ADAPTIVE ARRAY SIGNAL PROCESSING USING THE CONCENTRIC RING ARRAY AND THE SPHERICAL ARRAY

A Dissertation presented to the Faculty of the Graduate School University of Missouri

In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy by

Luis M. Vicente Dr. K. C. Ho, Dissertation Supervisor May 2009
The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

**ADAPTIVE ARRAY SIGNAL PROCESSING USING THE CONCENTRIC RING ARRAY AND THE SPHERICAL ARRAY**

presented by Luis M. Vicente

a candidate for the degree of Doctor of Philosophy

and hereby certify that in their opinion it is worthy of acceptance.

______________________________________________
Dr. K. C. Ho

______________________________________________
Dr. James Keller

______________________________________________
Dr. Satish Nair

______________________________________________
Dr. Zhihai (Henry) He

______________________________________________
Dr. Cerry M. Klein
No eran molinos...

... para Magdalena, Clara Luz

y Juliana.
ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Ho for his guidance, help, and support to complete this work. I also want to thank my committee members Dr. Keller, Dr. Nair, Dr. Klein, and Dr. He for their time, support, and because they believed in me.

I would like to thank the University of Missouri for the opportunity to study here in this great campus that I will always keep in my heart.

I would also like to thank Dra. Pabón, Dr. Teixeira, Prof. Pérez, and many more at the Universidad Politécnica de Puerto Rico, who helped me in all the ways they could. I am committed to work with them to the best of my abilities to make our institution stand out in all aspects.

I would like to thank my fellow students Xiaoning Lu, Le Yang, Ming Sun, Zhenhua Ma, Jun Tao, and the visiting professors Dr. Fucheng Guo and Dr. Jung Sik Jeong for their good comments and help.

I would like to thank my good friends Sara, Adrian, Perro Loco Goodman, and all the Desterrados. I can not forget the Flat Branch and all the friends that work there and made my hard times, easier.

Finally, I would like to thank my beloved spouse Magdalena, my daughters Clara Luz and Juliana, my parents Isabel and Eladio, my sister Isabel and my brother Eladio, and all my relatives both in Spain and Puerto Rico for their unconditional love and encouragement. This accomplishment is also yours.
## Contents

ACKNOWLEDGEMENTS ........................................... ii

LIST OF TABLES ............................................. x

LIST OF FIGURES ............................................ xii

ABSTRACT ...................................................... xvi

CHAPTER

1 Introduction ................................................. 1
  1.1 A not so Imaginary Tale ..................................... 1
  1.2 Bird Strikes and Security Measures ........................... 2
  1.3 Security in Puerto Rico ...................................... 5
  1.4 Background and Previous Works on Array Signal Processing and Beam-
      forming .................................................. 6
  1.4.1 Signal Field, Near and Far Field ......................... 7
  1.4.2 Array Geometry ........................................ 8
  1.4.3 Sensor Coefficients and Beamforming .................... 10
  1.4.4 Element Space and Beamspace Beamformers ............... 13
  1.4.5 Partial Adaptive Beamforming ........................... 15
  1.5 Motivation ............................................... 16
1.5.1 Assumptions and Limitations ............................................ 18
1.6 Thesis Contribution .......................................................... 19
1.6.1 Partial Adaptive Concentric Ring Array .............................. 19
1.6.2 Partial Adaptive Spherical Array ........................................ 20
1.6.3 Beamformer Robustness Against Sensor Position Errors ........ 20
1.7 Thesis Outline ............................................................... 21
1.8 Notation and Acronyms ...................................................... 21
1.9 Chapter Summary ............................................................. 26

2 Fundamentals of Beamforming.................................................. 27
2.1 Coordinate Systems in Array Signal Processing ....................... 27
  2.1.1 Cartesian Coordinates ................................................... 28
  2.1.2 Spherical Coordinates .................................................. 29
2.2 Propagating Waves .......................................................... 29
  2.2.1 The Wave Equation ...................................................... 30
  2.2.2 Solution to the Wave Equation ....................................... 30
  2.2.3 The Array Steering Vector ............................................. 32
  2.2.4 Phase Shift, Delay ...................................................... 32
2.3 Modeling the Signal Field .................................................... 33
  2.3.1 Signal of Interest Model ................................................. 33
  2.3.2 Interferences Model ..................................................... 34
  2.3.3 Background Noise Model .............................................. 34
  2.3.4 Beamformer Input Signal Model ..................................... 35
2.4 Array Geometry ............................................................. 35
  2.4.1 Spatial Sampling Nyquist Criterion .................................. 35
  2.4.2 Array Aperture .......................................................... 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.2</td>
<td>Modified Type I Partial Adaptive Beamformer</td>
<td>89</td>
</tr>
<tr>
<td>5.3</td>
<td>CBSES Partial Adaptive Beamformer</td>
<td>89</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Analysis</td>
<td>92</td>
</tr>
<tr>
<td>5.4</td>
<td>Simulations and Results</td>
<td>93</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>98</td>
</tr>
<tr>
<td>5.6</td>
<td>Acknowledgments</td>
<td>98</td>
</tr>
<tr>
<td>6</td>
<td>Beamforming Using the Spherical Array</td>
<td>99</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>99</td>
</tr>
<tr>
<td>6.2</td>
<td>Previous Works on Spherical Array</td>
<td>100</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Early works</td>
<td>100</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Recent works</td>
<td>101</td>
</tr>
<tr>
<td>6.3</td>
<td>Spherical Array Sensor Arrangements</td>
<td>103</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Equiangle Sensor Arrangement</td>
<td>103</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Sengupta Sensor Arrangement</td>
<td>105</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Gaussian Sampling Sensor Arrangement</td>
<td>106</td>
</tr>
<tr>
<td>6.4</td>
<td>Proposed Spherical Array Sensor Arrangements</td>
<td>106</td>
</tr>
<tr>
<td>6.4.1</td>
<td>First Proposed Sampling Sensor Arrangement</td>
<td>106</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Second Proposed Sampling Sensor Arrangement</td>
<td>109</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Comparison Between the Different Sensor Arrangements</td>
<td>110</td>
</tr>
<tr>
<td>6.5</td>
<td>Study of Narrowband Beamforming Using the Spherical Array</td>
<td>111</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Beamformer Output Signal to Noise Ratio</td>
<td>111</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Beamformer Output Signal to Interference Plus Noise Ratio</td>
<td>113</td>
</tr>
<tr>
<td>6.6</td>
<td>Computational Complexity Between the Fully and the Partial Adaptive Beamformers</td>
<td>116</td>
</tr>
<tr>
<td>6.6.1</td>
<td>Fully Adaptive Normalized Least Mean Squares</td>
<td>117</td>
</tr>
</tbody>
</table>
6.6.2 Fully Adaptive Recursive Least Squares ............... 118
6.6.3 Partial Adaptive Normalized Least Mean Squares .......... 118
6.6.4 Partial Adaptive Recursive Least Squares ............... 119
6.7 Convergence Speed .................................................. 120
6.7.1 Convergence Speed of the NLMS Algorithm ............. 120
6.7.2 Convergence Speed of the RLS Algorithm ............... 121
6.8 Narrowband Beamforming Simulations ......................... 121
6.8.1 Steady-State $SINR_0$ Monte Carlo Simulations .......... 122
6.8.2 Monte Carlo Simulations of $SINR_0$ Along Adaptation ... 122
6.8.3 Simulations With Change of interference DOAs .......... 125
6.8.4 Computational Complexity Example ......................... 127
6.9 Broadband Beamforming Using the Spherical Array ......... 128
6.9.1 The Frequency Domain Broadband Beamformer .......... 128
6.9.2 Using Nesting on Spherical Array ......................... 128
6.9.3 Design of a Nested SA for Acoustic Signals ............ 130
6.10 Phase Mode Beamforming with Spherical Arrays .......... 133
6.10.1 Spherical Harmonics ............................................. 133
6.10.2 Obtaining the $c_{o,m}^k$ Coefficients ....................... 135
6.10.3 Order Limit of the Phase Mode Beamformer ............ 136
6.10.4 Beamforming of a Single Plane Wave ..................... 137
6.10.5 Radius Limit in Phase Mode Beamforming ............... 137
6.10.6 Equivalent Element Space Weights for Spherical Harmonics . 138
6.10.7 Example of Phase Mode Beamforming With Different Sensor Arrangements .......... 139
6.10.8 Comparison Between Phase Mode and Element Space Beamforming Example .......... 140
7 Robustness Against Sensor Position Errors 144

7.1 Literature Review 144

7.1.1 Remarks 147

7.2 Objective 147

7.3 Location Error Model for Element Space Beamformers 148

7.3.1 Steering Vectors Under Location Errors 148

7.3.2 Beamformer Input, Output Signals Under Location Errors 150

7.3.3 Beamformer Output Signal to Interference Plus Noise Ratio 151

7.3.4 Two Different Types of Sensor Position Errors 151

7.4 Unknown Time Invariant Location Errors $\Delta \textbf{p}_k$ 152

7.4.1 Classical Robustness Method 153

7.5 Proposed Robust Method for Time Invariant Location Errors 154

7.5.1 Example. Ideal case: Known \( C_s \) 156

7.5.2 Example. Practical case: Known \( s_0 \) and \( \sigma_p^2 \) Using SA 160

7.5.3 Example. Practical case: Known \( s_0 \) and \( \sigma_p^2 \) Using CRA 162

7.6 Modeling of Time Variant Random Location Errors $\Delta \textbf{p}_k$ 163

7.6.1 Expected Beamformer Output Under Location Errors 163

7.6.2 Expected Beampattern Under Location Errors 165

7.7 Proposed Robust Beamformer for Time Variant Location Errors 166

7.7.1 Correlation Matrices Under Location Errors 166

7.7.2 $SINR_0$ Under Location Errors 168

7.7.3 Proposed Method for Robustness 169

7.7.4 Example. Robustness Against Time Variant Location Errors 171

7.8 Summary 172
8 Summary and Future Research

8.1 Completed Research

8.1.1 Using the Prior Knowledge in the Partial Adaptive Beamformer

8.1.2 Optimization of the Penalty Factor

8.1.3 Combined Beamspace Element Space Beamformer

8.1.4 Partial Adaptive Beamforming With Spherical Array

8.1.5 Robust Beamforming Against Sensor Position Errors

8.2 Published Papers on Completed Research

8.3 Future Research

A The Discrete Fourier Transform (DFT)

