides a conceptual 3-D view of the steerable beam shapes of individual planar sensor microarrays in a cubic geometry.

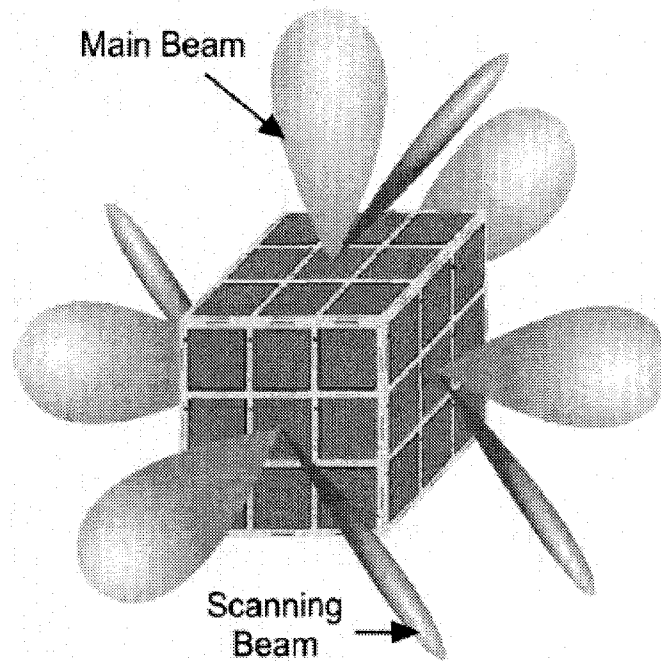


Figure 4.1 Steerable beam shapes of individual planar sensor microarrays in a cubic geometry [1]

4.2 Beamforming

4.2.1 Introduction of beamforming

Acoustical beamforming is a spatial filtering technique which operates on the output of an array of acoustical sensors in order to enhance the amplitude of a coherent wavefront relative to background noise and directional interface.

The goal of beamforming is to sum multiple sensor elements in phase to achieve a narrower response in the potential sound source direction.

By delaying and adding shaded outputs from an array of sensors, we can form time-domain beamforming. Figure 4.2 is the block diagram of a filter and sum beamforming. The shading of the sensor output is used to improve the spatial response of the beam, which is similarly to “Windowing” in DSP theory. Each beam is formed by delaying and summing the outputs from individual sensor elements.

From energy point of view, we can also say that beamforming can direct the majority of signal energy transmitted from a group of transducers in a chosen angular direction.

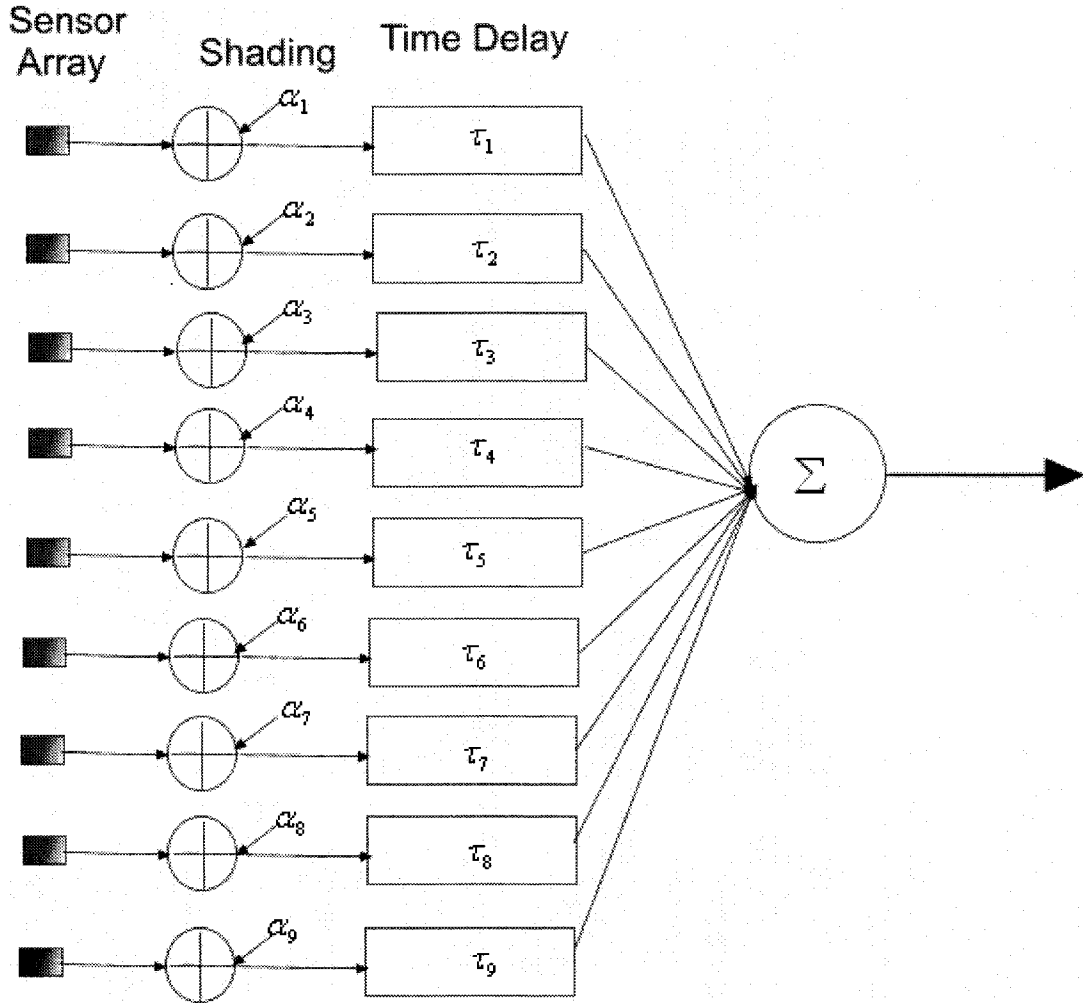


Figure 4.2 Delay-and-Sum Beamforming

4.2.2 Mathematical model of 3X3 planar array beamforming

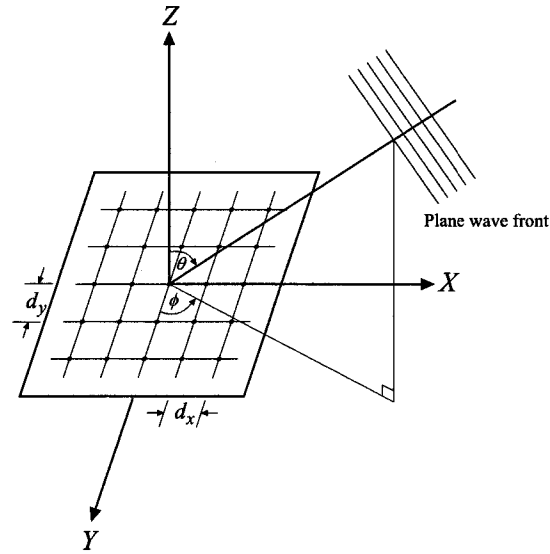


Figure 4.3 Planar array in 3-D coordinate

In the proposed design, each planar microarray is a 3X3 acoustical micro sensor. The array factor of an array of sensors is a function of the number of elements, their geometrical arrangement, their relative magnitudes, their relative phases, and their spacing. If the elements have identical amplitudes, phases and spacing, the array factor will be in a simpler form.

For small uniformly paced array elements, the array factor $AF(\theta, \phi)$ with a plane wave incident at an angle (θ, ϕ) as shown in figure 4.3 can be approximated as:

$$AF(\theta, \phi) = \left(\frac{\sin\left(\frac{M}{2}\psi_x\right)}{\frac{M}{2}\psi_x} \right) \left(\frac{\sin\left(\frac{N}{2}\psi_y\right)}{\frac{N}{2}\psi_y} \right) \quad (4.1)$$

$$\psi_x = kd_x \sin \theta \cos \phi + \beta_x \quad (4.2)$$

$$\psi_y = kd_y \sin \theta \sin \phi + \beta_y \quad (4.3)$$

$$\beta_x = -kd_x \sin \theta_0 \cos \phi_0 \quad (4.4)$$

$$\beta_y = -kd_y \sin \theta_0 \sin \phi_0 \quad (4.5)$$

where k is the wavenumber expressed as ω/c , ω is the angular frequency in rad/s and c is the velocity of sound. The parameters d_x and d_y are the inter-element spacing along the x and y axis, β_x and β_y are the phase shifts applied to the output of each array element to steer the beam to a particular direction; and M and N are the number of transducers in x and y directions, respectively.

4.3 Beam steering

4.3.1 Introduction of beam steering

Beam steering is used to steer the main beam to the potential sound source in order to track the speaker.

Array physical dimensions: 6 x 6 mm²

Inter-element spacing: 100 μm

Scanning Frequency: 15 KHz

Bandwidth between -3dB points: 30 Hz to 10,000 Hz

β_x and β_y are the phase shifts that must be applied to the output of each array element. By changing the value of β_x and β_y , we will be able to steer a beam to a particular direction.

As figure 4.4 shows, five scanning beams on each side of the cubic geometry will scan automatically by applying different phase delay. When this is a potential sound source detected by a specific scanning beam, the acoustical signal will be measured and stored in the memory.

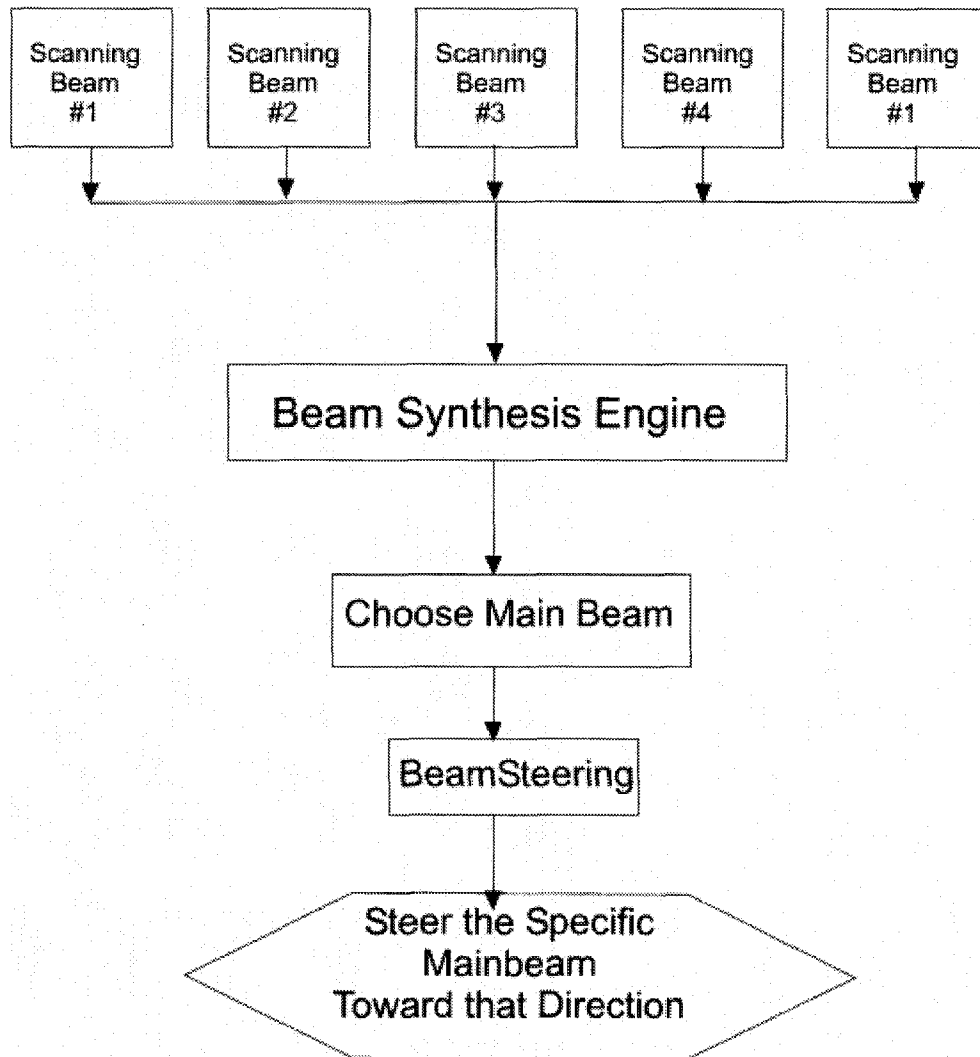


Figure 4.4 Block Diagram of the developed beam synthesis engine

Then the beam synthesis engine will choose the associated main beam, and by applying beam steering algorithms to the output of the array, the specific mainbeam will be steered to that position and capture the sound source. After localizing the speaker, beam synthesis engine will modify the signal to make it suitable for the human hearing range.

4.3.2 Mathematical model for beam steering

We wish to electronically steer the array pattern towards some other direction ϕ_0 , without physically rotating it. The corresponding wavenumber at the desired look-direction will be:

Steering phase:

$$\psi_0 = kd \cos \phi_0 \quad (4.6)$$

Such steering operation can be achieved by wavenumber translation in ψ -space that is replacing the broadside pattern $A(\psi)$ by the translated pattern $A(\psi - \psi_0)$.