B Delay-and-sum Weight Vector with Lagrange Multipliers

C Optimum Weight Vector with Lagrange Multipliers

D Eigenvalue Analysis Example

D.0.1 Eigenvalue Spread

D.0.2 Initial and Steady-state Time Constants

D.0.3 Misadjustment

E Function of Gaussian Random Variables

Bibliography

VITA
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>22</td>
</tr>
<tr>
<td>1.2</td>
<td>23</td>
</tr>
<tr>
<td>1.3</td>
<td>24</td>
</tr>
<tr>
<td>1.4</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>67</td>
</tr>
<tr>
<td>5.1</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>97</td>
</tr>
<tr>
<td>5.3</td>
<td>97</td>
</tr>
<tr>
<td>6.1</td>
<td>110</td>
</tr>
<tr>
<td>6.2</td>
<td>123</td>
</tr>
<tr>
<td>6.3</td>
<td>123</td>
</tr>
<tr>
<td>6.4</td>
<td>123</td>
</tr>
<tr>
<td>6.5</td>
<td>125</td>
</tr>
<tr>
<td>6.6</td>
<td>126</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Arbitrary array inside a field composed of the signal of interest (SOI), interferences, and the background noise.</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>1.2</td>
<td>Near field, far field wavefront curvatures.</td>
<td>7</td>
</tr>
<tr>
<td>1.3</td>
<td>Uniform Linear and Ring Arrays</td>
<td>8</td>
</tr>
<tr>
<td>1.4</td>
<td>Narrowband beamformer.</td>
<td>10</td>
</tr>
<tr>
<td>1.5</td>
<td>Time domain broadband beamformer using FIR</td>
<td>12</td>
</tr>
<tr>
<td>1.6</td>
<td>Frequency domain broadband beamformer using DFT</td>
<td>12</td>
</tr>
<tr>
<td>1.7</td>
<td>Element space and beamspace beamformers</td>
<td>13</td>
</tr>
<tr>
<td>2.1</td>
<td>Coordinate systems.</td>
<td>28</td>
</tr>
<tr>
<td>2.2</td>
<td>Array with sensor coordinates and impinging field with an arbitrary DOA</td>
<td>31</td>
</tr>
<tr>
<td>2.3</td>
<td>ULA with $K$ elements.</td>
<td>37</td>
</tr>
<tr>
<td>2.4</td>
<td>Ring array with $K$ elements.</td>
<td>38</td>
</tr>
<tr>
<td>2.5</td>
<td>Spherical array with $K$ elements.</td>
<td>39</td>
</tr>
<tr>
<td>2.6</td>
<td>Narrowband Beamformer.</td>
<td>42</td>
</tr>
<tr>
<td>2.7</td>
<td>ULA Beampattern using delay-and-sum weights</td>
<td>52</td>
</tr>
<tr>
<td>2.8</td>
<td>ULA Beampattern using MVDR weights</td>
<td>53</td>
</tr>
<tr>
<td>3.1</td>
<td>Concentric Ring Array (CRA).</td>
<td>60</td>
</tr>
</tbody>
</table>
3.2 Block Diagram of the proposed partially adaptive array. 62
3.3 Residual interference and noise power in log scale with respect to iterations. A: fully adaptive. B: original partially adaptive as in [4]. C: improved partially, one interference known, two unknown. D: improved partially, two interferences known, one unknown. 65

4.1 Concentric Ring Array (CRA). 72
4.2 Residual error vs. log $\alpha$ for two field scenarios. 75
4.3 Proposed beamformer block diagram. 79
4.4 Scenario 1: output $SINR$ vs. iterations for the proposed beamformer (upper plot) and the previous beamformers (lower plot). 82
4.5 Scenario 2: output $SINR$ vs. iterations for the proposed beamformer (upper plot) and the previous beamformers (lower plot). 82

5.1 Concentric Ring Array (CRA). 87
5.2 CBSES Partial Adaptive Beamformer, block diagram. 90
5.3 Beampattern and residual error level vs. iterations of Fully Adaptive, Type I, Modified Type I, and CBSES arrays for scenario A. 94
5.4 Residual error level vs. iterations of Fully Adaptive, Type I, Modified Type I, and CBSES arrays for scenario B (upper plot) and scenario C (lower plot). 94

6.1 288 elements, equiangle spherical array. 104
6.2 182 elements, Sengupta spherical array. 105
6.3 288 elements, Gaussian sampled spherical array. 107
6.4 186 elements, first proposed spherical array. 108
6.5 200 elements, second proposed spherical array. 109
6.6 Fully Adaptive Beamformer, block diagram. ........................................ 117
6.7 Partial Adaptive Beamformer, block diagram. ........................................ 119
6.8 $SINR_0$ adaptation for fully and partial adaptive beamformers. .............. 124
6.9 $SINR_0$ adaptation results, 20 ensembles. ........................................... 126
6.10 Frequency domain broadband beamformer using DFT. .......................... 129
6.11 nested spherical array. ........................................................................ 130
6.12 $b_\theta(\kappa r)$ vs. Order vs. $\kappa r$. ......................................................... 138
6.13 Beampatterns spherical array. ............................................................... 139
6.14 Beampatterns spherical array. ............................................................... 140
6.15 Directivity index and Array gain vs. $\kappa r$. ........................................ 141
7.1 Expected Array gain vs. $\sigma_p^2$. ......................................................... 157
7.2 Array gain vs. $SINR_i$. ........................................................................ 159
7.3 Array gain vs. $\sigma_p^2$. ........................................................................ 161
7.4 Array gain vs. $SINR_i$. ........................................................................ 161
7.5 $T = 600$. Medium sample size scenario. .............................................. 162
7.6 Array gain vs. $\sigma_p^2$. ........................................................................ 172
7.7 Array gain vs. $SINR_i$. ........................................................................ 173
Array signal processing is an interesting field that uses sensors placed in particular geometric arrangements for the detection and processing of signals. One of the most significant features is that the array is able to perform spatial discrimination besides the well known frequency filtering approach. The spatial filtering process is referred to as beamforming. The objective of beamforming is to enhance a desired signal meanwhile canceling interferences coming from other directions and suppressing the background noise. The arrangement of the sensing elements is essential in determining the performance of source localization and beamforming.

Among different geometry arrangements, the ring array is preferable for 3-D beamforming because provides full azimuth coverage and reduces the cone of uncertainty present in the uniform linear array (ULA) to just two direction of arrivals (DOAs) while maintaining an azimuthal uniform beampattern. The concentric ring array (CRA) has additional flexibility in adaptive beamforming. Furthermore, it can utilize nested array design and achieves frequency invariant characteristics. The spherical array (SA) has all the advantages of the ring array plus maintaining a uniform beampattern in all directions and eliminating the DOA uncertainty.

This thesis introduces new methods for the partial adaptive beamforming using CRA and SA for acoustic signals on a partially known interference environment. This work is originally based on the element space partial adaptive beamformers of D. Abraham and H. Cox. More recently, Y. Li employs a CRA where the whole array is decomposed into sub-arrays that perform element space individual beamforming using intra-ring weights. Then, the sub-array outputs are combined together with adaptive inter-ring weights to form the overall beamformer output.

The first contribution of this thesis resides in novel methods to choose the intra-
ring and inter-ring weights. They are designed to take advantage of the prior knowledge about the characteristics of some of the interferences present in the acoustic field without reducing the beamformer’s degrees of freedom (DOFs). The appropriate amount of prior knowledge included in the design of the intra-ring weights is in the form of a fixed penalty factor value. The intra-ring weights are designed to cancel the interferences with prior knowledge. The inter-ring weights are adaptively obtained to cancel the unknown interferences.

The second contribution of this thesis lies on the optimization of the penalty factor that is automatically obtained to minimize the amount of residual error in the beamformer output at any time.

The third contribution of this thesis is the idea of combining the element space along with the beamspace beamforming, where the prior knowledge is added in form of beamspace beams pointing towards the interferences with known characteristics, meanwhile keeping the sub-arrays that use element space beamforming to handle the interferences with unknown characteristics. The combined beamspace element space (CBSES) is found to be robust against interference uncertainties and presents a consistent behavior for different scenarios.

The fourth contribution extends the CRA element space partial adaptive beamformer to the SA. We analyze several sensor arrangements and we suggest two novel sensor arrangements for beamforming with the SA that uses parallel ring sub-arrays. The partial adaptive beamformer design achieves huge computational savings, faster convergence and similar performance than that of the fully adaptive beamformer. Finally, the design of a broadband beamformer using nesting on an SA is also presented. Array nesting will increase the frequency range of the array and will reuse elements from different nests. Thus, reducing the total number of sensor elements needed.

The last contribution of this thesis is the implementation of two robust algorithms
against the SA sensor misplacement. The proposed algorithms use a better distortionless constraint that includes the sensor position errors, contrary to the constraint used in the diagonal loading robust method, which does not include the errors.
Chapter 1

Introduction

This chapter starts with a brief story in order to catch the reader attention about array processing. The next section discusses the problem of bird strikes and some of the solutions used. The general motivation of this thesis is laid out. Then we present an abbreviated background and previous works on array signal processing and beamforming, the reader is referred to Chapter 2 for more details and mathematical expressions. The chapter follows with the motivation for the current research. Then, the thesis contribution is laid out followed along the thesis outline. This chapter ends with a table of the notations and acronyms used throughout the text and a brief summary.

1.1 A not so Imaginary Tale

\(^1\)Since the beginning of the time, living organisms needed to interact with their surrounding to thrive and avoid extinction. They evolved sensors able to capture the events around them, like eyes that detect the visible electromagnetic spectrum, able to discriminate between different colors, and also able to determine the localizations of the perceived objects. They also developed ears that could sense acoustic pressure and distinguish between different pitches and localizations as well. Some species like the humans have developed complex sensors scattered across their bodies beside the eyes and ears.

\(^1\)The title of this section is inspired by the book of P. Nahin, ”An imaginary tale, the story of \(\sqrt{-1}\)”
For example, the human skin is able to send messages to the brain so we can locate precisely the part of the body that is subjected to heat, cold, pressure, or even a mosquito bite. Other species developed complex arrays of sensors like the compound eyes of the insects.

The time came when humans could leave the trees, stand up, and started using their hands as complex helping tools. They developed their brains and began observing the surrounding events with a growing critical and abstract criterion. Civilization sprung and we used our hands and intelligence to create tools that helped us to carry on and develop as a race, superior over all other living forms in this planet. One of the keys of our development was the ability to observe, think, understand, and replicate the phenomena that surround us.

With the development of electricity and transducers that convert electromagnetic, acoustic, etc. into electrical signals, we opened a gate that leaded us to unprecedented advances in almost all fields of knowledge. We could emulate biological sensors like eyes or ears, and enhance them to widen the limitation of their biological counterparts.

We did not stop there. Analog to how the living creatures have several sensors placed on different positions of their bodies so they better discriminate the source of visual objects, sounds, etc, the engineers also placed several transducers in different arrangements. There was no limit to add more visual or acoustic sensors than the usual pair used by most of the species. By adding several sensors in a predetermined geometry and processing their electrical signals to our advantage, we entered in the area of array signal processing.

1.2 Bird Strikes and Security Measures

A bird strike or bird aircraft strike hazard (BASH) is a collision between an bird and an aircraft. It is a common threat to aircraft safety and has caused a number of fatal
accidents [1]. Fatalities for civil aircraft are quite low and it has been estimated that there has been only 1 fatal accident to a jetliner in one thousand million ($10^9$) flying hours [2]. The majority of bird strikes (65%) cause little damage to the aircraft [3]. Most accidents occur when the bird is ingested into the engines or hits the aircraft windscreen. The estimated annual damages are of $400 million [1] in the United States of America alone to $1.2 billion worldwide to commercial aircraft [4]. Some of the latest bird strikes are next.

On January 15, 2009, US Airways Flight 1549 from LaGuardia Airport to Charlotte/Douglas International Airport ditched into the Hudson River after experiencing a loss of both turbines. It is suspected that the engine failure was caused by running into a flock of geese at an altitude of about 975 m (3,200 feet), shortly after takeoff. All 150 passengers and 5 crew members were safely evacuated after a successful water landing.

On January 4, 2009, a bird strike is suspected in the crash of a PHI S-76 helicopter in Louisiana. While the final report has not been published, early reports point to a bird impacting the windscreen and retarding the throttles, leading to the death of 7 of the 8 persons on board.

On November 10, 2008, a Ryanair flight FR4102 Boeing 737 from Frankfurt to Rome made an emergency landing at Ciampino Airport after multiple bird strikes put both engines out of commission. Three passengers and two crew members were injured, none seriously.

On April 29, 2007, a Thomsonfly Boeing 757 from Manchester Airport, UK to Lanzarote Airport, Spain suffered a bird strike when at least one bird was ingested by the starboard engine. The plane landed safely back at Manchester Airport a while later. The incident was captured by a plane spotter, as well as the emergency call picked up by a plane spotter’s radio. The video was later published and can be seen in Youtube.

On September 3, 2007, Virgin America Flight 837 performed an emergency landing
at San Francisco International Airport due to a bird strike.

In the summer of 2007, Delta Air Lines suffered an incident in Rome, Italy, as one of its Boeing 767 aircraft, on takeoff, ingested yellow legged gulls into both engines. Although the aircraft returned to Rome safely, both engines were damaged and had to be changed.

United Air Lines suffered a twin engine bird ingestion by a Boeing 767 on departure from Chicago’s O’Hare Field in the spring of 2007. One engine caught fire and bird remains were found in the other engine.

Even the spacecrafts are not safe from strikes. On July 26, 2005 the Space Shuttle Discovery hit a vulture during the take-off of STS-114. The collision occurred early during take off and at low speeds, with no obvious damage to the shuttle.

NASA also lost an astronaut, Theodore Freeman, to a bird strike. He was killed when a goose shattered the plexiglass cockpit of his T-38 Talon, resulting in fragments being ingested by the engines, leading to a fatal crash.

On 22 September 1995, a U.S. Air Force E-3 Sentry AWACS aircraft, crashed shortly after take off from Elmendorf AFB, AK. The plane lost power to both port side engines after these engines ingested several Canada Geese during takeoff. The aircraft went down in a heavily wooded area about two miles northeast of the runway, killing all 24 crew members on board.

The greatest loss of life directly linked to a bird strike was on October 4, 1960, when Eastern Air Lines Flight 375, a Lockheed L-188 Electra flying from Boston, flew through a flock of common starlings during take off, damaging all four engines. The plane crashed shortly after take-off into Boston harbor, with 62 fatalities out of 72 passengers. Subsequently, minimum bird ingestion standards for jet engines were developed by the FAA.

To reduce the danger of bird strikes the scientific community proposed different bird
detection methods [5–8] where some of the methods include detection, beamforming, and classification of acoustic signals using arrays of microphones. Among these components, beamforming is a very interesting field that combines array signal processing, adaptive algorithms, and filtering. This thesis is devoted to the research of new beamforming methods to help the scientific community in the solution of this problem.

1.3 Security in Puerto Rico

Puerto Rico is one of the last territories of the USA. Its area occupies 3,515 square miles and a estimated population of around four million inhabitants (in 2007). Due to its strategic location in the Caribbean sea, the island is suffering of high rates of criminality from drug smuggling and poverty. In 2008 there was 807 assassinations, an increase of 11% with respect the previous year. The assaults in 2008 were of 5,467, or 138.3 per 100,000 inhabitants, and the number of aggravated assaults were 3,115. Just from January to March 2009, 220 assassinations have been perpetrated. Only in the weekend of April 10 2009, there has been 19 murders.

The department of Homeland Security Grant Information started the State Homeland Security Program (SHSP)\(^3\) that only in 2008 assigned a total Funding of $861,280,000. $6,170,000 in Puerto Rico only. Also, the Urban Area Security Initiative Allocations (UASI) assigned a program funding of $781,630,000. In the San Juan Area $2,032,500.

The deployment of video cameras in public areas is known to be a crime deterrent. In some cases the quantity of deployed cameras is large and all of them have to be monitored at the same time by the guard on duty, by watching a wall filled with several video screens. Sometimes it is hard to know where the action is happening because the large quantity of screens and because there is not acoustic information from the

---

\(^2\)source: Puerto Rico Police Dept.
\(^3\)source: http://www.dhs.gov/xlibrary/assets/grant-program-overview-fy2008.pdf
recordings.

Acoustic arrays of sensors arranged in circular or spherical arrays could help the detection of where the action is coming from and guide the attention of the guard to the correct video screen. Fast detection and beamforming methods are necessary because the need of an early response. Classification algorithms would be needed so they focus on acoustic events related with crimes, like gunshots, people shouts, etc., and reject other acoustic events like, exploding tire, tire screeches, etc.

The current work about array processing and beamforming has, as a final objective, the design and implementation of these kind of systems that would help in the prevention of crimes in the beautiful island of Puerto Rico.

1.4 Background and Previous Works on Array Signal Processing and Beamforming

![Figure 1.1: Arbitrary array inside a field composed of the signal of interest (SOI), 3 interferences, and the background noise.](image)

Inside the field of electrical engineering, array signal processing is an area of signal
processing that deals with the processing of signals acquired from several sensors organized in a particular geometric arrangement, or array. Among the applications of array signal processing we find radar, sonar, seismology, biomedicine, communications, etc [9]. The objective is to detect, locate, and estimate the characteristics of a signal referred to as the signal of interest (SOI) that is contaminated with noise and interferences as shown in Fig. 1.1. The interferences are signals that arrive at different directions, or direction of arrivals (DOAs) other than that of the SOI. The noise is considered to be Gaussian and spatially uncorrelated from sensor to sensor. The noise can be originated inside each sensor like thermal noise or outside, as an isotropic uncorrelated noise. The sensor elements are omnidirectional and have identical characteristics.

1.4.1 Signal Field, Near and Far Field

The signals that impinge on the array can be originated in the near field or in the far field. Far field signals are also termed plane waves since each frequency component travel in planes of equal phase, or wavefronts [9]. Near field signals present a wavefront curvature depending on the distance of the sources to the array. Figure 1.2 show the field wavefronts shape for near and far field assumptions. This work assumes the far field assumption.

Because the nature of signal propagation in wavefronts, each sensor will receive the
same signal with a different time delay depending on its relative position with respect to the wavefront. This phenomenon will allow array signal processing to perform spatial filtering besides the well known frequency filtering, which can be done with just a single sensor. Spatial filtering allows us to discriminate a signal in space; that is, we can favor a signal that arrives at a particular DOA and attenuate other signals that arrive from other DOAs, even with the same frequency content. Also, the spatially uncorrelated noise power could be attenuated by the use of several sensors by adding signals coherently and noise incoherently. For a detailed explanation on the signal field mathematical models and spatial filtering, please refer to Chapter 2.

1.4.2 Array Geometry

![Uniform Linear and Ring Arrays](image)

Figure 1.3: Uniform Linear and Ring Arrays

The geometry of the array is of paramount importance in the ability of the array to detect, locate, and estimate the characteristics of signals in different scenarios. Therefore, for different applications, different geometries will yield better results. The first step in array signal processing is the selection of the array geometry. Literature sources classify the geometry into linear arrays, planar arrays and volumetric arrays [9].
A) The Uniform Linear Array

Among the different array geometries, the simplest, and one of the most used and studied is the linear array. The linear array arranges its elements on an imaginary straight line where all elements lie at a certain distance from each other. If the inter element distance is kept constant, we obtain the uniform linear array (ULA), shown in Fig. 1.3(a). Because the uniform linear arrangement of the elements, the ULA maintains a constant delay between adjacent elements for plane waves. This allows the use of techniques used in the design of FIR filters, like minimax, spectral weighting methods, equiripple tapering, etc. In spite of its advantages, the ULA suffers the problem of a DOA cone ambiguity, as shown in Fig. 1.3(a), where different signals impinging the array along the cone will be interpreted as the same [10, 11]. This limits the operation region of the ULA to 180° of azimuth coverage on a 2-D plane that encloses the ULA, and it is not suitable for 3-D beamforming. More details about the ULA, array response and mathematical expressions will be given in Chapter 2.

B) The Ring Array and the Concentric Ring Array

Within the planar arrays, the uniform ring array is preferable for 3-D beamforming because it provides 360° azimuth coverage, reduces the cone of uncertainty to just two DOAs, yields to a 180° elevation coverage, and maintains azimuthally uniform beam-pattern. An example of a ring array is shown in Fig. 1.3(b). The concentric ring array (CRA) composed of several ring arrays, has more flexibility in adaptive beamforming than the single ring array, is suitable for nested array design, and achieves frequency invariant characteristics [12–17].
C) The Spherical Array

For volumetric arrays, the spherical array (SA) is the preferred one because it has all the advantages of the ring array plus maintaining an identical beampattern both in azimuth and elevation, and eliminating the DOA uncertainty. [18–22].

This thesis focuses on the ring array, the CRA, and the SA. Please refer to Chapter 2 for details on the ring and spherical arrays.

1.4.3 Sensor Coefficients and Beamforming

After the array geometry, another important aspect is the selection of the processing that will be applied on the sensors. The processing is referred to as beamforming and the processor is called the beamformer. There are narrowband beamformers and broadband beamformers depending on the frequency characteristics of the signals.

A) Narrowband beamforming

For narrowband signals, where a time delay is equivalent to a phase shift, the sensors will receive the SOI and interferences with a difference of phase depending on the array
geometry. The processing applied to each sensor could be as simple as a multiplying coefficient or weight as it is shown in Fig. 1.4. The simplest weight selection is the one that enhances the SOI by aligning the phase shift in all sensors, and then adding the weighted signals to find the beamformer output. This choice of weights is termed delay-and-sum [23] and the beamformer is referred to as the spatial matched filter [10] or conventional beamformer [9]. This beamformer is optimum for a scenario where only the SOI and the isotropic noise is present, in the sense that it maximizes the output signal-to-noise ratio ($SNR_0$) [10]. Other processing methods use the statistical correlation of the data to design the weights. These methods adaptively reduce the interference signal levels to that of the isotropic noise, which is the same as to maximize the output signal-to-interference-plus-noise ratio ($SINR_0$). The beamformers that use these methods are referred to as the optimum fully adaptive beamformers. The term fully adaptive is given because the beamformers use adaptation on all sensor weights. Among the several methods we can cite the unit gain on interference-plus-noise of Robey et al. [24] and the minimum variance distortionless response (MVDR) or Capon beamformer [25]. More details about the conventional and the MVDR beamformers will be given in the second chapter.

B) Broadband beamforming

Broadband beamforming is used when the SOI has a certain frequency bandwidth where a time delay can not be modeled as a phase shift. Broadband beamforming can be applied either in the time domain or in the frequency domain [26].

The broadband time domain beamformer utilizes a weighted tapped delay line (WTDD) or finite impulse response (FIR) filter at the output of each sensor to achieve a frequency-dependent weighting that will allow the beamformer to enhance all frequency components.
Figure 1.5: Time domain broadband beamformer using FIR

of the SOI meanwhile attenuating the interferences and the background noise [27–30]. Figure 1.5(a) shows a time domain beamformer composed of $K$ FIR filter blocks that are applied to each array sensor. Figure 1.5(b) shows in detail an FIR filter composed of $m$ WTDDs.

Figure 1.6: Frequency domain broadband beamformer using DFT.