Thus, we define:

Steered array factor:

$$A'(\psi) = A(\psi - \psi_0) \quad (4.7)$$

The translated wavenumber variable,

Steered wave number:

$$\psi' = \psi - \psi_0 = kd(\cos \phi - \cos \phi_0) \quad (4.8)$$

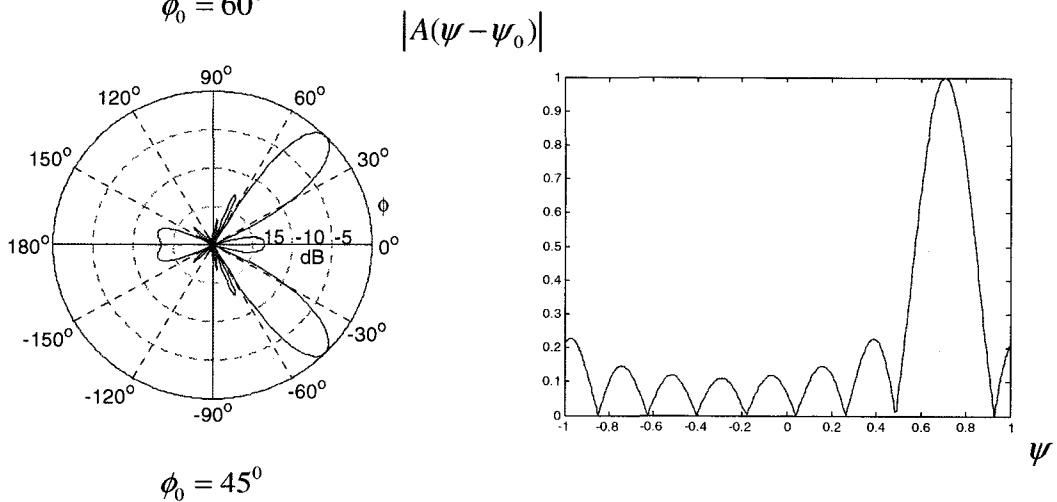
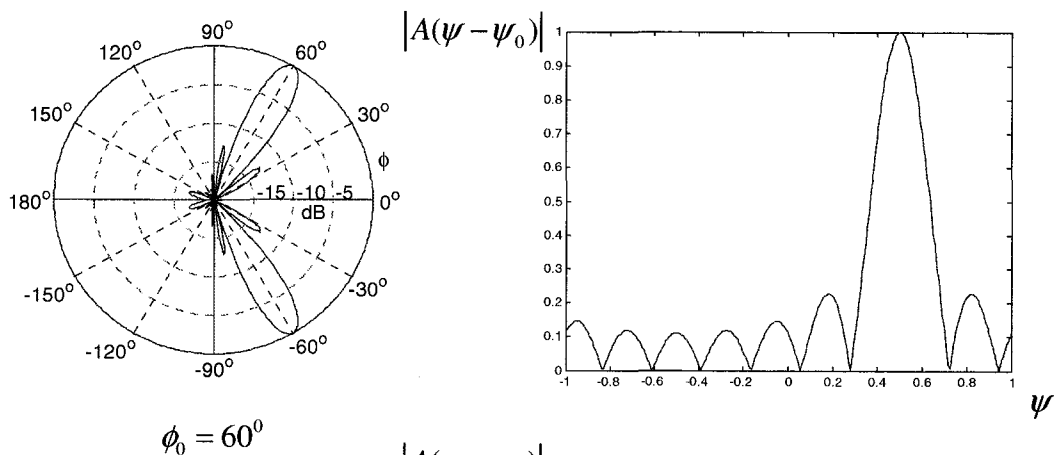
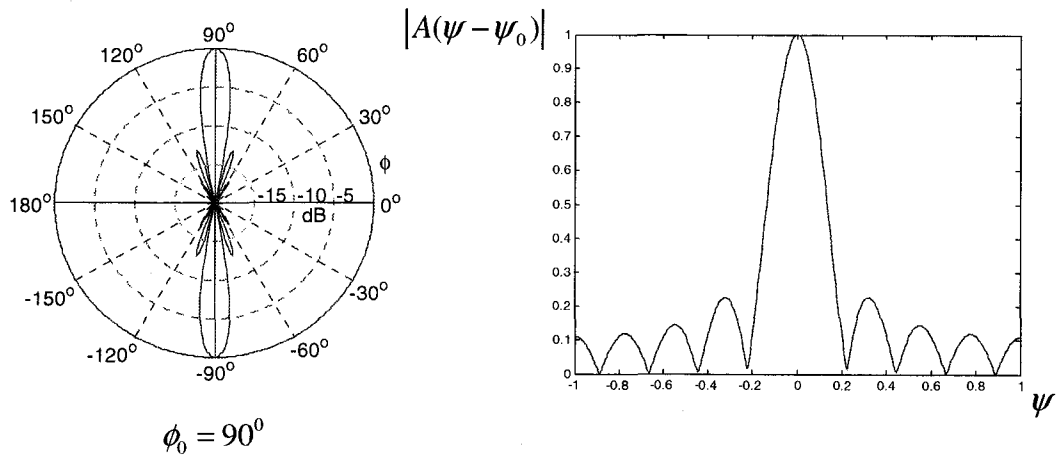


Figure 4.5 Scanning beam patterns in both polar graph and wavenumber form

4.4 Beam scanning

4.4.1 Scanning beam concept

It is theoretically possible to construct more than one beam from an array of discrete elements by varying the electronic delays. Thus the same planar array can be used to form a scanning beam using a different delay in addition to forming the main beam.

In the proposed scheme, the scanning beam will rotate dynamically in a pre-specified region using microelectronic based beam steering engine. If the sound source it captures is greater than or less than a threshold value (e.g. 20 db and 95 db), then the “position” of the scanning beam will be stored along with its spatial position, and associated phase delay. Then the information will be sent to the beam synthesis engine. The beam synthesis engine will then determine the appropriate main beam and steer the main beam in that direction using the phase delay stored in the memory.

4.4.2 Mathematical Model of the Scanning Beam

There are $2N_x + 1 = 3$ rows of elements, each row parallel to the y axis, with common spacing d_x between rows. Let each row contain $2N_y + 1 = 3$ elements whose common spacing is d_y .

By the element, let the coordinate be:

$$x_m = md_x \quad (4.9)$$

$$y_n = nd_y \quad (4.10)$$

In which $-N_x \leq m \leq N_x$, and $-N_y \leq n \leq N_y$. With this notation, the array factor can be written as:

$$A(\theta, \phi) = \sum_{m=-N_x}^{N_x} \sum_{n=-N_y}^{N_y} e^{j \sin \theta (md_x \cos \phi + nd_y \sin \phi)} \quad (4.11)$$

The array factor for a planar array is the product of the array factors for two linear arrays: One is laid out along the x axis, and the other is laid out along the y axis.

Then the array factor can be written as:

$$A(\theta, \phi) = A_x(\theta, \phi) * A_y(\theta, \phi) \quad (4.12)$$

In which,

$$A_x(\theta, \phi) = \sum_{m=-N_x}^{N_x} e^{j m k d_x \sin \theta \cos \phi} \quad (4.13)$$

$$A_y(\theta, \phi) = \sum_{n=-N_y}^{N_y} e^{j n k d_y \sin \theta \cos \phi} \quad (4.14)$$

Then we have,

$$A_y(\theta, \phi) = \left(\sum_{m=-N_x}^{N_x} e^{j m k d_x \sin \theta \cos \phi} \right) \left(\sum_{n=-N_y}^{N_y} e^{j n k d_y \sin \theta \cos \phi} \right) \quad (4.15)$$

In the developed design, A_x gives a pattern which consists of a conical main beam and sidelobes, symmetric about the x axis. The main beam of A_x makes an angle θ_x with the positive x axis which satisfies the relation:

$$k d_x \cos \theta_x = k d_x \sin \theta \cos \phi \quad (4.16)$$

So we have,

$$\cos \theta_x = \sin \theta \cos \phi \quad (4.17)$$

Similarly, the factor A_y gives a pattern which consists of conical main beams and sidelobes, symmetric about the y axis. The main beam of A_y makes an angle θ_y with the positive y axis which satisfies the relation:

$$\cos \theta_y = \sin \theta \sin \phi \quad (4.18)$$

As a consequence, we get the mathematical model of the scanning beam.

$$A = \left(\sum_{-N_x}^{N_x} e^{jm(kd_x \cos \theta_x)} \right) \left(\sum_{-N_y}^{N_y} e^{jn(kd_y \cos \theta_y)} \right) \quad (4.19)$$

4.4.3 Polar graphs of the scanning beam

These polar graphs of the scanning beam based on the above discussed mathematical model are generated using Matlab Software.

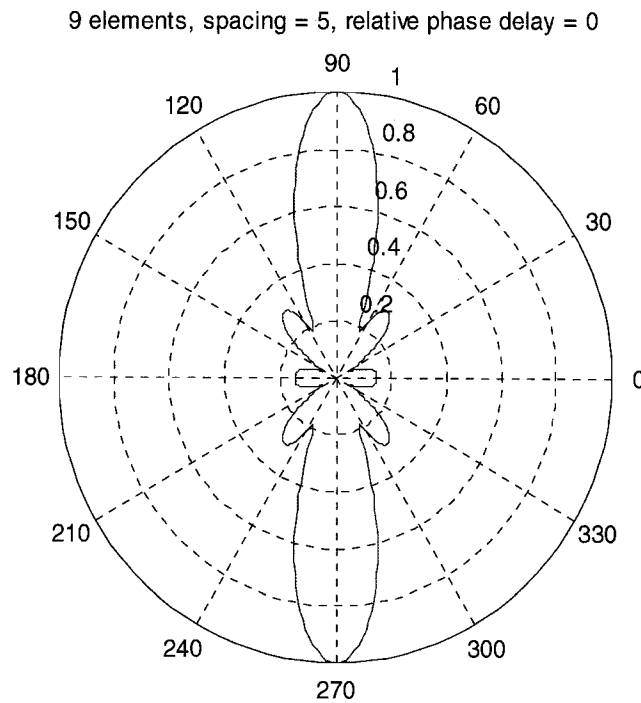


Figure 4.6 Polar graph of the scanning beam (phase delay=0)

9 elements, spacing = 5, relative phase delay = 0.7854

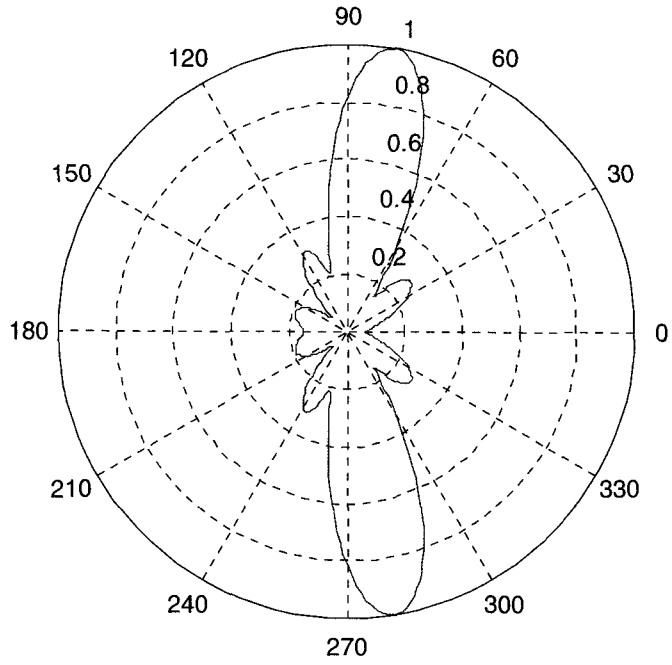


Figure 4.7 Polar graph of the scanning beam (phase delay=0.7854)

9 elements, spacing = 5, relative phase delay = 1.0472

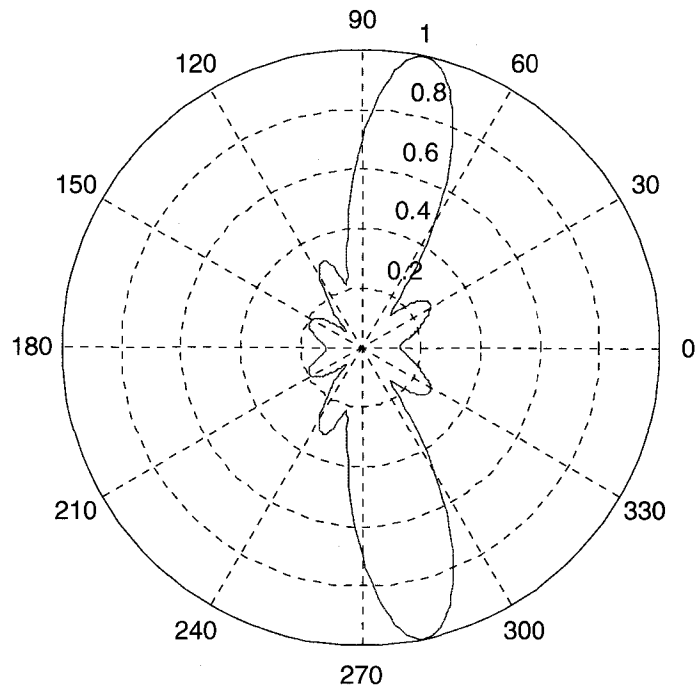


Figure 4.8 Polar graph of the scanning beam (phase delay=1.0472)

9 elements, spacing = 5, relative phase delay = 1.5708

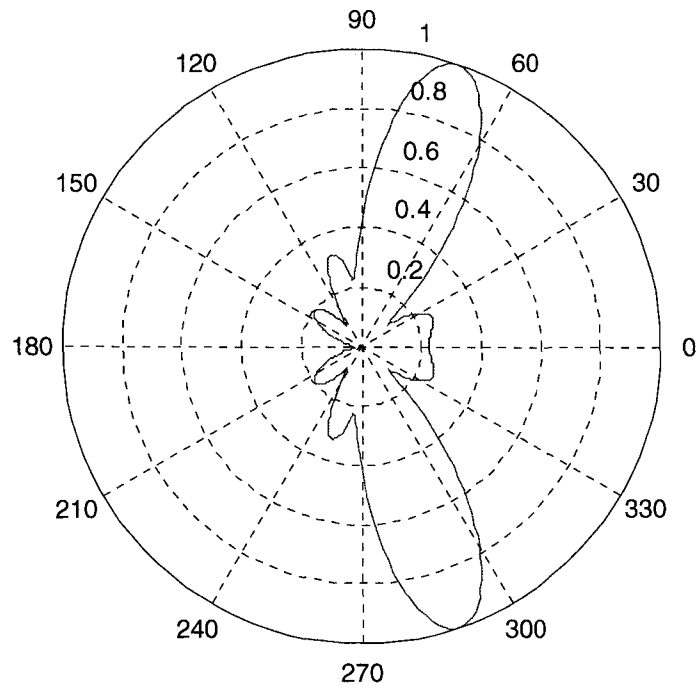


Figure 4.9 Polar graph of the scanning beam (phase delay=1.5708)

9 elements, spacing = 5, relative phase delay = 2.3562

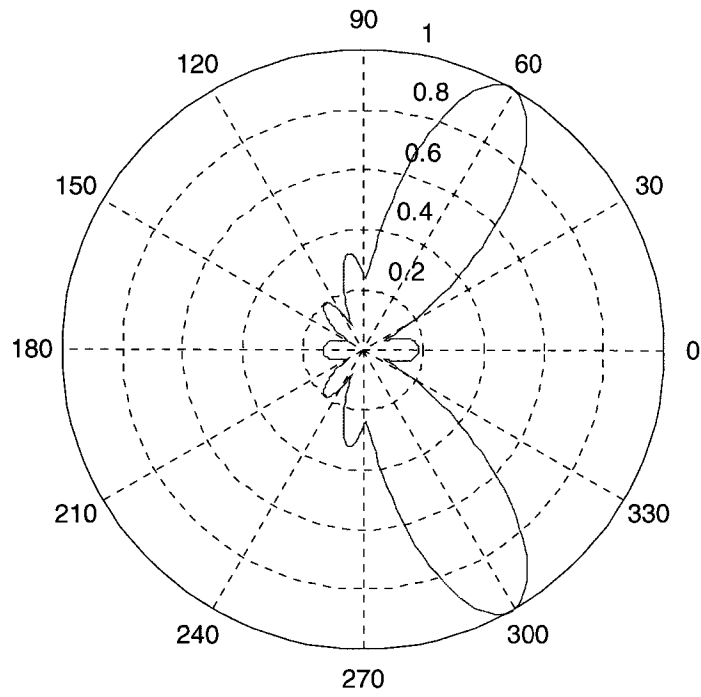


Figure 4.10 Polar graph of the scanning beam (phase delay=2.3562)