The broadband frequency domain beamformer uses a combination of narrowband beamformers applied to each frequency components of the array input signal. Fig. 1.6 shows a frequency domain beamformer that reads a block of array data samples or snapshots, decomposes the signal bandwidth into several frequency bins by the discrete
Fourier transform (DFT), implements narrowband beamforming at each of the frequency bins, and obtains the output block of snapshots by the inverse DFT (IDFT) \([9]\). We assume that the snapshot block size is large enough so adjacent blocks are statistically independent, also having independence between adjacent frequency bins. This thesis uses the frequency domain beamformer for processing broadband signals. The details of the DFT transform are shown in the Appendix A.

### 1.4.4 Element Space and Beamspace Beamformers

![Diagram](image)

(a) Element space beamformer  
(b) Beamspace beamformer

Figure 1.7: Element space and beamspace beamformers

In the previous subsection the processors have been applied individually to each sensor. This technique is referred to as element space beamforming. There is another technique called beamspace \([9, 31–36]\) where the array snapshots are processed jointly to form a set of beams that span the space of interest. For a \(K\) elements array, the array signals are first pre-processed by a linear transformation from dimension \(K\) to dimension \(M\) as shown in Fig. 1.7(b), then the transformed outputs are processed and combined to obtain the beamformer output. Beamspace is effective when the DOAs of the interferences are known. The beamformer can form beams steered toward the interference DOAs, and cancel the interferences using adaptive methods that will operate
on a lower dimension space vector formed from the beams’ outputs. In those cases
the beamspace beamformer will achieve a $\text{SINR}_0$ similar to that of the fully adaptive
beamformer [37, 38]. However, when the DOAs of all the interferences are not known,
it is necessary to have enough number of beams to cover all possible directions. One
solution is to create orthogonal beams as a spatial DFT that spans all space; however,
this solution usually involves as much complexity as the element space beamforming.

An interesting different type of beamspace transformation used in circular arrays is
the Davies’ transformation [37–39] which applies a linear transformation similar to an
IDFT followed by a compensation scaling on each IDFT component. This transformation
creates harmonic spatial beams referred to as spatial phase modes that correspond to
each of the IDFT components. The beauty of this method is that it will create an array
response similar to that of the ULA, and amenable for the use of beamforming techniques
that are exclusive of the ULA and FIR filter design techniques. This beamformer is also
called the virtual array (VA) processing because it creates a virtual ULA. The limitation
of the Davies transformation is that the beamforming is restricted to a SOI whose DOA
lies in the plane of the ring array. In other words, the beamformer is not able to steer
in elevation. Another disadvantage is that the isotropic noise at the array input is also
scaled, yielding a non-uniform noise power at each mode. For some modes, the noise is
hugely amplified worsening the beamformer performance. Di Claudio [13], Ju-lan and
Zi-shu [40] on separate investigations found a way to reduce this noise warping using an
array composed of two concentric rings.

Other interesting methods include the one from Chan and Chen [17] where a phase
mode transformation in the form of a spatial IDFT is applied on the CRA, along with
a compensation bank of FIR filters to achieve frequency invariant properties. The com-
pensation filters will remove the frequency dependence at each mode. Finally the com-
pensated modes are combined with a set of element space beamformer weights. They
also claim to successfully extending this method to the SA [18].

1.4.5 Partial Adaptive Beamforming

The number of adaptive coefficients required to achieve a desired performance could be huge [9]. The requirements to avoid spatial aliasing, achieve high SNR, maintain the amount of FFT bins or FIR filter length to achieve certain frequency and spatial resolution for broad band applications impose constraints with respect the minimum number of adaptive weights. A large number of coefficients increases the computational complexity and reduces the tracking ability in non-stationary environments. Partially adaptive beamformers are widely used to solve this problem [9, 41–44]; they use algorithms that reduce the dimensional space before adaptation. The cost is that partially adaptive beamformers reduce the array number of degrees of freedom (DOFs), hence decreasing the number of interferences that can be canceled with respect the fully adaptive array, and frequently suffer of lower SINR₀.

A) Eigenvector Beamformers

Among the partially adaptive beamformers, the eigenvector beamformers [9] compute the eigenvalues and eigenvectors of the correlation data matrix. In some of the eigenvector beamformers, it is sufficient to obtain the eigenvectors corresponding to the largest eigenvalues that correspond to the spatial correlated interferences and SOI with the largest power. Therefore we can use efficient computation techniques [45] and avoid doing a complete eigendecomposition of the correlation matrix. Some methods project the input onto a reduced rank subspace called the eigenspace that contains the signal and interferences. The principal components beamformer and the cross-spectral eigenspace beamformers are widely used [9]. The latter is more sensitive to SOI mismatch. These methods need to obtain the exact number of interferences present in the field to prop-
erly operate. If the number of interferences is underestimated, the performance will
be degraded since the SOI usually corresponds to a discarded eigenvector in low \( SNR \)
scenarios. Another method is the dominant-mode rejection beamformer [46], which is
preferred for low \( SNR \) scenarios.

**B) Beamspace Partial Beamformers**

Another kind of partially adaptive beamformers are the beamspace beamformers. As we
previously said, the beamspace beamformers create beams toward the interferences and
adaptively combine the beams’ outputs to cancel the interferences. Usually the number
of beams needed is much less than the number of array elements, therefore the number
of adaptive coefficients is much less than those of the fully adaptive beamformer. As we
previously explained, there should be a good estimation of the interferences DOAs for
the beamspace beamformer to operate efficiently.

**C) Element Space Partial Beamformers**

There is a third kind of partially adaptive beamformers that decompose the whole array
into several sub-arrays that are suitable for fixed element space beamforming; then,
combine their outputs with adaptive weights to obtain the beamformer output. Some
early works belong to Abraham [47] and Cox [48, 49] that use a combination of linear
hydrophone arrays. Other articles consider a similar approach using the CRA [50, 51].
This thesis will focus on this type of beamformers as a starting point.

**1.5 Motivation**

The element space partial adaptive beamformer that applies sub-arrays to perform fixed
beamforming, and combines their outputs with adaptive weights is very attractive. First,
it is not necessary to perform the eigendecomposition operations needed by the principal
components and dominant-mode rejection beamformers. Secondly, the reduced space
created by the sub-arrays do not favor any spatial region contrary to the beamspace
beamformers. Therefore, as long as there are enough DOFs to cancel the interferences
present in the acoustic field, the beamformer could efficiently operate with the proper
limitations of an partially adaptive array about $SINR_0$.

In some applications where the objective is to detect and determine the location
of particular signals, like airport bird monitoring, security systems on parkings, malls,
schools playgrounds, exercise yards in penitentiaries, etc., there are interference sources
that are stationary and we could obtain a priori information about their characteristics.
Usually there are devices that create permanent and localized acoustic interferences like
loudspeakers, external air conditioner units, etc. We could use the available prior inform-
ation to design better partially adaptive beamformers that do not sacrifice the DOFs
to cancel those interferences. The solution is to use fixed beamforming that includes
the prior information in the design of the sub-arrays weights so those interferences are
eliminated at the sub-array level, and leaving the adaptive weights to cancel the remain
interferences. The partial adaptive methods used in the CRA could be easily extended to
the SA. We will study the feasibility and performance investigation for partial adaptive
beamforming using the SA.

There are extensively scientific articles about the robustness of fully adaptive beam-
formers with respect uncertainties in the sensor locations, DOAs, etc. [52–65]. They
usually include a regularization in the form of a diagonal loading of the spatial corre-
lation matrix. For partially adaptive beamformers, the diagonal loading is performed
on the reduced rank correlation matrix. It would be interesting to investigate alterna-
tive methods that enhance the robustness of the beamformer, different than the usual
diagonal loading techniques.

For acoustic applications, the beamforming is always performed under broadband
assumptions. There are several methods to achieve frequency invariance that use the fully
adaptive beamformer. For partially adaptive beamformers using the SA, the research has been focused on phase mode beamformers and recently on element space partially non-adaptive beamformers [14, 15]. We propose alternative solutions using array nesting for element space partially adaptive broadband beamformers.

1.5.1 Assumptions and Limitations

The research work described in this thesis made the following assumptions unless otherwise noted.

1. The propagation speed is constant for all frequencies.

2. The SOI and interferences originate in the far field.

3. The background noise is Gaussian and statistically independent from sensor to sensor and from snapshot to snapshot.

4. The interferences are uncorrelated with respect the SOI.

5. The SOI and interferences are not spatially spread, i.e., each one arrives from a particular and unique DOA.

6. Broadband beamforming is applied in the frequency domain assuming independence between snapshot blocks.

7. The sensors are omnidirectional and with the same gain.

8. There are no uncertainty in the sensors locations except in the cases where we study the robustness of the beamformer against uncertainties in the location of the sensors.

9. There is no uncertainty in the DOA of the SOI.
10. There is prior knowledge about the characteristics of some interferences.

11. For the cases where the interferences characteristics were obtained by prior knowledge, there is no uncertainty in the interferences characteristics unless explicitly noted.

1.6 Thesis Contribution

1.6.1 Partial Adaptive Concentric Ring Array

This thesis introduces three new methods for the beamforming of the CRA for acoustic signals in 3-D, which take advantage of the available prior knowledge about the characteristics of some of the interferences present in the sound field.

The first method uses the element space partial adaptive beamforming technique to find the beamformer weights. The CRA is decomposed into rings that perform individual beamforming using intra-ring weights. The design of the intra-ring weights includes a penalty factor that controls the amount of available prior knowledge, which is used to cancel those interferences at the ring level. The ring outputs are combined together with adaptive inter-ring weights to form the overall beamformer. The inter-ring weights adaptively change to cancel the interferences with unknown characteristics. Initially we used a fixed penalty factor value that in most scenarios performed better than the previous method from Li et al. [50, 51], which does not use the prior knowledge in the design of the weights.

The second method introduces a more advanced alternative that automatically obtains the appropriate penalty factor value so it minimizes the amount of residual interference and noise power at the beamformer output. The result is a beamformer that always maintains the minimum attainable residual interference and noise output power level at all scenarios.
The third method combines the element space along with the beamspace beamforming techniques. The prior knowledge is added in form of beamspace beams pointing towards the interferences with known characteristics. The beamspace part targets the interferences with prior knowledge and the element space part is used to handle the interferences with unknown characteristics. The obtained beamformer presents a consistent behavior in keeping a low residual interference and noise output power level for different scenarios. Also, it is robust against uncertainties in the DOA of the interferences obtained by prior knowledge, contrary to the previous element space partial adaptive methods.

1.6.2 Partial Adaptive Spherical Array

We extend the CRA partial adaptive narrowband beamformer to the SA that uses ring sub-arrays. The proposed beamformer reduces the complexity of adapting a large number of coefficients and at the same time increases the convergence speed during adaptation. The reduction in performance with respect the fully adaptive beamformer is almost negligible. We suggest two sensor arrangements methods for SA that position the element sensors in parallel rings and at the same time maintain a minimum number of elements by maximizing the inter-element distance and still avoid spatial aliasing. We compare the proposed arrangements with the ones used in earlier publications. We also propose a frequency domain broadband partial adaptive beamformer that expands the beamformer frequency range by applying array nesting.

1.6.3 Beamformer Robustness Against Sensor Position Errors

Finally this thesis propose two robust methods for beamforming using the CRA and the SA. The methods use constraints that are statistically related to the sensor position errors, contrary to the classical robust method based on diagonal loading.
1.7 Thesis Outline

The rest of this dissertation is organized as follows. Chapter 2 presents the fundamentals of beamforming in more depth than those described in this chapter. Chapter 3 corresponds to a published article titled ”An Improved Partial Adaptive Narrow-band Beamformer Using CRA” where we use the prior information in the design of the beamformer weights prior to partial adaptation. Chapter 4 corresponds to a published article titled ”Optimizing The Performance of the Partial Adaptive CRA in The Presence of Prior Knowledge” where the penalty factor that controls the amount of prior information in the design of the beamformer weights is optimized to render better performance. Chapter 5 is the ”Combined Beamspace and Element Space Technique for Partial Adaptive CRA” that introduces a novel technique that combines the element space and the beamspace methods for partial adaptive beamformers. Chapter 6 study partial adaptive beamforming using the SA. First, we suggest two sensor arrangements methods that will be used for partial adaptive element space beamforming. Then, we introduce a narrowband beamformer that uses the new sensor arrangements. We also introduce a frequency domain broadband beamformer that employs a bank of the proposed narrowband beamformers. We increase the frequency range of the broadband beamformer by using array nesting. Chapter 7 introduces two new robust beamforming methods against sensor position errors. The proposed methods use a improved constraint that takes into account the variance of the sensor position errors. Chapter 8 summarizes this dissertation and propose future research topics.

1.8 Notation and Acronyms

Table 1.1 shows acronyms used in this thesis. Tables 1.2 - 1.4 show the notation used in this thesis.
<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-D</td>
<td>Two-Dimensional.</td>
</tr>
<tr>
<td>3-D</td>
<td>Three-Dimensional.</td>
</tr>
<tr>
<td>$A_g$</td>
<td>Array gain against interferences and noise.</td>
</tr>
<tr>
<td>$A_w$</td>
<td>Array gain against uncorrelated noise.</td>
</tr>
<tr>
<td>BASH</td>
<td>Bird Aircraft Strike Hazard.</td>
</tr>
<tr>
<td>BW$_{NN}$</td>
<td>Null to null bandwidth.</td>
</tr>
<tr>
<td>CBSES</td>
<td>Combined Beamspace Element Space.</td>
</tr>
<tr>
<td>CRA</td>
<td>Concentric Ring Array.</td>
</tr>
<tr>
<td>$DI$</td>
<td>Directivity Index.</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform.</td>
</tr>
<tr>
<td>DL</td>
<td>Diagonal Loading.</td>
</tr>
<tr>
<td>DOA</td>
<td>Direction of Arrival.</td>
</tr>
<tr>
<td>DOF</td>
<td>Degrees of Freedom.</td>
</tr>
<tr>
<td>EMSE</td>
<td>Excess Mean Square Error.</td>
</tr>
<tr>
<td>FA</td>
<td>Fully Adaptive.</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response.</td>
</tr>
<tr>
<td>GSC</td>
<td>Generalized Sidelobe Canceler.</td>
</tr>
<tr>
<td>HPBW</td>
<td>Half Power Beamwidth.</td>
</tr>
<tr>
<td>IDFT</td>
<td>Inverse Discrete Fourier Transform.</td>
</tr>
<tr>
<td>INR</td>
<td>Interference to Noise Ratio.</td>
</tr>
<tr>
<td>LCMV</td>
<td>Linearly Constrained Minimum Variance.</td>
</tr>
<tr>
<td>LMS</td>
<td>Least Mean Squares.</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares.</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error.</td>
</tr>
<tr>
<td>MVDR</td>
<td>Minimum Variance Distortionless Response.</td>
</tr>
<tr>
<td>NB</td>
<td>Narrowband Beamformer.</td>
</tr>
<tr>
<td>NLMS</td>
<td>Normalized Least Mean Square.</td>
</tr>
<tr>
<td>RLS</td>
<td>Recursive Least Squares.</td>
</tr>
<tr>
<td>SAGE</td>
<td>Space Alternating Generalized EM algorithm.</td>
</tr>
<tr>
<td>$SINR_i$</td>
<td>Input Signal to Interference plus Noise Ratio.</td>
</tr>
<tr>
<td>$SINR_0$</td>
<td>Beamformer Output Signal to Interference plus Noise Ratio.</td>
</tr>
<tr>
<td>SIR</td>
<td>Signal to Interference Ratio.</td>
</tr>
<tr>
<td>SLA</td>
<td>Standard Linear Array.</td>
</tr>
<tr>
<td>SMI</td>
<td>Sample Matrix Inversion.</td>
</tr>
<tr>
<td>$SNR_i$</td>
<td>Input Signal to Noise Ratio.</td>
</tr>
<tr>
<td>$SNR_0$</td>
<td>Beamformer Output Signal to Noise Ratio.</td>
</tr>
<tr>
<td>SOI</td>
<td>Signal of Interest.</td>
</tr>
<tr>
<td>ULA</td>
<td>Uniform Linear Array.</td>
</tr>
<tr>
<td>WTDD</td>
<td>Weighted Tapped Delay Line.</td>
</tr>
<tr>
<td>Notation</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Array Geometry</strong></td>
<td></td>
</tr>
<tr>
<td>$K$</td>
<td>Number of array elements.</td>
</tr>
<tr>
<td>$M$</td>
<td>Number of array rings.</td>
</tr>
<tr>
<td>$K_m$</td>
<td>Number of array elements in ring $m$.</td>
</tr>
<tr>
<td>$r$</td>
<td>Ring (or sphere) ratio.</td>
</tr>
<tr>
<td>$p_k$</td>
<td>Sensor position vector $k$.</td>
</tr>
<tr>
<td>$d$</td>
<td>Inter-element distance.</td>
</tr>
<tr>
<td>$(x_p, y_p, z_p)$</td>
<td>Cartesian coordinates.</td>
</tr>
<tr>
<td>$i, j, k$</td>
<td>Unit vectors in Cartesian coordinates..</td>
</tr>
<tr>
<td>$j$</td>
<td>Pure imaginary complex number $\sqrt{-1}$.</td>
</tr>
<tr>
<td><strong>Field parameters</strong></td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>Total number of interferences.</td>
</tr>
<tr>
<td>$C$</td>
<td>Number of interferences with acquired prior knowledge.</td>
</tr>
<tr>
<td>$s(t)$</td>
<td>Complex amplitude of the SOI.</td>
</tr>
<tr>
<td>$i_l(t)$</td>
<td>Complex amplitude of the $l^{th}$ interference.</td>
</tr>
<tr>
<td>$\sigma_s^2$</td>
<td>SOI power.</td>
</tr>
<tr>
<td>$\sigma_i^2$</td>
<td>$l^{th}$ interference power.</td>
</tr>
<tr>
<td>$\sigma_n^2$</td>
<td>Uncorrelated noise power.</td>
</tr>
<tr>
<td>$t$</td>
<td>Time snapshot.</td>
</tr>
<tr>
<td>$f$</td>
<td>Frequency in Hz.</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Angular frequency in rads/sec.</td>
</tr>
<tr>
<td>$c$</td>
<td>Propagation speed.</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Wavelength.</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>Wavenumber magnitude equal to $\omega/c$.</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Elevation or polar angle, measured from the $z$ axis.</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Azimuth angle, measured from the $x$ axis counterclockwise.</td>
</tr>
<tr>
<td>$\theta_0$</td>
<td>SOI elevation or polar angle, measured from the $z$ axis.</td>
</tr>
<tr>
<td>$\phi_0$</td>
<td>SOI azimuth angle, measured from the $x$ axis counterclockwise.</td>
</tr>
<tr>
<td>$\theta_l$</td>
<td>$l^{th}$ interference elevation or polar angle, measured from the $z$ axis.</td>
</tr>
<tr>
<td>$\phi_l$</td>
<td>$l^{th}$ interference azimuth angle, measured from the $x$ axis.</td>
</tr>
<tr>
<td>$a$</td>
<td>Direction cosines, direction of arrival.</td>
</tr>
<tr>
<td>$a_0$</td>
<td>SOI direction of arrival.</td>
</tr>
<tr>
<td>$a_i$</td>
<td>$l^{th}$ interference direction of arrival.</td>
</tr>
<tr>
<td><strong>Array Signals</strong></td>
<td></td>
</tr>
<tr>
<td>$u$</td>
<td>Array input signal vector.</td>
</tr>
<tr>
<td>$u_s$</td>
<td>Component of array input signal vector from the SOI only.</td>
</tr>
<tr>
<td>$u_i$</td>
<td>Component of array input signal vector from the interferences only.</td>
</tr>
<tr>
<td>$u_{il}$</td>
<td>Component of array input signal vector from the $l^{th}$ interference only.</td>
</tr>
<tr>
<td>$n$</td>
<td>Component of array input signal vector from the noise only.</td>
</tr>
<tr>
<td>$u_k, [u]_k$</td>
<td>Input signal at array sensor $k$ (element of vector $u$).</td>
</tr>
</tbody>
</table>
### Table 1.3: Notation (II)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>Nominal steering vector.</td>
</tr>
<tr>
<td>s₀</td>
<td>Nominal SOI steering vector.</td>
</tr>
<tr>
<td>iₗ</td>
<td>Nominal ℓth interference steering vector.</td>
</tr>
<tr>
<td>˜s₀</td>
<td>True SOI steering vector.</td>
</tr>
<tr>
<td>˜iₗ</td>
<td>True ℓth interference steering vector.</td>
</tr>
</tbody>
</table>

**Beamforming parameters**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rs</td>
<td>SOI nominal spatial correlation matrix.</td>
</tr>
<tr>
<td>Ri</td>
<td>Interferences nominal spatial correlation matrix.</td>
</tr>
<tr>
<td>Riₗ</td>
<td>ℓth interference nominal spatial correlation matrix.</td>
</tr>
<tr>
<td>Rn</td>
<td>Uncorrelated noise spatial correlation matrix (= σ²ᵈI).</td>
</tr>
<tr>
<td>Rin</td>
<td>Interferences plus noise nominal spatial correlation matrix.</td>
</tr>
<tr>
<td>Ru</td>
<td>Nominal spatial correlation matrix (= Rs + Ri + Rn).</td>
</tr>
<tr>
<td>˜Rs</td>
<td>True SOI spatial correlation matrix, under location errors.</td>
</tr>
<tr>
<td>˜Riₗ</td>
<td>True ℓth interference spatial correlation matrix, under location errors.</td>
</tr>
<tr>
<td>˜Rin</td>
<td>True interference plus noise spatial correlation matrix, under location errors.</td>
</tr>
<tr>
<td>ε</td>
<td>Expectation of the exponential misplacement (= E[e⁻κ²σ²ᵈ]).</td>
</tr>
<tr>
<td>σ²_DL</td>
<td>Diagonal loading power.</td>
</tr>
<tr>
<td>jₒ</td>
<td>Spherical Bessel function.</td>
</tr>
<tr>
<td>Cs</td>
<td>Data SOI spatial correlation matrix.</td>
</tr>
<tr>
<td>Cᵢₙ</td>
<td>Data interference spatial correlation matrix.</td>
</tr>
<tr>
<td>Cu</td>
<td>Data spatial correlation matrix.</td>
</tr>
<tr>
<td>RₘₐₙC</td>
<td>Spatial correlation matrix from interferences with prior knowledge.</td>
</tr>
<tr>
<td>P</td>
<td>Element space partition matrix.</td>
</tr>
<tr>
<td>Pb</td>
<td>Combined beamspace element space partition matrix.</td>
</tr>
<tr>
<td>B</td>
<td>GSC blocking matrix.</td>
</tr>
<tr>
<td>y</td>
<td>Reduced dimension array signal (= PHu).</td>
</tr>
<tr>
<td>˜hₘ</td>
<td>Sparse Inter ring m weight vector (delay-and-sum) (K x 1).</td>
</tr>
<tr>
<td>˜hₘ</td>
<td>Inter ring m weight vector (delay-and-sum) (Kₘ x 1).</td>
</tr>
<tr>
<td>˜gₘ</td>
<td>Improved inter ring m weight vector (contains prior information).</td>
</tr>
<tr>
<td>bₙ</td>
<td>CBSES beamspace vector (contains prior information).</td>
</tr>
<tr>
<td>v</td>
<td>Array weight vector.</td>
</tr>
<tr>
<td>˜vₗ</td>
<td>Desired quiescent weight vector.</td>
</tr>
<tr>
<td>wₗ</td>
<td>GSC Quiescent weight vector.</td>
</tr>
<tr>
<td>wₛ</td>
<td>GSC low branch adaptive weight vector.</td>
</tr>
<tr>
<td>wₜₘₜ</td>
<td>GSC low branch steady-state weight vector.</td>
</tr>
<tr>
<td>wₙₛₛ</td>
<td>GSC weight vector (= wₚₗ - Bwₚₛ).</td>
</tr>
</tbody>
</table>