9 elements, spacing = 5, relative phase delay = 3.1416

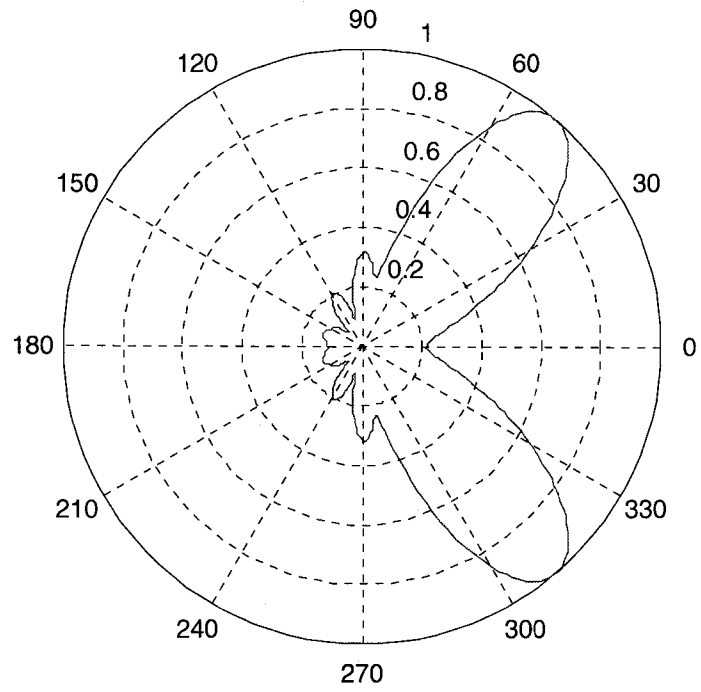


Figure 4.11 Polar graph of the scanning beam (phase delay=3.1416)

9 elements, spacing = 5, relative phase delay = 4.7124

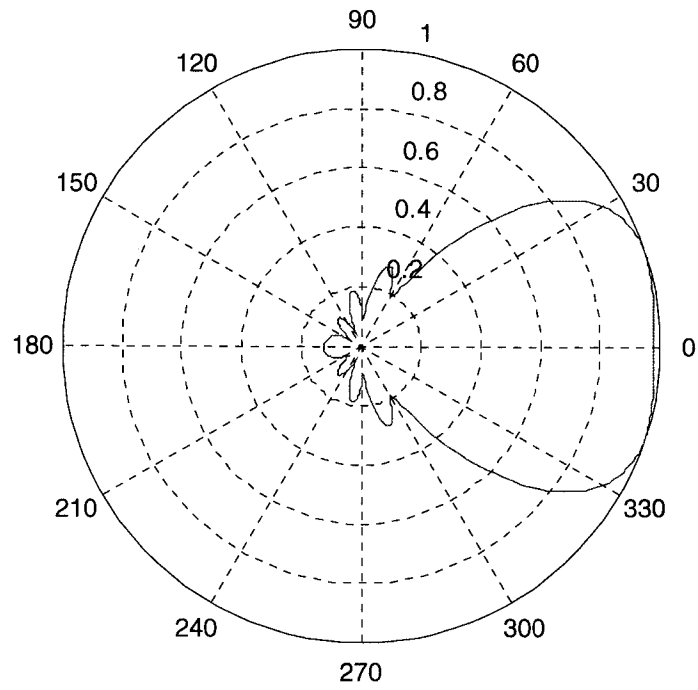


Figure 4.12 Polar graph of the scanning beam (phase delay=4.7124)

Chapter 5

Beam synthesis engine

The microelectronic beam synthesis engine controls the five steerable beams of each individual sensor microarray based on the inputs of the five scanning beams. The beam synthesis engine determines which scanning beam captured the right information, and picks the respective mainbeam out of the five mainbeams of each acoustical microarray planar. And then the mainbeam is steered to the potential sound source location. The sound signal can be localized accordingly. Then the potential sound signal will go through a series of acoustical filter. The Acoustical filter is a type of filter used for processing sound signal. It's typically designed to pass some frequency regions [24].

5.1 Beam synthesis

5.1.1 Introduction

The beam synthesis engine compares the received amplitudes simultaneously by means of five scanning beams. The angular errors of the target direction, with respect to the "broadside" point, are measured by the received level differences. The differences are compared to the sums in order to obtain tracking signals which are independent of the overall level of the received signals.

Tracking is then independent of signal amplitude fluctuations. Then the beam synthesis engine will send the tracking signal to the appointed steering beam to steer the respective main beam toward that direction.

5.1.2 Simulation for beam synthesis engine in Matlab

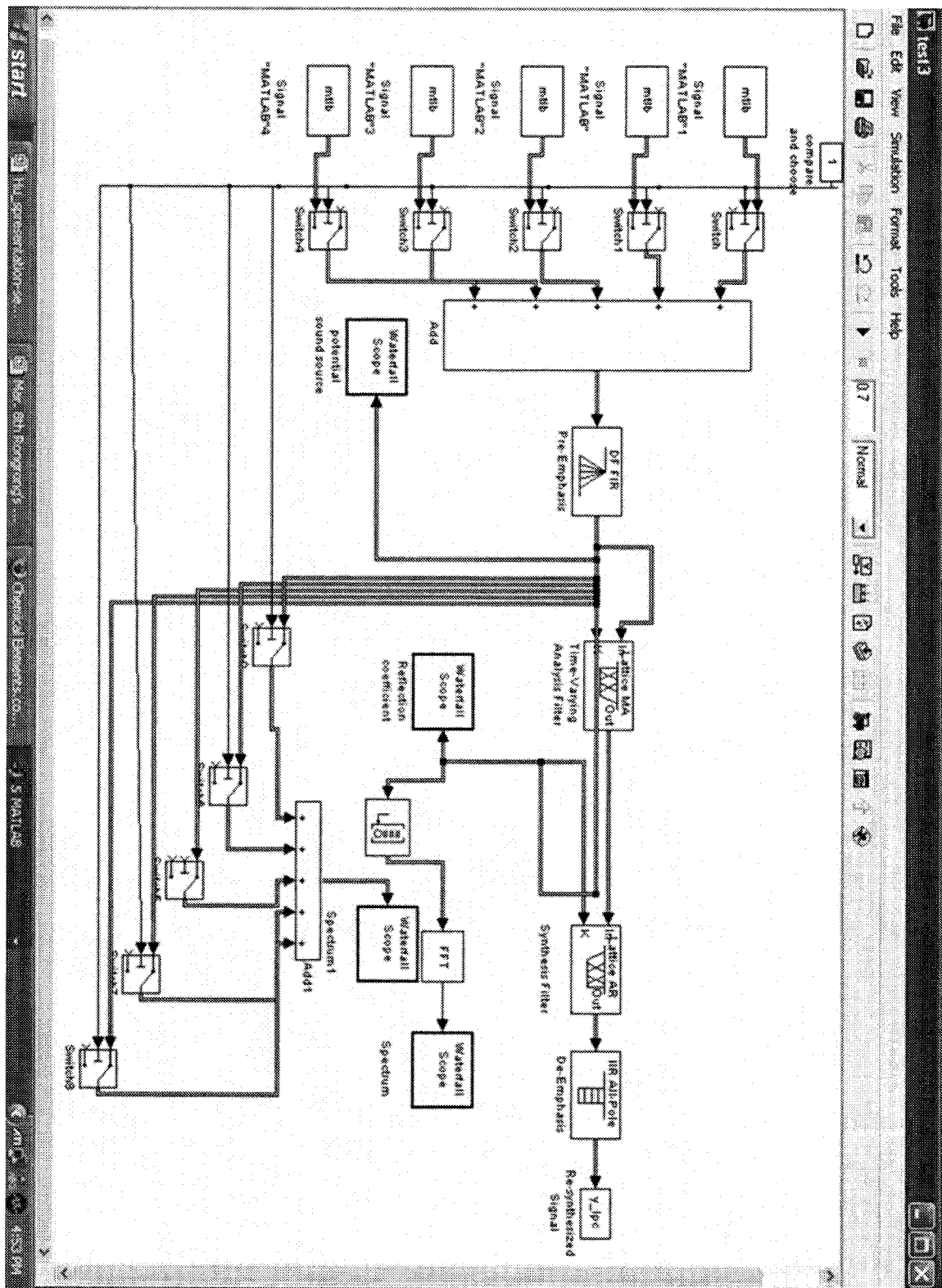


Figure 5.1 Simulation for beam synthesis engine in Matlab/Simulink

5.1.2.1 Simulation Specifications

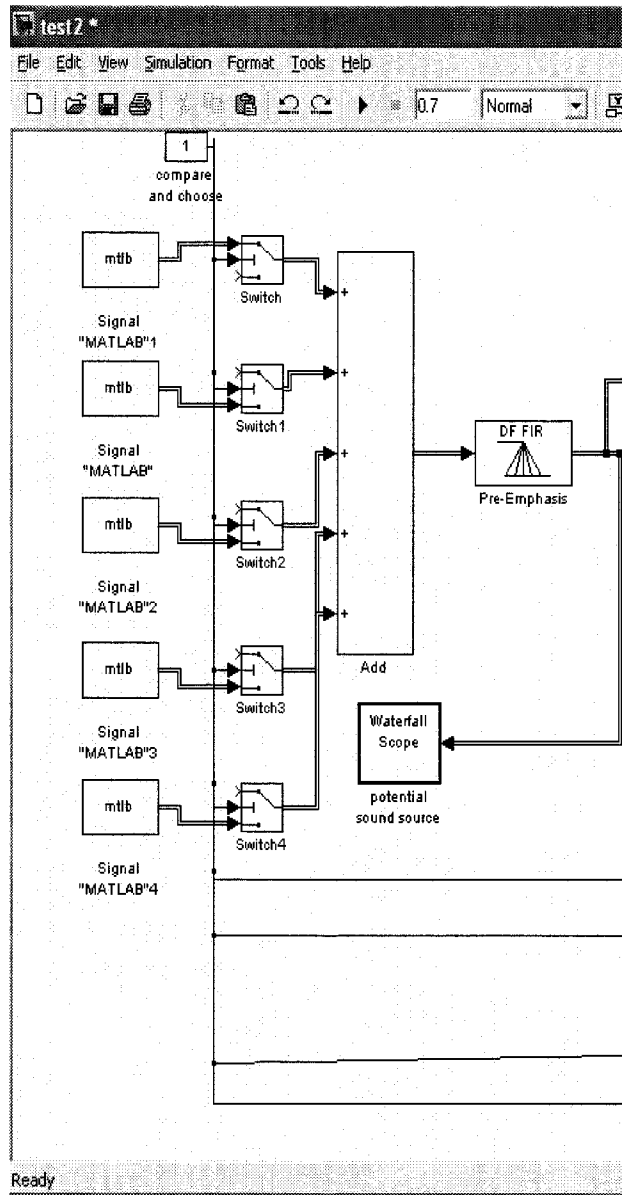


Figure 5.2 Beam scanning part

The five input signals simulate the 5 scanning beams located on each MEMS acoustical sensor microarray, which consist of 3X3 capacitive type acoustical sensor.

“Compare and choose” unit will be used to identify a potential speech source. The signal samples obtained at successive sample times.

In consideration of the fact that people's hearing range is from 20 Hz to 20,000 Hz (20 kHz), we chose 80 samples per frame, and the sample time is 1/8000 s.

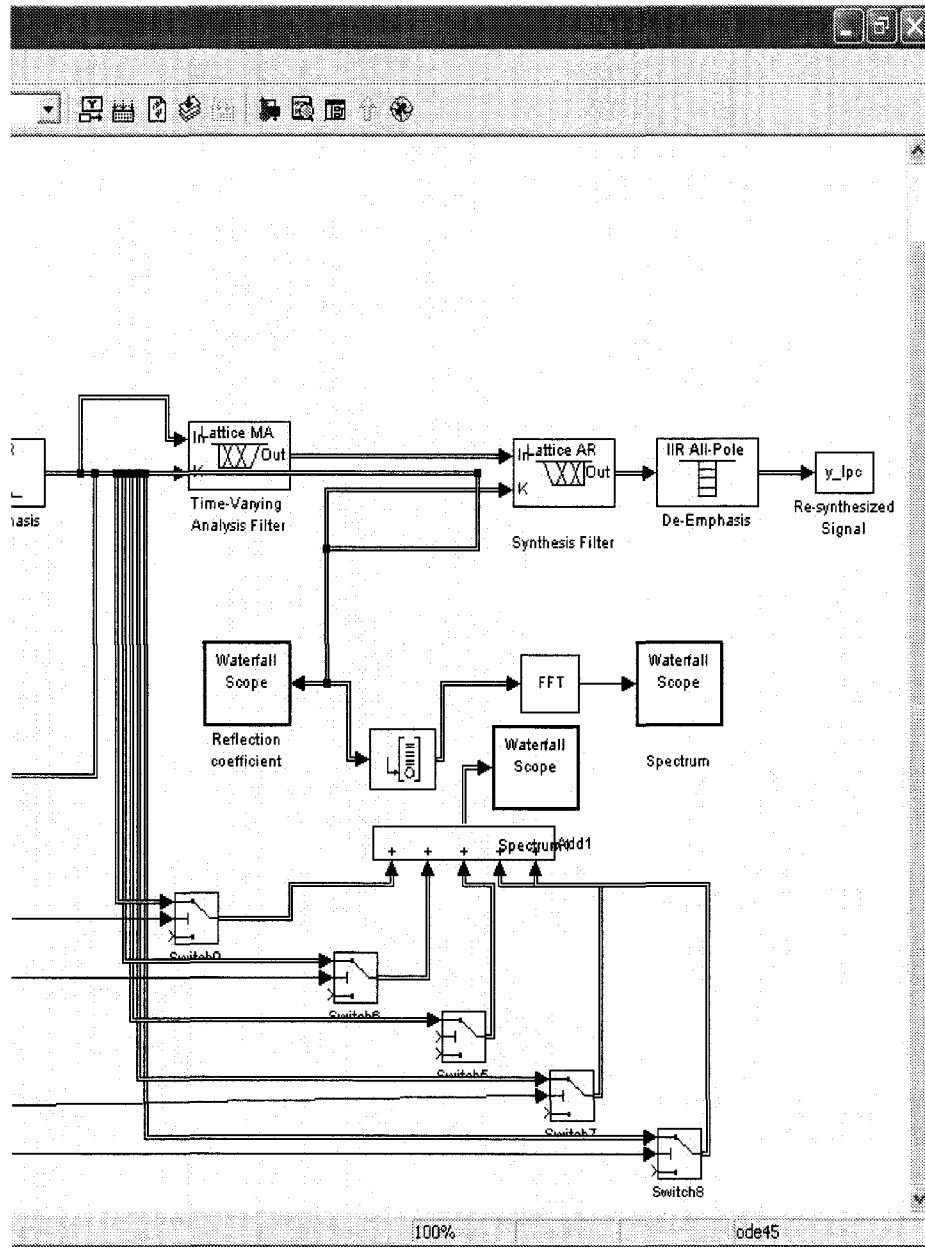


Figure 5.3 Beam synthesis part

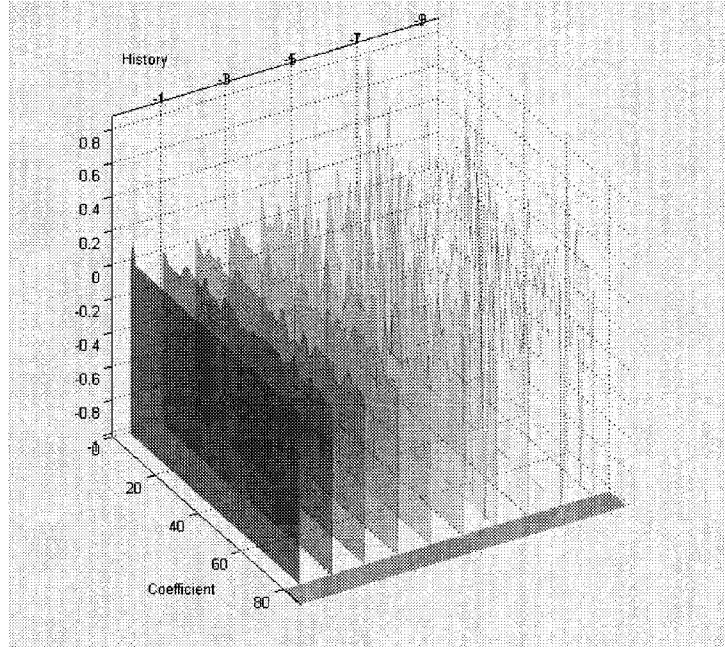
Pre-Emphasis unit is a digital filter, which independently filter each channel of the input over time using a specified digital filter implementation. Timing-varying

coefficients can update at a rate of one filter per frame. They can stay constant for the duration of an input frame and change for the next frame.

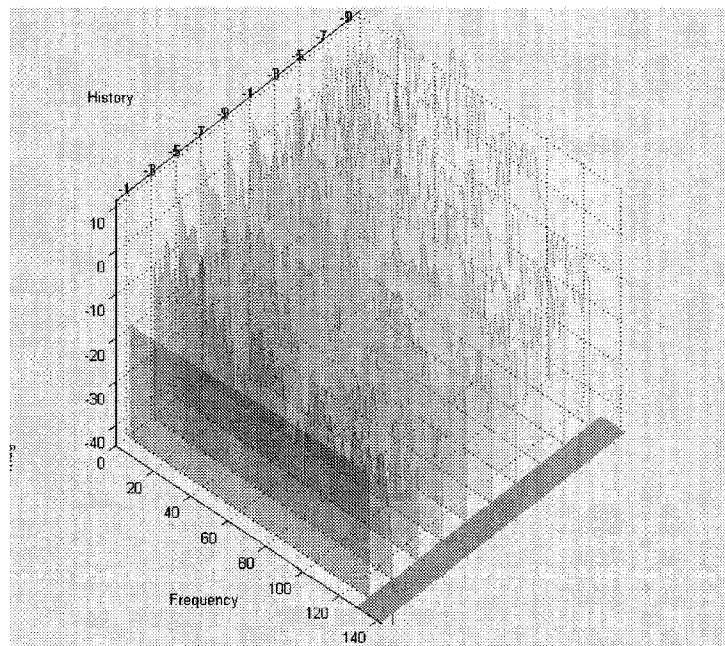
Potential sound source waterfall scope will show that the specific mainbeam is steered to that direction using beamsteering technology.

Spectrum waterfall scope will show the output of the synthesis engine. The output signal then can be amplified to a proper range that fits human ear.

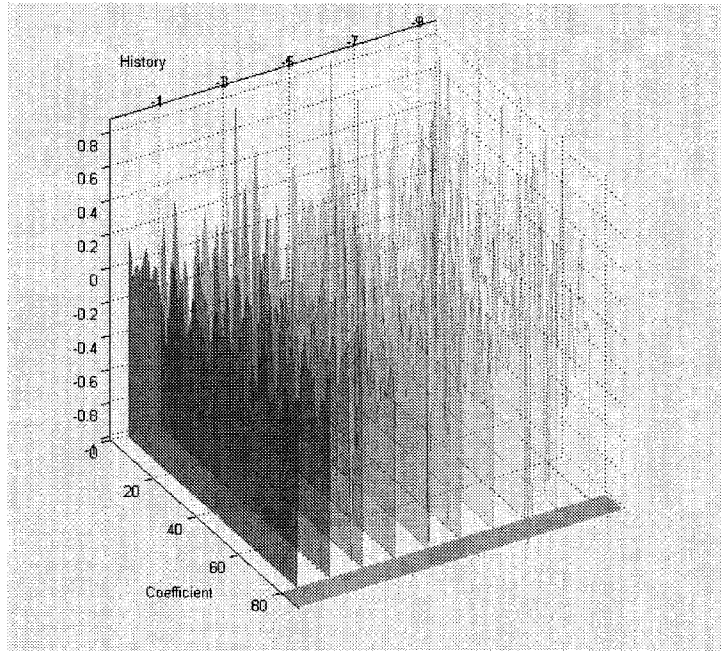
5.1.3 Simulation results



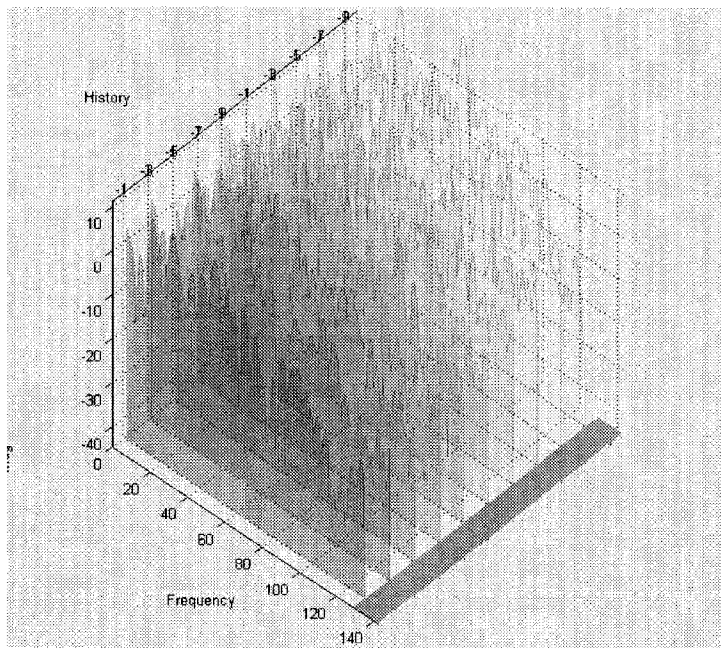
5.4a Potential Sound Source Waterfall Scope (Input)



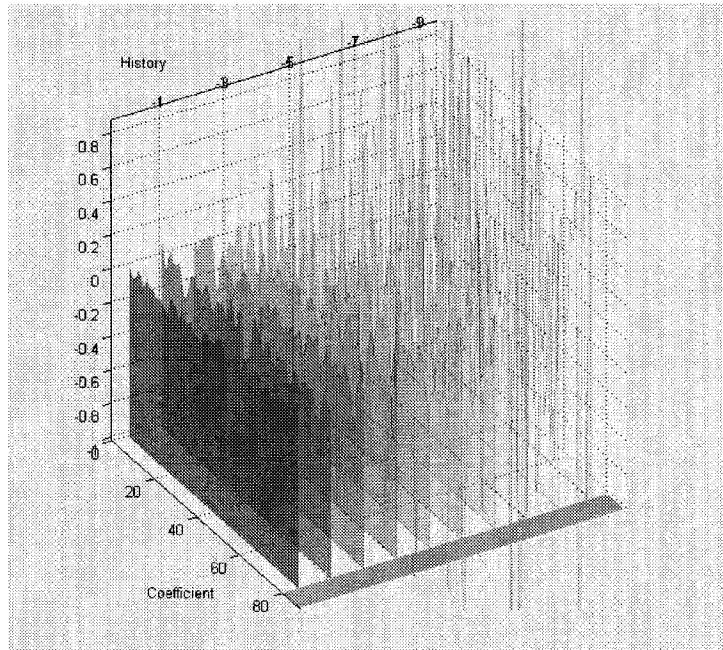
5.4b Spectrum Waterfall Scope (Output)



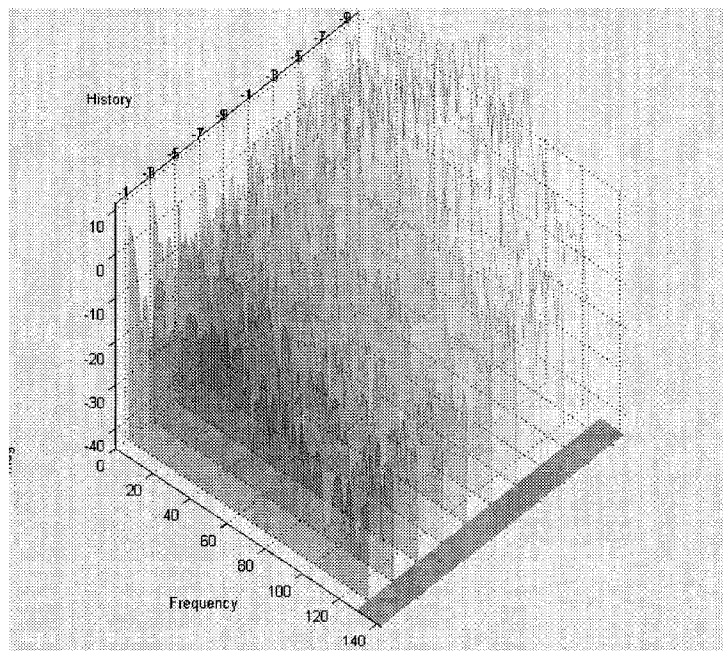
5.4c Potential Sound Source Waterfall Scope (Input)



5.4d Spectrum Waterfall Scope (Output)



5.4e Potential Sound Source Waterfall Scope (Input)



5.4f Spectrum Waterfall Scope (Output)

Chapter 6

Conclusion

6.1 Conclusion

The design of a cubic geometry acoustical sensor microarray cluster that can provide a 3-D dynamically variable directional acoustical sensitivity when used in conjunction with appropriate beamforming, beam steering and beam synthesis engines is presented. A beam synthesis algorithm has been developed to process the real time data from the five scanning beams associated with five planar arrays of acoustical sensors constituting the sides of a cubic geometry. The algorithm then steers one of the five main beams associated with five planar arrays in the direction of highest signal intensity level detected by the scanning beams to acquire speech signal from only that direction. A solder-ball based assembling method has been determined to be a more pragmatic one to assemble the 3-D cubic geometry from five planar arrays of acoustical sensors considering the fabrication complexity associated with microhinge structures. The bottom plate of the cubic microarrays cluster is to be used for interconnection and packaging purposes. Through FEA analysis carried out in ANSYS, it has been determined that 3 solder balls of 0.2 mm diameter are sufficient for hold two adjacent sides of the cubic microarray cluster at 90 degrees.

A readily steerable beam pattern is formed in real time from a receive aperture containing a sparsely populated array of acoustical elements with selected spacings. A summation of the synthetic spatial frequencies produces the

scannable beam with freedom from interference as that obtainable from the same size aperture fully filled with closely spaced elements. By applying the synthesized spatial frequencies to a fully filled transmit array of elements, the five scanning beams can work simultaneously and cooperate with each other. The simulation result demonstrates the effectiveness of the beam synthesis filters applied to the output of the array. Matlab/Simulink simulation of the beam synthesis engine provides excellent results.

The developed system can replace the sound acquisition systems used in conventional hearing aid instruments to provide a better speech intelligibility to hearing challenged persons to improve the quality of their lives.

The future works will be designing the electronic amplification algorithms: It can be achieved by a Thevenin equivalent circuit with a generator in series with internal impedance.

References

- [1] Julie E. Greenberg, Patrick M. Zurek, "Evaluation of an adaptive beamforming method for hearing aids", Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.
- [2] Xiaoming Wu; Tianling Ren; Litian Liu, "Sound source localization based on directivity of MEMS microphones", Solid-State and Integrated Circuits Technology, 2004. Proceedings. 7th International Conference on Volume 3, 18-21 Oct. 2004, pp.1884-1887, vol.3.
- [3] Po-Jen Zheng, J. Z. Lee, K. H. Liu, J. D. Wu, S. C. Hung, " Solder joint reliability of TFBGA assemblies with fresh and reworked solder balls", Microelectronics Reliability, Volume 43, 2003, pp. 925-934.
- [4] Malioutov, D.M.; Cetin, M.; Fisher, J.W.I.I.I.; Willsky, A.S., "Super resolution source localization through data-adaptive", 2002 4-6 Aug. 2002 pp.194-198.
- [5] M. W. Hoffman and R. W. Steward, "Simulation of multi-microphone hearing aids in multiple interface environments", British journal of audiology, Volume 30, No. 4, pp. 249-260, Aug. 1996.
- [6] Sazzadur Chowdhury, M. Ahmadi, W. C. Miller, "Microelectromechanical systems and system-on-chip connectivity", Circuits and systems magazine, IEEE, Volume 2, Issue 2, 2002, pp.4-28.
- [7] A. Mason, N. Yazdi, K. Najafi, and K.D. Wise, "A low power wireless micro-instrumentation system for environmental monitoring," in Proc. Transducers 95, Stockholm, June 1995, pp 107-110.
- [8] Stephen D. Senturia, "Microsystem design", Kluwer Academic Publishers, 2003, pp. 497-550.

[9] Chowdhury, S.; Ahmadi, M.; Miller, W. C., "The concept of a 3-D cubic acoustical sensor microarray cluster for use in a hearing instrument", in proceedings of 2004 International Conference on MEMS, NANO and Smart Systems, 2004, 25-27 Aug. 2004, pp. 138-142.

[10] Chowdhury, S.; Ahmadi, M.; Miller, W. C., "The concept of a 3-D cubic acoustical sensor microarray cluster for use in a hearing instrument", in proceedings of 2004 International Conference on MEMS, NANO and Smart Systems, 2004, 25-27 Aug. 2004, pp. 138-142.

[11] Yong Rui, Dinei Florêncio, Warren Lam, and Jinyan Su, "Sound source localization for circular arrays of directional microphone", Volume 3, 18-23 March 2005 pp.93-96.