---

aTo avoid confusion, in the section where jₒ is used, i will be used as the imaginary quantity √−I.
### Table 1.4: Notation (III)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Number of snapshots.</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of frequency bins.</td>
</tr>
<tr>
<td>$O$</td>
<td>Spherical harmonics order number.</td>
</tr>
<tr>
<td>$Q$</td>
<td>Number of constraints.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Penalty factor.</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Log of the penalty factor ($= \log \alpha$).</td>
</tr>
<tr>
<td>$\nabla_\beta P$</td>
<td>Gradient of function $P$ with respect $\beta$.</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Lagrange multiplier.</td>
</tr>
<tr>
<td>$\rho$</td>
<td>RLS forgetting factor.</td>
</tr>
<tr>
<td>$z(t)$</td>
<td>Beamformer output signal.</td>
</tr>
<tr>
<td>$z_q(t)$</td>
<td>Beamformer quiescent response.</td>
</tr>
<tr>
<td>$z_a(t)$</td>
<td>Beamformer adaptive response.</td>
</tr>
<tr>
<td>$z_s(t)$</td>
<td>Beamformer output signal due to the SOI only.</td>
</tr>
<tr>
<td>$z_{i_l}(t)$</td>
<td>Beamformer output signal due to the $l^{th}$ interference only.</td>
</tr>
<tr>
<td>$z_n(t)$</td>
<td>Beamformer output signal due to the spatially uncorrelated noise only.</td>
</tr>
<tr>
<td>$P_z$</td>
<td>Beamformer output signal power.</td>
</tr>
<tr>
<td>$P_s$</td>
<td>Beamformer output signal power due to the SOI only.</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Beamformer output signal power due to the interferences only.</td>
</tr>
<tr>
<td>$P_n$</td>
<td>Beamformer output signal power due to the spatially uncorrelated noise only.</td>
</tr>
<tr>
<td>$P_{in}$</td>
<td>Beamformer output signal power due to the interferences and noise only.</td>
</tr>
</tbody>
</table>
1.9 Chapter Summary

The first chapter of this thesis is a brief introduction to beamforming with the objective to familiarize the reader with the beamforming background and the previous works. We show the previous research on bird aircraft strike hazards and the motivation to improve the current beamforming techniques. Then we start explaining the signal field modeling of near and far field. The array geometry, the processing that are applied to the sensor elements, different beamforming techniques of narrowband and broadband, element space and beamspace, as well as fully and partial adaptive beamforming techniques. The chapter includes abundant references about the previous works on these topics. The motivation of the thesis is laid out as well and the assumptions and limitations. The chapter follows with the thesis contributions. The chapter ends with the thesis outline and the notation and acronyms tables. The next chapter will review these topics with more details and with a mathematical treatment.
Chapter 2

Fundamentals of Beamforming

This chapter complements the topics discussed on Chapter 1 by covering them with more details needed to understand the following chapters, which assume the reader is familiarized with basic knowledge about array signal processing and beamforming. We start with the coordinate systems used for beamforming in 3-D; the cartesian and the spherical coordinates systems, the latter being used widely in array processing of circular and spherical arrays. The chapter continues with the solution to the wave equation for narrowband signals. The signal model is laid out, the definition of the array response, and the array geometry is explained. The chapter continues with the beamforming of narrowband signals for an acoustic field composed of SOI, interferences and background noise. The beampattern is introduced and some examples are shown. The beamformer performance measures follows, and the chapter ends with a brief summary. This chapter is mainly based on the works of D. H. Johnson & D. E. Dudgeon [11], H. L. Van Trees [9], D. G. Manolakis et al. [10], and J. Li & P. Stoica [66].

2.1 Coordinate Systems in Array Signal Processing

This section formulates the coordinate systems used for array signal processing and beamforming.
2.1.1 Cartesian Coordinates

The cartesian coordinate system also called the rectangular coordinate system determines uniquely a point in the 3-D plane with respect an orthonormal basis represented by the unit vectors \( \mathbf{i}, \mathbf{j}, \mathbf{k} \), which are oriented in the same direction as the \( x, y, z \) axes respectively.\(^1\) Each point \( p \) in space is represented by a three scalar components \((x_p, y_p, z_p)\), where each component \( x_p, y_p, \) and \( z_p \) equals the minimum Euclidean distance of the point \( p \) to the planes defined by \( yz, xz, \) and \( xy \) respectively. Fig. 2.1(a) shows the coordinate axes \( x, y, z \), the point \( p \) and its corresponding cartesian coordinates \((x_p, y_p, z_p)\). The vector that connects the origin of coordinates with the point \( p \) can be written as \( \mathbf{p} = x_p\mathbf{i} + y_p\mathbf{j} + z_p\mathbf{k} \) as it is shown in Fig. 2.1(a), or

\[
\mathbf{p} = \begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^T, \tag{2.1}
\]

where the operator \((\cdot)^T\) represents transpose.

---

\( ^1 \)Please do not confuse the vector \( \mathbf{j} \) with the imaginary scalar number \( j = \sqrt{-1} \).
2.1.2 Spherical Coordinates

The vector \( p \) forms an angle with respect the \( z \) axis referred to as the elevation, zenith, or polar angle \( \theta_p \). The projection of the vector \( p \) on to the \( xy \) plane forms an angle with respect the \( x \) axis referred to as the azimuth angle \( \phi_p \). The spherical coordinates are represented by the triad \((r_p, \theta_p, \phi_p)\), where \( r_p \) is the magnitude of \( p \) and the other two terms have already been defined. Fig. 2.1(b) shows the spherical coordinates for the point \( p \).

The relationship between the spherical and the cartesian coordinates for the point \( p \) is

\[
\begin{align*}
    x_p &= r_p \sin \theta_p \cos \phi_p \\
    y_p &= r_p \sin \theta_p \sin \phi_p \\
    z_p &= r_p \cos \theta_p.
\end{align*}
\]

Equivalently, the relationship between the cartesian and the spherical coordinates for the point \( p \) is

\[
\begin{align*}
    r_p &= \sqrt{x_p^2 + y_p^2 + z_p^2} \\
    \theta_p &= \arctan \left( \frac{\sqrt{x_p^2 + y_p^2}}{z_p} \right) \\
    \phi_p &= \arctan \left( \frac{y_p}{x_p} \right).
\end{align*}
\]

For 3-D beamforming of ring and spherical arrays, the spherical coordinates are the most adequate. Also, for beamforming on far field assumption, the distance from the array elements to the source is not as important as its polar and azimuth angles. In practice, the distance to the source is unknown and only the polar and azimuth angles are taken into account. The polar and azimuth angles that indicate the source of a signal is referred to as the DOA of the signal.

2.2 Propagating Waves

This section shows the wave equation and its solution for monochromatic or narrowband signals in an acoustic far field assumption. The array input signals are arranged in a
vector and the steering vector definition is introduced. The section ends with the idea of phase shift equivalent to a time delay.

2.2.1 The Wave Equation

A space-time signal can be written as $u(x_p, y_p, z_p, t)$, where $u$ represents a general scalar acoustic field, and $(x_p, y_p, z_p)$ the cartesian coordinates of the field amplitude in a generic point $p$. The time component has been added to the cartesian coordinates as a fourth variable of $u$ since the field could be time dependent. The information that is received by any array sensor element is carried by propagating waves from its source point. In acoustics, the wave equation has the following expression

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2},$$

(2.4)

where $c$ is the propagation speed of acoustic waves, or speed of sound and $\frac{\partial^2 u}{\partial x^2}$ means partial second derivative of the function $u(x, y, ...)$ with respect the variable $x$. This equation governs how the signal travels from a radiating source to an array of elements [11].

2.2.2 Solution to the Wave Equation

The solution to the wave equation for a monochromatic or narrowband signal of angular frequency $\omega$ on a generic point $p$ is

$$u(p, a, \omega, t) = A(t)e^{j(\omega t - \kappa^T p)},$$

(2.5)

where $p$ is the vector that represents a generic point in space as in (2.1), $A(t)$ is a slow varying function if compared to $\omega$, $j = \sqrt{-1}$, $t$ is the time, and

$$\kappa = \frac{\omega}{c} a$$

(2.6)

is the wavenumber defined for a plane wave with angular frequency $\omega$ propagating in a locally homogeneous medium with constant speed $c$ [9]. $a$ is a unit vector that indicates
the direction of the propagated signal. It is expressed as

\[
a = \begin{bmatrix}
-\sin \theta \cos \phi \\
-\sin \theta \sin \phi \\
-\cos \theta 
\end{bmatrix}.
\] (2.7)

The minus signs occur because the signal travels towards the origin of coordinates. \( \theta \), \( \phi \) are the polar and azimuth angle of the impinging signal that define the signal DOA. Please note that the angles come from the spherical coordinates system and that the term \( r_p \) is missing. Each component of (2.7) is also called the direction cosines with respect to each axis.

When we replace (2.7) in (2.5) we obtain the signal expression on a point defined by \( p \) as an explicit function of \( a \)

\[
u(p, a, \omega, t) = A(t)e^{j(\omega t)}e^{j(-\kappa a^T p)},
\] (2.8)

where \( \kappa = \omega/c \) is the wavenumber magnitude. As we can notice from (2.8), we can separate the time-frequency content from the wavenumber-spatial term. Therefore the signal read at each sensor in the array will keep the same frequency content but will have different spatial content because the sensors are arranged in different locations \( p \). If we

![Diagram](image-url)
consider an array of \( K \) elements located at positions \( p_1, \ldots, p_K \) the signal read for each sensor (2.8) can be arranged in an input signal vector \( u \) as

\[
\mathbf{u}(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_K, \mathbf{a}, \omega, t) = \begin{bmatrix}
    u(p_1, \mathbf{a}, \omega, t) \\
    u(p_2, \mathbf{a}, \omega, t) \\
    \vdots \\
    u(p_K, \mathbf{a}, \omega, t)
\end{bmatrix} = A(t)e^{j\omega t} \begin{bmatrix}
    e^{-j\kappa a^T p_1} \\
    e^{-j\kappa a^T p_2} \\
    \vdots \\
    e^{-j\kappa a^T p_K}
\end{bmatrix}, \quad (2.9)
\]

where \( A(t)e^{j\omega t} = s(t) \) is a common factor that can be extracted out of the vector. Fig. 2.2(a) shows an arbitrary array of \( K \) elements defined by their spherical coordinates inside a field composed of a single narrowband signal with a generic DOA of \((\theta, \phi)\). The field signal read by the same array elements is shown in Fig. 2.2(b).