[12] Warren L. Stutzman, Gary A. Thiele, "Antenna theory and design, second edition", John Wiley & Sons INC., 1998, pp. 365-425

[13] Sazzadur Chowdhury, M. Ahmadi, W. C. Miller, "Microelectromechanical systems and system-on-chip connectivity", Circuits and systems magazine, IEEE, Volume 2, Issue 2, 2002, pp.4-28

[14] D. Varon, G.I.Zysman, "Some properties and limitations of electronically steerable phased array antennas", American and telegraph company, September 1967, Volume XLVI, No. 7, pp 163-185

[15] Boris Sheleg, "A Matrix-Fed Circular Array for continuous scanning", Proceedings of the IEEE, November 1968, Volume 56, pp 2016-2027

[16] Jean Claude Sureau, Alexander Hessel, "Element pattern for circular arrays of waveguide-fed axial slits on large conducting cylinders", IEEE transactions on antennas and propagation, January 1971, Volume ap-19

- [17] Major A. Johnson, "Phased-array beam steering by multiplex sampling", Proceedings of IEEE, November 1968, Volume 56, No.11, pp. 1801-1811
- [18] R. S. Elliott, "Beamwidth and directivity of large scanning arrays", Proceedings of IEEE, 1960, Volume 60, pp. 3-19
- [19] Chang-Wook Baek, Seunghyun Song, Jae-Hyoung Park, "A V-Band micromachined 2-D beam-steering antenna driven by magnetic force with polymer-based hinges", IEEE transactions on microwave theory and techniques, January 2003, Volume 51, No.1, pp 325-331
- [20] Rene J. Allard, Douglas H. Werner, Pingjuan L. Werner, "Radiation pattern synthesis for arrays of conformal antenna mounted on arbitrarily-shaped three-dimensional platforms using genetic algorithms", IEEE transactions on antennas and propagation, May 2003, Volume 51, No.5, pp. 1054-1062
- [21] Manuel Vicente-Lozano, Giorgio Franceschetti, Francisco J. Ares-Pena, Eduardo Moreno-Piquero, "Analysis and synthesis of a printed array for satellite communication with moving vehicles", IEEE transactions on antennas and propagation, November 2002, Volume 50, No.11, pp. 1555-1558
- [22] Barry J. Forman, "Directivity characteristics of scannable planar arrays", IEEE transactions on antennas and propagation, May 1972, Volume ap-20, No.3, pp. 245-252
- [23] Sharad R. Laxpati, "Planar array synthesis with prescribed pattern nulls", IEEE transactions on antennas and propagation, November 1982, Volume ap-30, No.6, pp. 1176-1183

- [24] Yaowu Mo, Tsunehisa Tanaka, Koji Inoue, Kaoru Yamashita, Yoshihiko Suzuki, "Front-end processor using BBD Distributed delay-sum architecture for Micromachined ultrasonic sensor array", *Journal of Microelectromechanical Systems*, August 2003, Volume 12, No. 4, pp. 506-512
- [25] Jack Chen, Chang Liu, "Development and characterization of surface micromachined, out-of-plane hot-wire anemometer", *Journal of Microelectromechanical Systems*, December 2003. Volume 12, No. 6, pp 979-988
- [26] Royer M, Holmen P, Wurm M, Aadland P and Glenn M 1983 "ZnO on Si integrated acoustic sensor" *Sens. Actuators A* **4**, pp.357–362
- [27] T. Tanabe, S. Harada, "Examination of compound formation at interface of tin-bismuth-silver solder and copper substrate by using electron probe micro analysis", Volume 31, *X-Ray Spectrometry*, 2002, pp. 3-6
- [28] Jerry E. Boyns, C. W. Gorham, A. David Munger, Joseph H. Provencher, John Reindel, "Step-scanned circular-array antenna", *IEEE transactions on antennas and propagation*, September 1970, Volume ap-18, No. 5, pp 223-232
- [29] D.G. Tucker, "Arrays with Constant Beam-Width over a Wide Frequency-Range," *Nature*, vol. 180, pp. 496-497, 1957.
- [30] J. C. Morris, "Broad-Band Constant Beam-Width Transducers," *Journal of Sound and Vibration*, vol. 1, pp. 28-40, 1964.
- [31] J. L. Flanagan, J. D. Johnston, A. Zahn, and G. W. Elko, "Computer-Steered Microphone Arrays for Sound Transduction in Large Rooms," *Journal of the Acoustical Society of America*, vol. 78, no. 5, pp. 1508-1518, 1985.

Appendix A

Program 1

```
% program name: Polar graph of the scanning beam.m
```

```
% by Rongrong Hu
```

```
% Date: May 24, 2005
```

```
N=[9 9 9]; % number of elements
```

```
d=[5 5 5]; % spacing between elements
```

```
p=[0 pi/6 pi/4 pi/3 pi/2 3*pi/4 pi 3*pi/2]; % phase delay
```

```
t=0:360;
```

```
t=t*2*pi/360;
```

```
z=S(1,0,0);
```

```
Z=sum(abs(z),2);
```

```
polar(t,Z'./max(Z'));
```

```
title('One element, no phase delay');
```

```
disp('printing file n1d0p0.polar.png');
```

```
eval(sprintf('print -dpng n1d0p0.polar.png'));
```

```
for i=1:3
```

```

for j=1:3
    for k=1:8
        z=S(N(i),d(j),p(k));
        Z=sum(abs(z),2);
        polar(t,Z'./max(Z'));
        title([num2str(N(i)) ' elements, spacing = ' num2str(d(j)) ...
            ', relative phase delay = ' num2str(p(k))]);
        filenm = ['n' num2str(N(i)) 'd' num2str(d(j)) 'p' num2str(p(k)) '.polar.png'];
        disp(['printing file ' filenm]);
        eval(sprintf('print -dpng "%s"',filenm));
        figure(2);
        z=S(N(i),d(j),p(k));
        Z=sum(abs(z),2);
        polar(t,Z'./max(Z'));
        title([num2str(N(i)) ' elements, spacing = ' num2str(d(j)) ...
            ', relative phase delay = ' num2str(p(k))]);
        filenm = ['n' num2str(N(i)) 'd' num2str(d(j)) 'p' num2str(p(k)) '.polar.png'];
        disp(['printing file ' filenm]);
        eval(sprintf('print -dpng "%s"',filenm));
        figure(3);
        z=S(N(i),d(j),p(k));
        Z=sum(abs(z),2);
        polar(t,Z'./max(Z'));
        title([num2str(N(i)) ' elements, spacing = ' num2str(d(j)) ...
            ', relative phase delay = ' num2str(p(k))]);
        filenm = ['n' num2str(N(i)) 'd' num2str(d(j)) 'p' num2str(p(k)) '.polar.png'];
        disp(['printing file ' filenm]);
        eval(sprintf('print -dpng "%s"',filenm));
        figure(4);
        z=S(N(i),d(j),p(k));
        Z=sum(abs(z),2);

```

```

polar(t,Z'./max(Z));
title([num2str(N(i)) ' elements, spacing = ' num2str(d(j)) ...
      ', relative phase delay = ' num2str(p(k))]);
filenm = ['n' num2str(N(i)) 'd' num2str(d(j)) 'p' num2str(p(k)) '.polar.png'];
disp(['printing file ' filenm]);
eval(sprintf('print -dpng "%s"',filenm));
figure(5);
z=S(N(i),d(j),p(k));
Z=sum(abs(z),2);
polar(t,Z'./max(Z));
title([num2str(N(i)) ' elements, spacing = ' num2str(d(j)) ...
      ', relative phase delay = ' num2str(p(k))]);
filenm = ['n' num2str(N(i)) 'd' num2str(d(j)) 'p' num2str(p(k)) '.polar.png'];
disp(['printing file ' filenm]);
eval(sprintf('print -dpng "%s"',filenm));

end;
end;
end;

```

Appendix B

Program 2

```
%Program name: 3D BeamPattern.m
% by: Rongrong Hu
%Date: June 20, 2005
% 3-D beam pattern:350Hz beam for r=4,theta0=0,phi0=pi/2
Phi=0:pi/10:2*pi;
Theta=-pi/2:pi/10:pi;
[PHI,THETA]=meshgrid(Phi,Theta);
r0=0.5;
A=0;
B=0;
for i=-4:1:4,
    for j=-4:1:4,
        AF=(1+j*(1.9e-4))*(sin(1.5*(2*(1.06e-
4)*sin(PHI).*cos(THETA)+j*0.39))./(1.5*(2*(1.06e-
4)*sin(PHI).*cos(THETA)+j*0.39)));
        A=A+AF;

        af=(1+i*(1.9e-4))*(sin(1.5*(2*(1.06e-
4)*sin(PHI).*sin(THETA)+i*0.39))./(1.5*(2*(1.06e-
4)*sin(PHI).*sin(THETA)+i*0.39)));
        B=B+af;
    end
end
```

```
end
R=B.*A;
[X,Y,Z]=sph2cart(THETA,PHI,R); %get cartesian value
mesh(X,Y,Z); %display
```

Appendix C

Program 3

%Source files for Beam Synthesis Engine simulation Test3.mdl

% by Rongrong Hu

% Date: Nov.12, 2005

Model {

 Name "Beam Synthesis Engine test3"

 GraphicalInterface {

 NumRootInports 0

 NumRootOutports 0

 ParameterArgumentNames ""

 ComputedModelVersion "1.188"

 NumModelReferences 0

 NumTestPointedSignals 0

 }

 RequirementInfo "9"

 SavedCharacterEncoding "US-ASCII"

 PreLoadFcn "load mtlb;"

 SaveDefaultBlockParams on

 SampleTimeColors off

 LibraryLinkDisplay "none"

 WideLines on

 ShowLineDimensions on

ShowPortDataTypes off
ShowLoopsOnError on
IgnoreBidirectionalLines off
ShowStorageClass off
ShowTestPointIcons on
ShowViewerIcons on
SortedOrder off
ExecutionContextIcon off
ShowLinearizationAnnotations on
RecordCoverage off
CovPath "/"
CovSaveName "covdata"
CovMetricSettings "d"
CovNameIncrementing off
CovHtmlReporting on
covSaveCumulativeToWorkspaceVar on
CovSaveSingleToWorkspaceVar on
CovCumulativeVarName "covCumulativeData"
CovCumulativeReport off
CovReportOnPause on
ScopeRefreshTime 0.035000
OverrideScopeRefreshTime on
DisableAllScopes off
DataTypeOverride "UseLocalSettings"
MinMaxOverflowLogging "UseLocalSettings"
MinMaxOverflowArchiveMode "Overwrite"
BlockNameDataTip off
BlockParametersDataTip on
BlockDescriptionStringDataTip off
ToolBar on
StatusBar on

BrowserShowLibraryLinks off
BrowserLookUnderMasks off
Created "Mon Jul 20 16:50:05 1998"
Creator "The MathWorks Inc."
UpdateHistory "UpdateHistoryNever"
ModifiedByFormat "%<Auto>"
LastModifiedBy "batserve"
ModifiedDateFormat "%<Auto>"
LastModifiedDate "Sat Aug 7 06:50:17 2004"
ModelVersionFormat "1.%<AutoIncrement:188>"
ConfigurationManager "none"
LinearizationMsg "none"
Profile off
ParamWorkspaceSource "MATLABWorkspace"
AccelSystemTargetFile "accel.tlc"
AccelTemplateMakefile "accel_default_tmf"
AccelMakeCommand "make_rtw"
TryForcingSFcnDF off
ExtModeBatchMode off
ExtModeEnableFloating on
ExtModeTrigType "manual"
ExtModeTrigMode "oneshot"
ExtModeTrigPort "1"
ExtModeTrigElement "any"
ExtModeTrigDuration 1000
ExtModeTrigDurationFloating "auto"
ExtModeTrigHoldOff 0
ExtModeTrigDelay 0
ExtModeTrigDirection "rising"
ExtModeTrigLevel 0
ExtModeArchiveMode "off"

```

ExtModeAutoIncOneShot off
ExtModeIncDirWhenArm off
ExtModeAddSuffixToVar off
ExtModeWriteAllDataToWs off
ExtModeArmWhenConnect off
ExtModeSkipDownloadWhenConnect off
ExtModeLogAll on
ExtModeAutoUpdateStatusClock off
BufferReuse on
StrictBusMsg "None"
ProdHWDeviceType "Specified"
ShowModelReferenceBlockVersion off
ShowModelReferenceBlockIO off
Array {
  Type "Handle"
  Dimension 1
  Simulink.ConfigSet {
    $ObjectID 1
    Version "1.0.4"
    Array {
      Type "Handle"
      Dimension 7
      Simulink.SolverCC {
        $ObjectID 2
        Version "1.0.4"
        StartTime "0.0"
        StopTime "0.49"
        AbsTol "auto"
        FixedStep "auto"
        InitialStep "auto"
        MaxNumMinSteps "-1"
      }
    }
  }
}

```

```

MaxOrder      5
ExtrapolationOrder  4
NumberNewtonIterations  1
MaxStep       "auto"
MinStep       "auto"
RelTol        "1e-3"
SolverMode    "SingleTasking"
Solver        "FixedStepDiscrete"
SolverName    "FixedStepDiscrete"
ZeroCrossControl "UseLocalSettings"
PositivePriorityOrder off
AutoInsertRateTranBlk off
SampleTimeConstraint "Unconstrained"
RateTranMode   "Deterministic"
}
Simulink.DataIOCC {
  $ObjectID    3
  Version      "1.0.4"
  Decimation   "1"
  ExternalInput "[t, u]"
  FinalStateName "xFinal"
  InitialState  "xInitial"
  LimitDataPoints off
  MaxDataPoints "1000"
  LoadExternalInput off
  LoadInitialState off
  SaveFinalState off
  SaveFormat    "Structure"
  SaveOutput    off
  SaveState     off
  SignalLogging on
}

```

```

SaveTime      off
StateSaveName "xout"
TimeSaveName  "tout"
OutputSaveName "yout"
SignalLoggingName "sigsOut"
OutputOption  "RefineOutputTimes"
OutputTimes   "[]"
Refine        "1"
}
Simulink.OptimizationCC {
  $ObjectID    4
  Version      "1.0.4"
  BlockReduction off
  BooleanDataType off
  ConditionallyExecuteInputs on
  ConditionalExecOptimization "on_for_testing"
  InlineParams off
  InlineInvariantSignals off
  OptimizeBlockIOStorage on
  BufferReuse on
  EnforceIntegerDowncast on
  ExpressionFolding on
  FoldNonRolledExpr on
  LocalBlockOutputs on
  ParameterPooling on
  RollThreshold 5
  SystemCodeInlineAuto off
  StateBitsets off
  DataBitsets off
  UseTempVars off
  ZeroExternalMemoryAtStartup on

```

```

ZeroInternalMemoryAtStartup on
InitFltsAndDblsToZero on
NoFixptDivByZeroProtection off
OptimizeModelRefInitCode off
LifeSpan    "inf"
}
Simulink.DebuggingCC {
  $ObjectID    5
  Version      "1.0.4"
  ConsistencyChecking "none"
  ArrayBoundsChecking "none"
  AlgebraicLoopMsg "warning"
  ArtificialAlgebraicLoopMsg "warning"
  CheckSSInitialOutputMsg on
  CheckExecutionContextPreStartOutputMsg off
  CheckExecutionContextRuntimeOutputMsg off
  SignalResolutionControl "TryResolveAllWithWarning"
  BlockPriorityViolationMsg "warning"
  MinStepSizeMsg "warning"
  SolverPrmCheckMsg "none"
  InheritedTInSrcMsg "warning"
  DiscreteInheritContinuousMsg "warning"
  MultiTaskDSMMsg "warning"
  MultiTaskRateTransMsg "error"
  SingleTaskRateTransMsg "none"
  TasksWithSamePriorityMsg "warning"
  CheckMatrixSingularityMsg "none"
  IntegerOverflowMsg "none"
  Int32ToFloatConvMsg "warning"
  ParameterDowncastMsg "error"
  ParameterOverflowMsg "error"

```

```

ParameterPrecisionLossMsg "warning"
UnderSpecifiedDataTypeMsg "none"
UnnecessaryDatatypeConvMsg "none"
VectorMatrixConversionMsg "none"
InvalidFcnCallConnMsg "error"
FcnCallInpInsideContextMsg "Use local settings"
SignalLabelMismatchMsg "none"
UnconnectedInputMsg "warning"
UnconnectedOutputMsg "warning"
UnconnectedLineMsg "warning"
SFcnCompatibilityMsg "none"
UniqueDataStoreMsg "none"
RootOutportRequireBusObject "warning"
AssertControl "UseLocalSettings"
EnableOverflowDetection off
ModelReferenceIOMsg "none"
ModelReferenceVersionMismatchMessage "none"
ModelReferenceIOMismatchMessage "none"
ModelReferenceCSMismatchMessage "none"
ModelReferenceSimTargetVerbose off
UnknownTsInhSupMsg "warning"
ModelReferenceDataLoggingMessage "warning"
ModelReferenceSymbolNameMessage "warning"
}
Simulink.HardwareCC {
  $ObjectID 6
  Version "1.0.4"
  ProdBitPerChar 8
  ProdBitPerShort 16
  ProdBitPerInt 32
  ProdBitPerLong 32
}

```

```

ProdIntDivRoundTo "Undefined"
ProdEndianness    "Unspecified"
ProdWordSize      32
ProdShiftRightIntArith on
ProdHWDeviceType  "Specified"
TargetBitPerChar  8
TargetBitPerShort 16
TargetBitPerInt   32
TargetBitPerLong  32
TargetShiftRightIntArith on
TargetIntDivRoundTo "Undefined"
TargetEndianness   "Unspecified"
TargetWordSize     32
TargetTypeEmulationWarnSuppressLevel 0
TargetPreprocMaxBitsSint 32
TargetPreprocMaxBitsUint 32
TargetHWDeviceType "Specified"
TargetUnknown      off
ProdEqTarget       on
}
Simulink.ModelReferenceCC {
  $ObjectID      7
  Version        "1.0.4"
  UpdateModelReferenceTargets "IfOutOfDateOrStructuralChange"
  CheckModelReferenceTargetMessage "error"
  ModelReferenceNumInstancesAllowed "Multi"
  ModelReferencePassRootInputsByReference on
  ModelReferenceMinAlgLoopOccurrences off
}
Simulink.RTWCC {
  $BackupClass    "Simulink.RTWCC"

```

```

$ObjectID      8
Version        "1.0.4"
SystemTargetFile "grt.tlc"
GenCodeOnly    off
MakeCommand    "make_rtw"
TemplateMakefile "grt_default_tmf"
GenerateReport off
SaveLog        off
RTWVerbose     on
RetainRTWFile  off
ProfileTLC     off
TLCDebug       off
TLCCoverage    off
TLCAssert     off
ProcessScriptMode "Default"
ConfigurationMode "Optimized"
ConfigAtBuild   off
IncludeHyperlinkInReport off
LaunchReport    off
Array {
  Type          "Handle"
  Dimension      2
  Simulink.CodeAppCC {
    $ObjectID    9
    Version      "1.0.4"
    ForceParamTrailComments off
    GenerateComments on
    IgnoreCustomStorageClasses on
    IncHierarchyInIds off
    MaxIdLength  31
    PreserveName off
  }
}

```



```

PreserveNameWithParent off
ShowEliminatedStatement on
IncAutoGenComments off
SimulinkDataObjDesc off
SFDataObjDesc off
IncDataTypeInIds off
PrefixModelToSubsysFcnNames on
CustomSymbolStr "$R$N$M"
MangleLength 1
DefineNamingRule "None"
ParamNamingRule "None"
SignalNamingRule "None"
InsertBlockDesc off
SimulinkBlockComments on
EnableCustomComments off
InlinedPrmAccess "Literals"
ReqsInCode off
}
Simulink.GRTTargetCC {
$BackupClass "Simulink.TargetCC"
$ObjectID 10
Version "1.0.4"
TargetFcnLib "ansi_tfl_tmw.mat"
GenFloatMathFcnCalls "ANSI_C"
UtilityFuncGeneration "Auto"
GenerateFullHeader on
GenerateSampleERTMain off
IsPILTarget off
ModelReferenceCompliant on
IncludeMdITerminateFcn on
CombineOutputUpdateFcns off

```

```

SuppressErrorStatus off
IncludeFileDelimiter "Auto"
ERTCustomFileBanners off
SupportAbsoluteTime on
LogVarNameModifier "rt_"
MatFileLogging on
MultiInstanceERTCode off
SupportNonFinite on
SupportComplex on
PurelyIntegerCode off
SupportContinuousTime on
SupportNonInlinedSFcns on
ExtMode off
ExtModeStaticAlloc off
ExtModeTesting off
ExtModeStaticAllocSize 1000000
ExtModeTransport 0
ExtModeMexFile "ext_comm"
RTWCAPISignals off
RTWCAPIParams off
RTWCAPISStates off
GenerateASAP2 off
}
PropName "Components"
}
}
PropName "Components"
}
Name "Configuration"
SimulationMode "normal"
CurrentDlgPage "Solver"

```

```

}
  PropName      "ConfigurationSets"
}
Simulink.ConfigSet {
  $PropName     "ActiveConfigurationSet"
  $ObjectID     1
}
BlockDefaults {
  Orientation    "right"
  ForegroundColor "black"
  BackgroundColor "white"
  DropShadow     off
  NamePlacement  "normal"
  FontName       "Helvetica"
  FontSize       10
  FontWeight     "normal"
  FontAngle      "normal"
  ShowName       on
}
BlockParameterDefaults {
  Block {
    BlockType     ComplexToReallmag
    Output        "Real and imag"
    SampleTime    "-1"
  }
  Block {
    BlockType     DataTypeConversion
    OutDataTypeMode "Inherit via back propagation"
    OutDataType    "sfix(16)"
    OutScaling     "2^0"
    LockScale      off
  }
}

```

```

ConvertRealWorld    "Real World Value (RWV)"
RndMeth             "Zero"
SaturateOnIntegerOverflow on
SampleTime          "-1"
}
Block {
  BlockType          FrameConversion
  OutFrame           "Frame based"
}
Block {
  BlockType          FromWorkspace
  VariableName       "simulink_input"
  SampleTime         "-1"
  Interpolate        on
  OutputAfterFinalValue "Extrapolation"
}
Block {
  BlockType          Inport
  BusObject          "BusObject"
  BusOutputAsStruct  off
  PortDimensions     "-1"
  SampleTime         "-1"
  DataType           "auto"
  OutDataType        "sfix(16)"
  OutScaling         "2^0"
  SignalType         "auto"
  SamplingMode       "auto"
  Interpolate        on
}
Block {
  BlockType          Outputport

```

```

Port      "1"
BusObject "BusObject"
BusOutputAsStruct off
PortDimensions "-1"
SampleTime "-1"
DataType   "auto"
OutDataType "sfix(16)"
OutScaling "2^0"
SignalType "auto"
SamplingMode "auto"
OutputWhenDisabled "held"
InitialOutput "[]"
}
Block {
  BlockType Reference
}
Block {
  BlockType "S-Function"
  FunctionName "system"
  SFunctionModules ""
  PortCounts "[]"
}
Block {
  BlockType SubSystem
  ShowPortLabels on
  Permissions "ReadWrite"
  PermitHierarchicalResolution "All"
  SystemSampleTime "-1"
  RTWFcnNameOpts "Auto"
  RTWFileNameOpts "Auto"
  SimViewingDevice off
}

```

```

    DataTypeOverride    "UseLocalSettings"
    MinMaxOverflowLogging "UseLocalSettings"
}
Block {
    BlockType        ToWorkspace
    VariableName     "simulink_output"
    MaxDataPoints    "1000"
    Decimation       "1"
    SampleTime       "0"
    FixptAsFi        off
}
}
AnnotationDefaults {
    HorizontalAlignment "center"
    VerticalAlignment  "middle"
    ForegroundColor    "black"
    BackgroundColor    "white"
    DropShadow         off
    FontName           "Helvetica"
    FontSize           10
    FontWeight         "normal"
    FontAngle          "normal"
}
LineDefaults {
    FontName           "Helvetica"
    FontSize           9
    FontWeight         "normal"
    FontAngle          "normal"
}
System {
    Name               "dsplpc"

```

```

Location      [24, 117, 977, 522]
Open         on
ModelBrowserVisibility off
ModelBrowserWidth 200
ScreenColor   "white"
PaperOrientation "landscape"
PaperPositionMode "auto"
PaperType     "usletter"
PaperUnits    "inches"
ZoomFactor    "100"
ReportName    "simulink-default.rpt"
Block {
  BlockType    SubSystem
  Name         " "
  Ports        []
  Position      [148, 307, 221, 365]
  BackgroundColor "cyan"
  DropShadow   on
  FontName     "Arial"
  FontSize     14
  FontWeight   "bold"
  TreatAsAtomicUnit off
  MinAlgLoopOccurrences off
  RTWSystemCode "Auto"
  MaskType     "Frame-Based Simulation Info"
  MaskDisplay  "disp('Info')"
  MaskIconFrame on
  MaskIconOpaque on
  MaskIconRotate "none"
  MaskIconUnits "autoscale"
  System {

```

```

Name      " "
Location  [71, 324, 723, 641]
Open      off
ModelBrowserVisibility off
ModelBrowserWidth 200
ScreenColor "white"
PaperOrientation "landscape"
PaperPositionMode "auto"
PaperType "usletter"
PaperUnits "inches"
ZoomFactor "100"
  Name      "8 kHz\nSignal\n\"MATLAB\""
  Ports     [0, 1]
  Position  [15, 143, 70, 177]
  SourceBlock "dpsrcs4/Signal From\n\nWorkspace"
  SourceType "Signal From Workspace"
  ShowPortLabels on
  X         "mtlb*0.32"
  Ts        "1/8000"
  nsamps    "80"
  OutputAfterFinalValue "Setting to zero"
}
Block {
  BlockType Reference
  Name      "Autocorrelation"
  Ports     [1, 1]
  Position  [345, 135, 395, 185]
  DialogController "dspDDGCreate"
  DialogControllerArgs "DataTag0"
  SourceBlock "dspstat3/Autocorrelation"
  SourceType "Autocorrelation"

```



```

AllPositiveLags    off
maxlag             "12"
bias               "Biased"
domain             "Time"
additionalParams   off
allowOverrides     on
outputMode         "Same as input"
outputWordLength   "16"
outputFracLength   "15"
accumMode          "Same as product output"
accumWordLength    "40"
accumFracLength    "30"
prodOutputMode     "Same as input"
prodOutputWordLength "32"
prodOutputFracLength "30"
roundingMode       "Floor"
overflowMode       off
LockScale          off
}
Block {
  BlockType      Reference
  Name           "De-Emphasis"
  Ports          [1, 1]
  Position       [765, 140, 835, 180]
  DialogController "dspDDGCreate"
  DialogControllerArgs "DataTag1"
  SourceBlock     "dsparch4/Digital Filter"
  SourceType      "Digital Filter"
  TypePopup       "IIR (all poles)"
  IIRFiltStruct   "Direct form II transposed"
  AllPoleFiltStruct "Direct form"
}

```

FIRFiltStruct "Direct form"
 CoeffSource "Specify via dialog"
 NumCoeffs "[1 -.95]"
 DenCoeffs "[1 -.95]"
 BiQuadCoeffs "[1 0.3 0.4 1 0.1 0.2]"
 LatticeCoeffs "[0.2 0.4]"
 denIgnore on
 FiltPerSampPopup "One filter per frame"
 IC "0"
 INum "0"
 ICden "0"
 additionalParams off
 allowOverrides on
 showCoeff off
 firstCoeffMode "Same as input"
 firstCoeffWordLength "16"
 firstCoeffFracLength "15"
 secondCoeffMode "Same as numerator"
 secondCoeffWordLength "16"
 secondCoeffFracLength "15"
 thirdCoeffMode "Same as input"
 thirdCoeffWordLength "16"
 thirdCoeffFracLength "15"
 showOut off
 outputMode "Same as input"
 outputWordLength "16"
 outputFracLength "15"
 showAcc off
 accumMode "Same as product output"
 accumWordLength "40"
 accumFracLength "30"

```

showMpy      off
prodOutputMode    "Same as input"
prodOutputWordLength  "32"
prodOutputFracLength  "30"
showMem      off
memoryMode     "Same as input"
memoryWordLength  "16"
memoryFracLength  "14"
roundingMode   "Floor"
overflowMode   off
ScaleValues    "1"
scaleValueFracLength  "14"
tapSumMode     "Same as input"
tapSumWordLength  "32"
tapSumFracLength  "30"
stageIOMode    "Same as input"
stageIOWordLength  "16"
stageInFracLength  "15"
stageOutFracLength  "15"
LockScale     off
}
Block {
  BlockType     Reference
  Name          "FFT1"
  Ports        [1, 1]
  Position      [755, 268, 795, 302]
  ShowName     off
  DialogController  "dspDDGCreate"
  DialogControllerArgs  "DataTag2"
  SourceBlock   "dspxfm3/FFT"
  SourceType    "FFT"

```

```

CompMethod      "Table lookup"
TableOpt        "Speed"
BitRevOrder     off
additionalParams off
SkipNorm        off
allowOverrides  on
firstCoeffMode  "Same as input"
firstCoeffWordLength "16"
firstCoeffFracLength "15"
outputMode      "Same as input"
outputWordLength "16"
outputFracLength "12"
accumMode       "Same as product output"
accumWordLength "32"
accumFracLength "24"
prodOutputMode  "Same as input"
prodOutputWordLength "32"
prodOutputFracLength "24"
roundingMode    "Floor"
overflowMode    off
LockScale       off
}
Block {
  BlockType      Reference
  Name           "LPC\nSpectrum"
  Ports          [1]
  Position       [835, 260, 895, 310]
  SourceBlock    "dspsnks4/Waterfall"
  SourceType     "Waterfall"
  ShowPortLabels off
  TraceProperties off
}

```

```

NumTraces      "10"
CMapStr        "autumn"
TNewest        "0.35"
TOldest        "0.1"
HistoryLength  "100"
HistoryFull    "Overwrite"
UpdateInterval "1"
ExportMode     "All visible"
MLExportName   "ExportData"
AutoExport     off
DisplayProperties off
MouseMode      "Orbit"
AxisGrid       off
Snapshot       off
Suspend        off
SyncSnapshots  off
OpenScopeAtSimStart on
FigPos         "[452 44 407 353]"
CameraView     "[89945.9047576273 49093.2447020758 -7462.529306"
               "92949 -16.6995606364064 71 5.5 2.5389885178677 -5.30668596526891
               0.8084184343"
               "68868 0.0592442608385446 5.583225828012 14.2 1]"
InportDimsHint "[129 1]"
AxisProperties  off
YMin           "-44.6156897764664"
YMax           "11.2165685036536"
AxisColor      "[1 1 1]*.9"
XLabel         "Frequency"
YLabel         "Mag^2"
ZLabel         "History"
TriggerProperties off