### 2.2.3 The Array Steering Vector

The last term of (2.9) can be written as

\[
\mathbf{s} = \begin{bmatrix}
    e^{-j\kappa a^T p_1} \\
    e^{-j\kappa a^T p_2} \\
    \vdots \\
    e^{-j\kappa a^T p_K}
\end{bmatrix}, \quad (2.10)
\]

The vector \( \mathbf{s} \) is referred to as the steering vector, array manifold vector, or array response vector. The steering vector incorporates all the characteristics of the array and determines the response of the array to any signal with a particular DOA. Each element of \( \mathbf{s} \) is a simple phase shift. We also can appreciate that \( \mathbf{s} \) in (2.10) not only depends on the array elements positions \( \mathbf{p}_k \) and the DOA of the impinging signal through \( \mathbf{a} \) in (2.7), but also on the signal frequency and its propagation speed through \( \kappa \). Therefore the steering vector in (2.10) is not independent on the frequency of the signal.

### 2.2.4 Phase Shift, Delay

The different phase shift in the elements of the steering vector can also be seen as a different delay experienced by the signal at each sensor. The signal expression for the
$k^{th}$ element can be rearranged as

$$u(p_k, a, \omega, t) = A e^{j\omega(t - \frac{a^T p_k}{c})}. \quad (2.11)$$

The second term in the exponential represents a delay of $\tau_k = \frac{a^T p_k}{c}$. Therefore the signal reaches each array element with a different delay that only depends on the element position, the DOA of the signal, and the propagation speed. This difference of phase or delay suffered by the arriving signal on each sensor elements will be exploited to perform spatial filtering or beamforming.

### 2.3 Modeling the Signal Field

This section introduces the signal model of a narrowband acoustic far field that is composed of the SOI, interferences and background noise.

#### 2.3.1 Signal of Interest Model

The SOI is defined as a plane wave signal having a DOA of $(\theta_0, \phi_0)$. The signal read at the $k^{th}$ element of an arbitrary array composed of $K$ elements is

$$u_s(p_k, a_0, \omega, t) = s(t)e^{-j(\kappa a_0^T p_k)}, \quad (2.12)$$

where the subscript $(\cdot)_s$ stands for the SOI, $s(t)$ is the amplitude, and $a_0$ is (2.7) particularized to $(\theta_0, \phi_0)$.

The SOI array input vector is obtained from (2.9) when particularized to $s_0$

$$u_s(a_0, \omega, t) = s(t)s_0, \quad (2.13)$$

where the variables $p_1, p_2, \ldots, p_K$ in $u_s$ have been dropped out for readability. The SOI steering vector $s_0$ is (2.10) particularized to $a_0$

$$s_0 = \begin{bmatrix} e^{-j\kappa a_0^T p_1} \\ e^{-j\kappa a_0^T p_2} \\ \vdots \\ e^{-j\kappa a_0^T p_K} \end{bmatrix}. \quad (2.14)$$
The term $a_0$ contains the SOI DOA: $(\theta_0, \phi_0)$ as

$$a_0 = \begin{bmatrix}
-\sin \theta_0 \cos \phi_0 \\
-\sin \theta_0 \sin \phi_0 \\
-\cos \theta_0
\end{bmatrix}. \quad (2.15)$$

### 2.3.2 Interferences Model

Any signal arriving from a different DOA is considered an interference. Therefore we will model the interference with a similar expression than that of the SOI. For an interference arriving at a DOA $(\theta_l, \phi_l)$ the mathematical expression would be

$$u_{il}(a_l, \omega, t) = i_l(t) i_l, \quad (2.16)$$

where the subscript $(,)_{il}$ stands for the $l$th interference from a total of $L$, $i_l(t)$ is the amplitude of the interference signal, and the steering vector expression is defined as

$$i_l = \begin{bmatrix}
e^{-jka_l^T p_1} \\
e^{-jka_l^T p_2} \\
\vdots \\
e^{-jka_l^T p_K}
\end{bmatrix}. \quad (2.17)$$

Please notice that for the interference steering vector, we will be using the character\textsuperscript{2} $i$ instead of $s$. The term $a_l$ contains the interference DOA and it is equal to

$$a_l = \begin{bmatrix}
-\sin \theta_l \cos \phi_l \\
-\sin \theta_l \sin \phi_l \\
-\cos \theta_l
\end{bmatrix}. \quad (2.18)$$

### 2.3.3 Background Noise Model

The background noise is modeled as a time uncorrelated random process with zero mean and Gaussian distribution with variance $\sigma_n^2$. The noise is also uncorrelated from sensor to sensor. The noise vector is represented by $n(t)$.

\textsuperscript{2}From now on $i$ will refer to the interference steering vector and not to the unitary vector of the rectangular coordinates.
2.3.4 Beamformer Input Signal Model

The array input signal model is composed of the sum of the SOI, \( L \) interferences, and the background noise as defined in the previous subsections. Its mathematical expression is the sum of (2.13), (2.16), and \( n(t) \) as

\[
u(t) = s(t)s_0 + \sum_{l=1}^{L} i_l(t)i_l + n(t), \tag{2.19}\]

were we dropped the variables \( a_s, a_i, \) and \( \omega \) from \( u \) for readability. From the \( L \) interferences we assume to have prior knowledge of \( C \) where \( C \leq L \). The prior knowledge for the interference \( l \) means that we know its DOA \((\theta_{i_l}, \phi_{i_l})\) and power.

2.4 Array Geometry

This section begins discussing the minimum inter-element distance necessary to avoid spatial aliasing. We introduce the concept of array aperture and frequency resolution. Then, we present three examples of arrays, the ULA, the ring array and the SA. The array steering vectors are obtained and we discuss about the particularities of each one.

2.4.1 Spatial Sampling Nyquist Criterion

The sensor elements that compose an array can be considered as space samplers that discretize a continuous wavefront propagating at a certain frequency. Similar to the time sampling conditions, which constraints the sampling frequency to be at least twice the maximum frequency of the signal in order to avoid frequency aliasing, the inter-element distance that samples the space must satisfy certain conditions to avoid spatial aliasing. The spatial sampling Nyquist sampling criterion constraints the inter-element distance to be

\[
d \leq \frac{\lambda}{2}, \tag{2.20}\]
where \( \lambda \) is the smallest wavelength present in any frequency components of the arriving signal. For arrays that do not satisfy this condition, spatial aliasing will occur.

### 2.4.2 Array Aperture

The array aperture is a continuous spatial extension where the spatial signals are collected [10]. The extension of the aperture determines the accuracy to which the array can determine the DOA of a signal and will set the spatial resolution. The greater the aperture, the finer the spatial resolution of the array. The array spatial resolution is defined as the minimum spatial separation between two signals in which the array is able to distinguish.

The objective is to have the greatest possible aperture for the minimum number of sensor elements. The spatial resolution does not increase if the number of sensor elements of the array increases meanwhile keeping the same aperture. Therefore we will try to space the sensors as far as possible from each other. To satisfy the spatial sampling Nyquist criterion, the distance between adjacent sensors should not be larger than \( \frac{\lambda}{2} \). Therefore the aperture of the array and the spatial sampling Nyquist criterion imposes a constraint in the minimum number of sensors needed.

### 2.4.3 The Uniform Linear Array

The ULA places the elements on the \( y \) axis at equidistant locations. The center of the array is located at the origin of coordinates. The inter-element distance is \( d \leq \frac{\lambda}{2} \). An array having \( d = \frac{\lambda}{2} \) is referred to as the standard linear array (SLA) [9]. Fig. 2.3 shows an example of an ULA with \( K \) elements. The location vector for the \( k^{th} \) element is

\[
p_k = \begin{bmatrix} 0, \left(k - \frac{K+1}{2}\right) d, 0 \end{bmatrix}^T, k = 1, 2, \ldots, K.
\] (2.21)

To find the steering vector we substitute (2.21) in (2.10) and recalling that (2.7) is \( \mathbf{a} = [-\sin \theta \cos \phi, -\sin \theta \sin \phi, -\cos \theta]^T \), we obtain
\[ s = \begin{bmatrix}
    e^{j \frac{2 \pi d}{\lambda} \left( \frac{K-1}{2} \right) \sin \theta \sin \phi} & e^{j \frac{2 \pi d}{\lambda} \left( \frac{K-1}{2} - 1 \right) \sin \theta \sin \phi} & \cdots & e^{-j \frac{2 \pi d}{\lambda} \left( \frac{K-1}{2} \right) \sin \theta \sin \phi}
\end{bmatrix}^T. \quad (2.22) \]

The cone angle is defined as \( \sin \phi_c = \sin \theta \sin \phi \) [10] and creates a cone of DOA uncertainty. Any signal arriving at the same cone angle will have the same steering vector. Therefore this array is not suitable for 3-D beamforming and is most used for 2-D beamforming in a plane that contains the array. This thesis will consider ULA beamforming on the \( xy \) plane or for a polar angle of \( \theta = \pi/2 \).

The particular arrangement of the ULA makes the steering vector to have Vandermonde structure, which in general has the expression \( v = \begin{bmatrix} 1 & z & \cdots & z^{K-1} \end{bmatrix}^T \). The Vandermonde structure of the ULA is useful for different applications such as DOA estimation, beamforming in correlated signal environment, and beampattern synthesis [9]. Also the delay is constant between adjacent elements, therefore the techniques used for FIR filters are applicable here, like minimax, spectral weighting methods, equiripple tapering, and in particular the Dolph-Chebyshev tapering.
2.4.4 The Ring Array

The ring array places the elements at equidistant locations on a circle of radius $r$ lying on the $xy$ plane. The inter-element distance is $d = 2r \sin \left( \frac{\pi}{K} \right)$ and the inter-element arc distance is $\frac{2\pi}{K}r$. The center of the array is located at the origin of coordinates. Fig. 2.4 shows an example of ring array with $K$ elements. The $k$th element in spherical coordinates is $(r, \frac{\pi}{2}, \phi_k)$. The location vector for the $k$th element in cartesian coordinates is

$$p_k = [r \cos \phi_k, r \sin \phi_k, 0]^T, k = 1, 2, ..., K,$$  \hspace{1cm} (2.23)

where we used the expression (2.2) and $\phi_k = (k - 1) \frac{2\pi}{K}$. To find the steering vector we substitute (2.23) and (2.7) in (2.10). Rearranging and using some trigonometrical relationships we obtain the steering vector expression for the ring array as

$$s = \begin{bmatrix} e^{jkr \sin \theta \cos(\phi - \phi_1)} & e^{jkr \sin \theta \cos(\phi - \phi_2)} & \cdots & e^{jkr \sin \theta \cos(\phi - \phi_k)} \end{bmatrix}^T.$$  \hspace{1cm} (2.24)

The ring array suffers of a two DOA ambiguity angles in elevation. For a particular azimuth angle $\phi$, any signal arriving at polar angles of $\theta$ or $\pi - \theta$ will yield the same $\sin \theta$. 
value, therefore those polar angles will create the same steering vector and signals arriving from both elevations will be taken as the same signal. Therefore, this beamformer is said to have an elevation coverage of $180^\circ$. Despite this limitation, the ring array is widely used for 3-D beamforming.