```

```

TrigStartMode    "Immediately"
TrigStartT       "10"
TrigStartN       "1000"
TrigStartFcn     "trigPower"
TrigStopMode     "Never"
TrigStopT        "0.1"
TrigStopN        "10"
TrigStopFcn      "trigend"
TrigRearmMode    "User-defined"
TrigRearmT       "0"
TrigRearmN       "100"
TrigRearmFcn     "trigrearm"
XformProperties  off
XformMode        "FFT->Mag dB Fs/2"
XformFcn         "fliplr"
XformExpr        "20*log10(abs(u(129:2:end)))"
}
Block {
    BlockType      Reference
    Name           "Levinson-\nDurbin"
    Ports          [1, 1]
    Position       [415, 136, 475, 184]
    DialogController "dspDDGCreate"
    DialogControllerArgs "DataTag3"
    SourceBlock    "dspsolvers/Levinson-Durbin"
    SourceType     "Levinson-Durbin"
    coeffOutFcn    "K"
    outP           off
    zeroInpHandling on
    additionalParams off
    allowOverrides off
}

```

```

firstCoeffMode      "Binary point scaling"
firstCoeffWordLength "16"
firstCoeffFracLength "15"
secondCoeffMode     "Binary point scaling"
secondCoeffWordLength "16"
secondCoeffFracLength "15"
outputMode          "Same as input"
outputWordLength    "16"
outputFracLength     "15"
accumMode           "Same as input"
accumWordLength      "32"
accumFracLength      "30"
prodOutputMode      "Same as input"
prodOutputWordLength "32"
prodOutputFracLength "30"
roundingMode        "Floor"
overflowMode        off
}
Block {
  BlockType      SubSystem
  Name           "NT-specific Demo1"
  Ports          []
  Position       [35, 308, 130, 365]
  BackgroundColor "cyan"
  DropShadow     on
  NamePlacement  "alternate"
  ShowName       off
  OpenFcn        "dsplpc_win32"
  FontName       "Arial"
  FontSize       14
  FontWeight     "bold"

```

```

TreatAsAtomicUnit    off
MinAlgLoopOccurrences off
RTWSystemCode        "Auto"
MaskType              "NT-only Demo"
MaskDisplay           "disp('PC/Windows\\nLPC Demo')"
MaskIconFrame         on
MaskIconOpaque        on
MaskIconRotate        "none"
MaskIconUnits         "autoscale"
System {
Name                  "NT-specific Demo1"
Location              [51, 666, 589, 839]
Open                  off
ModelBrowserVisibility off
ModelBrowserWidth    200
ScreenColor           "white"
PaperOrientation      "landscape"
PaperPositionMode    "auto"
PaperType             "usletter"
PaperUnits            "inches"
ZoomFactor            "100"
}
}
Block {
BlockType             Reference
Name                  "Overlap\nAnalysis\nWindows"
Ports                 [1, 1]
Position              [205, 133, 255, 187]
SourceBlock           "dspbuff3/Buffer"
SourceType            "Buffer"
N                     "160"
}

```



```

V      "80"
ic     "0"
}
Block {
  BlockType      Reference
  Name           "Pre-Emphasis"
  Ports          [1, 1]
  Position       [95, 140, 165, 180]
  DialogController "dspDDGCreate"
  DialogControllerArgs "DataTag4"
  SourceBlock    "dsparch4/Digital Filter"
  SourceType     "Digital Filter"
  TypePopup     "FIR (all zeros)"
  IIRFiltStruct "Direct form II transposed"
  AllPoleFiltStruct "Direct form"
  FIRFiltStruct  "Direct form"
  CoeffSource    "Specify via dialog"
  NumCoeffs     "[1 -.95]"
  DenCoeffs     "[1 0.1]"
  BiQuadCoeffs  "[1 0.3 0.4 1 0.1 0.2]"
  LatticeCoeffs "[0.2 0.4]"
  denIgnore     on
  FiltPerSampPopup "One filter per frame"
  IC            "0"
  ICnum        "0"
  ICden        "0"
  additionalParams off
  allowOverrides on
  showCoeff    off
  firstCoeffMode "Same as input"
  firstCoeffWordLength "16"

```

firstCoeffFracLength "15"
 secondCoeffMode "Same as numerator"
 secondCoeffWordLength "16"
 secondCoeffFracLength "15"
 thirdCoeffMode "Same as input"
 thirdCoeffWordLength "16"
 thirdCoeffFracLength "15"
 showOut off
 outputMode "Same as input"
 outputWordLength "16"
 outputFracLength "15"
 showAcc off
 accumMode "Same as product output"
 accumWordLength "40"
 accumFracLength "30"
 showMpy off
 prodOutputMode "Same as input"
 prodOutputWordLength "32"
 prodOutputFracLength "30"
 showMem off
 memoryMode "Same as input"
 memoryWordLength "16"
 memoryFracLength "14"
 roundingMode "Floor"
 overflowMode off
 ScaleValues "1"
 scaleValueFracLength "14"
 tapSumMode "Same as input"
 tapSumWordLength "32"
 tapSumFracLength "30"
 stageIOMode "Same as input"

```

stageIOWordLength    "16"
stageInFracLength    "15"
stageOutFracLength   "15"
LockScale            off
}
Block {
  BlockType          ToWorkspace
  Name               "Re-synthesized\nSignal"
  Position           [875, 148, 915, 172]
  VariableName       "y_lpc"
  MaxDataPoints      "inf"
  SampleTime         "-1"
  SaveFormat         "Array"
}
Block {
  BlockType          Reference
  Name               "Reflection\nCoeffs"
  Ports              [1]
  Position           [535, 260, 595, 310]
  Orientation        "left"
  SourceBlock        "dpsnks4/Waterfall"
  SourceType         "Waterfall"
  ShowPortLabels     off
  TraceProperties    off
  NumTraces          "10"
  CMapStr            "autumn"
  TNewest            "0.7"
  TOldest            "0.1"
  HistoryLength      "50"
  HistoryFull        "Overwrite"
  UpdateInterval     "1"
}

```

ExportMode "All visible"
 MLEExportName "ExportData2"
 AutoExport off
 DisplayProperties off
 MouseMode "Orbit"
 AxisGrid on
 Snapshot off
 Suspend off
 SyncSnapshots off
 OpenScopeAtSimStart on
 FigPos "[35 41 410 358]"
 CameraView "[73343.0440471642 150383.489958622 -231607.0557"
 "69344 -0.0735265549347769 7.5 5.5 0.547487828138618 -2.62241953831449
 4.03911"
 "084299836 0.00240035045613671 1 7.88928028685195 5.25952019123463]"
 InportDimsHint "[12 1]"
 AxisProperties off
 YMin "-1.02418361455266"
 YMax "0.877130504683106"
 AxisColor "[1 1 1]*.9"
 XLabel "Coefficient"
 YLabel "RC Amplitude"
 ZLabel "History"
 TriggerProperties off
 TrigStartMode "Immediately"
 TrigStartT "10"
 TrigStartN "1000"
 TrigStartFcn "trigPower"
 TrigStopMode "Never"
 TrigStopT "0.1"
 TrigStopN "10"

```

TrigStopFcn      "trigend"
TrigRearmMode    "User-defined"
TrigRearmT       "0"
TrigRearmN       "100"
TrigRearmFcn     "trigrearm"
XformProperties  off
XformMode        "None"
XformFcn         "fliplr"
XformExpr        "20*log10(abs(u(129:2:end)))"
}
Block {
  BlockType      Reference
  Name           "Residual"
  Ports         [1]
  Position       [655, 40, 715, 90]
  NamePlacement  "alternate"
  SourceBlock    "dspsnks4/Waterfall"
  SourceType     "Waterfall"
  ShowPortLabels off
  TraceProperties off
  NumTraces      "10"
  CMapStr        "autumn"
  TNewest        "0.7"
  TOldest        "0.1"
  HistoryLength  "50"
  HistoryFull    "Overwrite"
  UpdateInterval "1"
  ExportMode     "All visible"
  MLExpertName   "ExportData2"
  AutoExport     off
  DisplayProperties off

```

```

MouseMode      "Orbit"
AxisGrid       on
Snapshot       off
Suspend        off
SyncSnapshots  off
OpenScopeAtSimStart  off
FigPos         "[524 276 410 358]"
CameraView     "[62164.2347969079 1792549.89917746 -405278.3703"
               "32737 -0.097024155318337 44.5 5.5 0.575539043328793 -33.5051317778483
               7.57548"
               "038023531 0.00104183380472551 1 92.0645761641202 10.3443344004629]"
InportDimsHint "[80 1]"
AxisProperties  off
YMin          "-0.580380532484202"
YMax          "0.386332221847528"
AxisColor     "[1 1 1]*.9"
XLabel        "Coefficient"
YLabel        "Amplitude"
ZLabel        "History"
TriggerProperties  off
TrigStartMode "Immediately"
TrigStartT    "10"
TrigStartN    "1000"
TrigStartFcn  "trigPower"
TrigStopMode  "Never"
TrigStopT     "0.1"
TrigStopN     "10"
TrigStopFcn   "trigend"
TrigRearmMode "User-defined"
TrigRearmT    "0"
TrigRearmN    "100"

```

```

TrigRearmFcn    "trigrearm"
XformProperties  off
XformMode       "None"
XformFcn        "fliplr"
XformExpr       "20*log10(abs(u(129:2:end)))"
}
Block {
  BlockType      SubSystem
  Name           "Subsystem1"
  Ports          []
  Position       [245, 326, 354, 365]
  ShowName       off
  OpenFcn        "if exist('mtlb','var'), sound(mtlb*0.32,8000); "
"else errordlg('Data not found in workspace - reload demo model.','Signal Proc"
"essing Blockset Demo Error'); end"
  TreatAsAtomicUnit  off
  MinAlgLoopOccurrences  off
  RTWSystemCode      "Auto"
  MaskDisplay        "disp('Original\\nSignal')"
  MaskIconFrame      on
  MaskIconOpaque     on
  MaskIconRotate     "none"
  MaskIconUnits      "autoscale"
  System {
    Name           "Subsystem1"
    Location       [203, 331, 569, 538]
    Open           off
    ModelBrowserVisibility  off
    ModelBrowserWidth  200
    ScreenColor    "white"
    PaperOrientation  "landscape"

```

```

PaperPositionMode "auto"
PaperType "usletter"
PaperUnits "inches"
ZoomFactor "100"
}
}
Block {
    BlockType SubSystem
    Name "Subsystem3"
    Ports []
    Position [363, 327, 468, 365]
    ShowName off
    OpenFcn "if exist('y_lpc','var'), sound(y_lpc,8000); els"
    "e errordlg('You must run the simulation first.','Signal Processing Blockset D"
    "emo Error'); end"
    TreatAsAtomicUnit off
    MinAlgLoopOccurrences off
    RTWSystemCode "Auto"
    MaskDisplay "disp('Processed\\nSignal')"
    MaskIconFrame on
    MaskIconOpaque on
    MaskIconRotate "none"
    MaskIconUnits "autoscale"
    System {
    Name "Subsystem3"
    Location [203, 331, 569, 538]
    Open off
    ModelBrowserVisibility off
    ModelBrowserWidth 200
    ScreenColor "white"
    PaperOrientation "landscape"

```



```

PaperPositionMode "auto"
PaperType "usletter"
PaperUnits "inches"
ZoomFactor "100"
}
}
Block {
  BlockType Reference
  Name "Time-Varying \nAnalysis Filter"
  Ports [2, 1]
  Position [525, 130, 595, 170]
  DialogController "dspDDGCreate"
  DialogControllerArgs "DataTag5"
  SourceBlock "dsparch4/Digital Filter"
  SourceType "Digital Filter"
  TypePopup "FIR (all zeros)"
  IIRFiltStruct "Direct form II transposed"
  AllPoleFiltStruct "Direct form"
  FIRFiltStruct "Lattice MA"
  CoeffSource "Input port(s)"
  NumCoeffs "[1 -.95]"
  DenCoeffs "[1 0.1]"
  BiQuadCoeffs "[1 0.3 0.4 1 0.1 0.2]"
  LatticeCoeffs "[0.2 0.4]"
  denIgnore on
  FiltPerSampPopup "One filter per frame"
  IC "0"
  ICnum "0"
  ICden "0"
  additionalParams off
  allowOverrides on

```

showCoeff off
firstCoeffMode "Same as input"
firstCoeffWordLength "16"
firstCoeffFracLength "15"
secondCoeffMode "Same as numerator"
secondCoeffWordLength "16"
secondCoeffFracLength "15"
thirdCoeffMode "Same as input"
thirdCoeffWordLength "16"
thirdCoeffFracLength "15"
showOut off
outputMode "Same as input"
outputWordLength "16"
outputFracLength "15"
showAcc off
accumMode "Same as product output"
accumWordLength "40"
accumFracLength "30"
showMpy off
prodOutputMode "Same as input"
prodOutputWordLength "32"
prodOutputFracLength "30"
showMem off
memoryMode "Same as input"
memoryWordLength "16"
memoryFracLength "14"
roundingMode "Floor"
overflowMode off
ScaleValues "1"
scaleValueFracLength "14"
tapSumMode "Same as input"

```

tapSumWordLength    "32"
tapSumFracLength    "30"
stageIOMode         "Same as input"
stageIOWordLength   "16"
stageInFracLength   "15"
stageOutFracLength  "15"
LockScale           off
}
Block {
  BlockType          Reference
  Name               "Time-Varying \nSynthesis Filter"
  Ports              [2, 1]
  Position            [670, 140, 740, 180]
  DialogController   "dspDDGCreate"
  DialogControllerArgs "DataTag6"
  SourceBlock        "dsparch4/Digital Filter"
  SourceType         "Digital Filter"
  TypePopup          "IIR (all poles)"
  IIRFiltStruct      "Direct form II transposed"
  AllPoleFiltStruct  "Lattice AR"
  FIRFiltStruct      "Direct form"
  CoeffSource        "Input port(s)"
  NumCoeffs          "[1 -.95]"
  DenCoeffs          "[1 0.1]"
  BiQuadCoeffs      "[1 0.3 0.4 1 0.1 0.2]"
  LatticeCoeffs      "[0.2 0.4]"
  denIgnore          on
  FiltPerSampPopup   "One filter per frame"
  IC                 "0"
  ICnum              "0"
  ICden              "0"

```

additionalParams off
allowOverrides on
showCoeff off
firstCoeffMode "Same as input"
firstCoeffWordLength "16"
firstCoeffFracLength "15"
secondCoeffMode "Same as numerator"
secondCoeffWordLength "16"
secondCoeffFracLength "15"
thirdCoeffMode "Same as input"
thirdCoeffWordLength "16"
thirdCoeffFracLength "15"
showOut off
outputMode "Same as input"
outputWordLength "16"
outputFracLength "15"
showAcc off
accumMode "Same as product output"
accumWordLength "40"
accumFracLength "30"
showMpy off
prodOutputMode "Same as input"
prodOutputWordLength "32"
prodOutputFracLength "30"
showMem off
memoryMode "Same as input"
memoryWordLength "16"
memoryFracLength "14"
roundingMode "Floor"
overflowMode off
ScaleValues "1"

```

scaleValueFracLength  "14"
tapSumMode            "Same as input"
tapSumWordLength     "32"
tapSumFracLength     "30"
stageIOMode          "Same as input"
stageIOWordLength    "16"
stageInFracLength    "15"
stageOutFracLength   "15"
LockScale            off
}
Block {
  BlockType      Reference
  Name           "Window"
  Ports          [1, 1]
  Position       [275, 134, 325, 186]
  DialogController "dspDDGCreate"
  DialogControllerArgs "DataTag7"
  SourceBlock    "dspsigops/Window\nFunction"
  SourceType     "Window Function"
  winmode        "Apply window to input"
  wintype        "Hamming"
  sampmode       "Continuous"
  samptime       "1"
  N              "64"
  Rs             "50"
  beta           "10"
  winsamp        "Symmetric"
  UserWindow     "hamming"
  OptParams      off
  UserParams     "{1.0}"
  additionalParams off
}

```

```

allowOverrides    on
dataType         "double"
isSigned         on
wordLen          "16"
udDataType       "sfixed(16)"
fracBitsMode     "Best precision"
numFracBits      "15"
firstCoeffMode   "Same word length as input"
firstCoeffWordLength "16"
firstCoeffFracLength "15"
prodOutputMode   "Inherit via internal rule"
prodOutputWordLength "16"
prodOutputFracLength "15"
outputMode       "Same as input"
outputWordLength "16"
outputFracLength "15"
roundingMode     "Floor"
overflowMode     off
LockScale        off
}
Block {
  BlockType      Reference
  Name           "Zero Pad"
  Ports          [1, 1]
  Position       [660, 267, 705, 303]
  ShowName       off
  SourceBlock    "dspsigops/Zero Pad"
  SourceType     "Zero Pad"
  padSigAt       "End"
  zpadAlong      "Columns"
  padNumOutRowsSpecMethod "User-specified"
}

```

```

numOutRows      "256"
padNumOutColsSpecMethod "User-specified"
numOutCols      "1"
trunc_flag      "None"
}
Block {
  BlockType      SubSystem
  Name           "background1"
  Ports         []
  Position       [79, 98, 606, 233]
  BackgroundColor "lightBlue"
  ShowName       off
  TreatAsAtomicUnit off
  MinAlgLoopOccurrences off
  RTWSystemCode  "Auto"
  MaskDisplay    "disp()"
  MaskIconFrame  on
  MaskIconOpaque on
  MaskIconRotate "none"
  MaskIconUnits  "autoscale"
  System {
    Name         "background1"
    Location     [209, 325, 575, 532]
    Open         off
    ModelBrowserVisibility off
    ModelBrowserWidth 200
    ScreenColor  "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType     "usletter"
    PaperUnits    "inches"
  }
}

```

```

ZoomFactor    "100"
}
}
Block {
  BlockType    SubSystem
  Name         "background2"
  Ports       []
  Position     [647, 93, 854, 229]
  BackgroundColor "lightBlue"
  ShowName     off
  TreatAsAtomicUnit off
  MinAlgLoopOccurrences off
  RTWSystemCode "Auto"
  MaskDisplay  "disp()"
  MaskIconFrame on
  MaskIconOpaque on
  MaskIconRotate "none"
  MaskIconUnits "autoscale"
  System {
    Name       "background2"
    Location   [205, 329, 571, 536]
    Open       off
    ModelBrowserVisibility off
    ModelBrowserWidth 200
    ScreenColor "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType    "usletter"
    PaperUnits   "inches"
    ZoomFactor   "100"
  }
}

```



```

}
Line {
  SrcBlock      "8 kHz\nSignal\n\"MATLAB\""
  SrcPort       1
  DstBlock      "Pre-Emphasis"
  DstPort       1
}
Line {
  SrcBlock      "Pre-Emphasis"
  SrcPort       1
  Points        [20, 0]
  Branch {
  DstBlock      "Overlap\nAnalysis\nWindows"
  DstPort       1
  }
  Branch {
  Points        [0, -45; 320, 0]
  DstBlock      "Time-Varying \nAnalysis Filter"
  DstPort       1
  }
}
Line {
  SrcBlock      "Overlap\nAnalysis\nWindows"
  SrcPort       1
  DstBlock      "Window"
  DstPort       1
}
Line {
  SrcBlock      "Autocorrelation"
  SrcPort       1
  DstBlock      "Levinson-\nDurbin"
}

```

```

    DstPort      1
}
Line {
    SrcBlock     "Window"
    SrcPort      1
    DstBlock     "Autocorrelation"
    DstPort      1
}
Line {
    SrcBlock     "Time-Varying \nAnalysis Filter"
    SrcPort      1
    Points       [20, 0]
    Branch {
        DstBlock "Time-Varying \nSynthesis Filter"
        DstPort  1
    }
    Branch {
        Points   [0, -85]
        DstBlock "Residual"
        DstPort  1
    }
}
Line {
    SrcBlock     "Levinson-\nDurbin"
    SrcPort      1
    Points       [5, 0]
    Branch {
        Points   [0, 45; 135, 0]
        Branch {
            Points [0, -35]
            DstBlock "Time-Varying \nSynthesis Filter"

```

```

    DstPort      2
  }
  Branch {
    Points      [0, 80]
    Branch {
      DstBlock   "Reflection\nCoeffs"
      DstPort    1
    }
    Branch {
      DstBlock   "Zero Pad"
      DstPort    1
    }
  }
}
}
  Branch {
    DstBlock     "Time-Varying \nAnalysis Filter"
    DstPort      2
  }
}
  Line {
    SrcBlock     "Time-Varying \nSynthesis Filter"
    SrcPort      1
    DstBlock     "De-Emphasis"
    DstPort      1
  }
  Line {
    SrcBlock     "De-Emphasis"
    SrcPort      1
    DstBlock     "Re-synthesized\nSignal"
    DstPort      1
  }
}

```

```

Line {
  SrcBlock      "Zero Pad"
  SrcPort       1
  DstBlock      "FFT1"
  DstPort       1
}
Line {
  SrcBlock      "FFT1"
  SrcPort       1
  DstBlock      "LPC\nSpectrum"
  DstPort       1
}
Annotation {
  Name          "\nLPC Analysis and Synthesis of Speech\n"
  Position      [368, 52]
  BackgroundColor "cyan"
  FontName      "Arial"
  FontSize      14
  FontWeight    "bold"
}
Annotation {
  Name          "Analysis"
  Position      [336, 248]
  FontName      "Arial"
  FontSize      11
  FontWeight    "bold"
}
Annotation {
  Name          "Synthesis"
  Position      [748, 251]
  FontName      "Arial"
}

```

```

    FontSize      11
    FontWeight    "bold"
  }
  Annotation {
    Name          " Audio playback in MATLAB:"
    Position      [358, 309]
    BackgroundColor "cyan"
    FontSize      14
  }
}
}
}
MatData {
  NumRecords     8
  DataRecord {
    Tag          DataTag7
    Data         " %>30 . : 8 ( 0 % "
"\ " $ ! 0 . . 8 ( ! % \ " $ "
"& 0 0 !@ %=I;F1O=P "
  }
  DataRecord {
    Tag          DataTag6
    Data         " %>30 . < 8 ( 0 % "
"\ " $ ! 0 . 0 8 ( ! % \ " $ "
"- 0 0 #0 $1I9VET86Q&:6QT97( "
  }
  DataRecord {
    Tag          DataTag5
    Data         " %>30 . < 8 ( 0 % "
"\ " $ ! 0 . 0 8 ( ! % \ " $ "
"- 0 0 #0 $1I9VET86Q&:6QT97( "
  }
}

```

```

DataRecord {
  Tag      DataTag4
  Data     " % )30 . < 8 ( 0 % "
"\ " $ ! 0 . 0 8 ( ! % \ " $ "
"- 0 0 #0 $1I9VET86Q&:6QT97( "
}
DataRecord {
  Tag      DataTag3
  Data     " % )30 . : 8 ( 0 % "
"\ " $ ! 0 . . 8 ( ! % \ " $ "
"( 0 0 \ " $QE=FEN<V]N"
}
DataRecord {
  Tag      DataTag2
  Data     " % )30 . 8 8 ( 0 % "
"\ " $ ! 0 . , 8 ( ! % \ " $ "
"# 0 0 ,1D94 "
}
DataRecord {
  Tag      DataTag1
  Data     " % )30 . < 8 ( 0 % "
"\ " $ ! 0 . 0 8 ( ! % \ " $ "
"- 0 0 #0 $1I9VET86Q&:6QT97( "
}
DataRecord {
  Tag      DataTag0
  Data     " % )30 . : 8 ( 0 % "
"\ " $ ! 0 . . 8 ( ! % \ " $ "
"( 0 0 \ " $%U=&]C;W)R"
}
}

```

Appendix D

Tensile properties of lead-free solders

Tensile properties of lead-free solders (courtesy to National institute of standards and Technology, USA)

Chemical Composition	Elastic Modulus		0.2%yield Strength		Tensile Strength		Relative Elongation (%)		Strength Coefficient		
	Wt%	ksi	GPa	psi	Mpa	psi	MPa	Uniform	Total	psi	MPa
Sn-37Pb		2,273	15.7	3,950	27.2	4,442	30.6	3	48	4,917	33.9
Sn-2Ag-36Pb		2,617	18.0	6,287	43.3	6,904	47.6	1	31	7,223	49.8
Sn-97Pb		2,753	19.0	1,126	7.8	2,383	16.4	27	38	3,934	27.1
Sn-3.5Ag		3,793	26.2	3,256	22.5	3,873	26.7	3	24	4,226	29.1
Sn-5Sb		6,460	44.5	3,720	25.7	5,110	35.2	3	22	4,177	28.8
Sn-58Bi		1,720	11.9	7,119	49.1	8,766	60.4	3	46	9,829	67.8
Sn-3.5Ag-0.5Sb-1Cd				7,545	52.0				15		
Sn-75Pb				3,426	23.6				53		
Sn-50Bi				8,263	57.0	8,965	61.8		53		
Sn-52Bi				6,414	44.2	8,834	60.9		57		
Sn-2Ag-46Bi-4Cu				9,806	67.6	10,070	69.4		3		
Sn-56Bi-2In				7,224	49.8	8,429	58.1		116		
Sn-2Ag-1.5Sb-29Pb				6,489	44.7	6,865	47.3		25		
Sn-3Ag-4Cu				6,276	43.3	7,006	48.3		22		
Sn-2.5Ag-2Bi-1.5Sb				7,070	48.7	8,117	56.0		21		
Sn-3Ag-1Bi-				8,361	57.6	9,256	63.8		21		

1.5Sb										
Sn-2Ag-9.8Bi-9.8In			14,560	100.4	15,380	106.0		7		
Sn-57Bi-2In			7,304	50.4	8,436	58.2		72		
Sn-2Ag-57Bi			9,487	65.4	10,390	71.6		31		
Sn-57Bi-2Sb			8,521	58.8	9,586	66.1		47		
Sn-57Bi-1Sb			8,285	57.1	8,944	61.7		60		
Sn-2Ag-56Bi-1.5Sb			9,063	62.5	9,946	68.6		27		
Sn-3Ag-55.5Bi-1.5Sb			8,665	59.7	9,379	64.7		45		
Sn-3Ag-55Bi-2Sb			8,984	61.9	9,807	67.6		44		
Sn-3Ag-54Bi-2In-2Sb			5,055	34.9	11,640	80.3		13		
Sn-3Ag-54Bi-2Cu-2Sb			11,440	78.9	12,280	84.7		4		
Sn-3Ag-2Sb			5,749	39.6	6,124	42.2		25		
Sn-3Ag-2Cu-2Sb			6,684	46.1	7,655	52.8		32		
Sn-3Ag-2Bi-2Sb			6,918	47.7	9,212	63.5		36		
Sn-3Ag-2Bi			5,463	37.7	7,930	54.7	6	30	8,946	61.7
Sn-2.5Ag-2Bi			6,592	45.5	7,564	52.2		26		
Sn-2Bi-1.5Cu-3Sb			7,343	50.6	9,350	64.5		28		
Sn-2Bi-8In			7,160	49.4	7,970	55.0		25		
Sn-10Bi-20In					6,938	47.8		4		

Sn-9Zn			7,478	51.6	7,708	53.1		27		
Sn-2Ag- 7.5Bi-0.5Cu			12,370	85.3	13,440	92.7		12		
Sn-2.6Ag- 0.8Cu- 0.5Sb			3,311	22.8	3,749	25.8	2	9	4,536	31.3
Sn-0.5Ag- 4Cu			3,724	25.7	4,312	29.7		27		
Sn-8.8In- 7.6Zn			6,033	41.6	6,445	44.4		14		
Sn-20In- 2.8Zn			5,095	35.1	5,381	37.1		31		
Sn-31.5Bi- 3Zn			10,500	72.4	11,210	77.3		53		
Sn-3.5Ag- 1.5In			4,616	31.8	4,987	34.4		26		
Sn-2Ag- 0.5Bi-7.5Sb			8,230	56.7	8,773	60.5		19		
Sn-0.2Ag- 2Cu-0.8SB			3,758	25.9	4,323	29.8		27		
Sn-2.5Ag- 19.5Bi			12,070	83.2	13,450	92.7		17		
Sn-3Ag- 41Bi			9,287	64.0	10,130	69.8		39		
Sn-55Bi- 2Cu			8,985	62.0	9,478	65.4		41		
Sn-48Bi- 2Cu			8,899	61.4	9,495	65.5		19		
Sn-57Bi			7,972	55.0	8,540	58.9		77		
Sn-56.7Bi- 0.3Cu-1In			8,359	57.6	8,985	62.0		38		
Sn-3.4Ag- 4.8Bi			6,712	46.3	10,349	71.4	5	16	17,795	122.7

Appendix E

Program 5

% Solder ball melting simulation using ANSYS

finish

/clear

/title, Hu's thesis

/prep7

! define geometry

k,1,0,0

k,2,0.02,0

k,3,0.02,0.02

k,4,0,0.02

a,1,2,3,4

! mesh 2D areas

ET,1,Plane55

MP,Dens,1,920

mp,c,1,2040

mp,kxx,1,1.8

```
esize,0.0005
amesh,all

finish
/solu

antype,4

time,60

nropt,full
lumpm,0
nsubst,20
neqit,100
autots,off
lnsrch,on
outres,all,all
kbc,1

IC,all,temp,228

nset,s,ext
sf,all,conv,10,328
nset,al
/gst,off

solve
finish

/post1
set,last
```

```
etable,melty,temp,  
esel,s,etab,melty,273  
finish
```

```
/solu  
antype,rest  
ekill,all  
esel,all
```

```
finish
```

```
/post1  
set,last  
esel,s,live  
plnsol,temp
```

VITA AUCTORIS

Name: Rongrong Hu

Place of Birth: Nanjing, China

Year of birth: 1981

Education: B. Eng. Automatic Control,
Nanjing University of Technology, 2002

M.A.Sc. Electrical and Computer Engineering,
University of Windsor, 2006