The ring array steering vector does not have the Vandermonde structure of the ULA, the delay between adjacent elements is not constant anymore; therefore, the techniques used on FIR filters are of no use here. Different and particular techniques tailored to the ring array should be used for equiripple beamforming, decorrelation, DOA estimation, etc.

### 2.4.5 The Spherical Array

The SA places the elements at particular locations on a sphere of radius $r$ centered

![Figure 2.5: Spherical array with $K$ elements.](image)
at the origin of coordinates. Fig. 2.5 shows an example of SA with $K$ elements. The $k^{th}$ element in spherical coordinates is $(r, \theta_k, \phi_k)$. The location vector for the $k^{th}$ element in cartesian coordinates is

$$\mathbf{p}_k = [r \sin \theta_k \cos \phi_k, r \sin \theta_k \sin \phi_k, r \cos \theta_k]^{T}, k = 1, 2, ..., K,$$  \hspace{1cm} (2.25)

where we used the expression (2.2). To find the steering vector we substitute (2.25) and (2.7) in (2.10). Rearranging and using some trigonometrical formulas we obtain the steering vector expression for the SA as

$$\mathbf{s} = \left[ e^{jkr \sin \theta \sin \theta_k \cos (\phi - \phi_k) + \cos \theta \cos \theta_k} \cdots e^{jkr \sin \theta \sin \theta_K \cos (\phi - \phi_K) + \cos \theta \cos \theta_K} \right]^{T}. \hspace{1cm} (2.26)$$

The SA does not suffer of any DOA ambiguity. This beamformer have an azimuth and elevation coverage of 360°. Similarly to the ring array, the SA steering vector does not have the Vandermonde structure of the ULA.

The uniform arrangement of $K$ elements in a SA is still an open problem in the mathematical community [67]. In contrast to the ULA and the ring array, there are a few choices for the SA to arrange all the elements with a perfect constant inter-element distance. Those choices correspond to the vertices of the five Platonic solids where all of them present the same distance between adjacent elements. In practice, the number of elements needed in the SA is much more than the twenty elements of the largest platonic solid. When more elements are needed, there are three preferred methods to place them on a sphere, the equiangle, the Gaussian sampling, and the near uniform sampling [20].

A) The equiangle method

This method arranges the elements on a sphere through rotating a $K$ elements ring array $K/2$ times on a sphere. The total number of elements generated by this method is $K^2$. The advantage is that all elements are separated by an equal angle in azimuth and elevation.
The disadvantage is that the elements are not uniformly distributed and the closer to the rotation axis, the denser the element distribution.

**B) The Gaussian sampling method**

This method arranges the elements equiangular in azimuth but not in elevation. The angles in elevation are computed as the zeros of the $\frac{K}{2}$ degree Legendre polynomial, where $K$ is the number of elements in the $xy$ plane. The total number of elements in this method is $K^2$.

**C) The near uniform sampling method**

The platonic solids are used if the number of elements needed is less than 20. For larger number of elements, several authors propose different schemes; Harding and Sloane [68], Lebedev [69], Hoffman [19], and Sengupta *et al.* [70] among others. The advantage of this sampling method is that it requires the smallest number of elements.

### 2.5 Beamformer Output for Narrowband Signals

This section obtains the mathematical expression of the beamformer output for an array input signal composed of a single narrowband signal as defined in section 2.3. Then, the model for the beamformer output signals is shown as well as the beamformer output power, the $SNR$ and $SINR_0$.

#### 2.5.1 Element Space Narrowband Signal Beamforming

As explained in the previous chapter, the array beamforming for narrowband signals using element space can be done with a group of complex valued coefficients that separately multiply the sensor input signals, then the weighted signals are added together to obtain the beamformer output. The array output mathematical expression for an array
of $K$ elements is

$$z(t) = \sum_{k=1}^{K} v_k^* u(p_k, a, \omega, t), \quad (2.27)$$

where $(\cdot)^*$ represent complex conjugation, $v_k$ is a complex valued coefficient of weight $z(t)$. 

Fig. 2.6(a) shows the narrowband beamformer. Arranging the array weights in a vector as

$$v = [v_1 \ v_2 \ \cdots \ v_k]^T \quad (2.28)$$

and also the array input signal in a vector using (2.9) and (2.10) as

$$u(p_1, p_2, \ldots, p_K, a, \omega, t) = s(t)s, \quad (2.29)$$

the beamformer output (2.27) can be written as

$$z(t) = v^H (s(t)s). \quad (2.30)$$

This operation can be seen as the internal product of the array weight vector and the input signal vector. The reason why the weight vector elements $v_k$ should be conjugated...
in (2.27) is now clear from (2.28) and (2.30). The beamformer output depends on the weight vector chosen \( \mathbf{v} \) and the steering vector of the signal \( \mathbf{s} \). The beamformer that is designed to have the output \( z(t) = s(t) \) is referred to as the distortionless beamformer. For a distortionless beamformer the internal product of the array weight vector and the signal steering vector should be

\[
\mathbf{v}^H \mathbf{s} = 1, \tag{2.31}
\]

which is referred to as the distortionless constraint. In section 2.6 we will find several solutions for the weight vector that satisfies the distortionless constraint.

### 2.5.2 Beamformer Output Signal Model

The beamformer output expression for an acoustic field composed of the SOI, \( L \) interferences, and background noise is

\[
z(t) = \mathbf{v}^H \mathbf{u}(t) = s(t)\mathbf{v}^H \mathbf{s}_0 + \sum_{l=1}^{L} i_l(t)\mathbf{v}^H \mathbf{i}_l + n(t)\mathbf{v}^H \mathbf{n}. \tag{2.32}
\]

The beamformer output due to the SOI is

\[
z_s(t) = s(t)\mathbf{v}^H \mathbf{s}_0. \tag{2.33}
\]

The beamformer output due to the \( l^{th} \) interference is

\[
z_{i_l}(t) = s(t)\mathbf{v}^H \mathbf{i}_l. \tag{2.34}
\]

The beamformer output due to all interferences is

\[
z_i(t) = \sum_{l=1}^{L} i_l(t)\mathbf{v}^H \mathbf{i}_l. \tag{2.35}
\]

Finally, the beamformer output due to the noise is

\[
z_n(t) = n(t)\mathbf{v}^H \mathbf{n}. \tag{2.36}
\]

43
2.5.3 Beamformer Output Power

Now, we will obtain the beamformer output power for the SOI, interferences and noise. But first, let’s define some power measurements at the input of the array elements.

A) Power at the input sensor elements

The input power at each sensor element due to the SOI is defined as $\sigma_s^2 = E[|s(t)|^2]$ where $E[X]$ is the expectation operator of the random variable $X$, and $|X|$ represents the magnitude of $X$. For the interferences and noise, the input powers are defined similarly obtaining $\sigma_i^2$ and $\sigma_n^2$. Usually the noise power is given as a $SNR$ at any input sensor or $SNR_i = \frac{\sigma_s^2}{\sigma_n^2}$. The $l^{th}$ interference power is given with respect the SOI power as the signal-to-interference ratio ($SIR$) at any input sensor, or $SIR_{il} = \frac{\sigma_i^2}{\sigma_s^2}$. Alternatively, is given as interference to noise ratio ($INR$) with expression $INR_{il} = \frac{\sigma_i^2 s^2}{\sigma_n^2}$ . The signal to interference plus noise power at the array input is defined as $SINR_i = \frac{\sigma_s^2}{\sigma_s^2 + \sum_{l=1}^{L} \sigma_i^2}$ for interferences whose origin is uncorrelated.

B) Beamformer output power

The beamformer output power due to the SOI is

$$ P_s = E[|z_s(t)|^2] = v^H R_s v, \quad (2.37) $$

where the beamformer output expression $z_s(t)$ comes from (2.33) and $R_s$ is the $(K \times K)$ SOI spatial nominal correlation matrix $R_s = \sigma_s^2 s_0 s_0^H$. Similarly for the interferences we find the beamformer output power to be

$$ P_{il} = E[|z_{il}(t)|^2] = v^H R_{il} v, \quad (2.38) $$

where $R_{il} = \frac{\sigma_i^2}{\sigma_s^2} s_0^i s_0^H$.

The beamformer output due to all interferences is

$$ P_i = E[|z_i(t)|^2] = v^H R_i v, \quad (2.39) $$
where $R_i$ is the correlation matrix of all interferences in the field.

The beamformer output power due to the noise is

$$P_n = E \left[ |z_n(t)|^2 \right] = v^H R_n v,$$  \hspace{1cm} (2.40)

where $R_n = \sigma_n^2 I$, since the noise is spatially uncorrelated. $I$ represents the $(K \times K)$ identity matrix. Therefore (2.40) becomes

$$P_n = \sigma_n^2 v^H v = \sigma_n^2 \|v\|^2.$$  \hspace{1cm} (2.41)

This result indicates that the background noise will be reduced by the squared Euclidean norm of the weight vector.

In general, the output power for a field composed of (2.19) is not the sum of the individual power terms. However, when the SOI is uncorrelated with the interferences and the noise [66], the beamformer output power can be written as

$$P_{\text{sin}} = E \left[ |z(t)|^2 \right] = v^H R_s v + v^H R_i v + \sigma_n^2 \|v\|^2.$$  \hspace{1cm} (2.42)

The importance of the correlation matrices will be clear when we study optimum beamformers, which use the information in the correlation data matrices to cancel the interferences and reduce the background noise.

### 2.5.4 Beamformer Output Signal to Noise Ratio

The beamformer output $SNR$ is defined as

$$SNR_0 = \frac{P_s}{P_n},$$  \hspace{1cm} (2.43)

where $P_s$ is defined in (2.37) and $P_n$ in (2.41). Plugging in their expression we find the general beamformer output $SNR$ expression

$$SNR_0 = \frac{v^H R_s v}{\sigma_n^2 \|v\|^2},$$  \hspace{1cm} (2.44)
For a distortionless beamformer the signal power is $P_s = \mathbf{v}^H \mathbf{R}_s \mathbf{v} = \sigma_s^2$, therefore (2.44) becomes

$$SNR_0 = \frac{SNR_i}{\|\mathbf{v}\|^2},$$

(2.45)

### 2.5.5 Beamformer Output Signal to Interference Plus Noise Ratio

For an acoustic field that contains interferences is important to know the relation between the SOI output power and the unwanted interferences and noise output power. The beamformer output $SINR$ is defined as

$$SINR_0 = \frac{P_s}{P_i + P_n},$$

(2.46)

where $P_s$ is defined in (2.37), $P_n$ in (2.41), and

$$P_i = \mathbf{v}^H \mathbf{R}_i \mathbf{v}.$$  

(2.47)

Plugging in their expression in (2.46), we find the general beamformer output $SINR$ expression

$$SINR_0 = \frac{\mathbf{v}^H \mathbf{R}_s \mathbf{v}}{\mathbf{v}^H \mathbf{R}_i \mathbf{v} + \sigma_n^2 \|\mathbf{v}\|^2},$$

(2.48)

### 2.6 The Beamforming Process

This section finds the beamformer output when implementing a particular choice of weights referred to as the delay-and-sum weights, where the acoustic field is composed of the SOI only. Later on, we will find the weight vector solution to a constrained minimization problem when the acoustic field is composed of the SOI and background noise. The section ends with the optimum beamforming, that finds the weight vector as a constrained minimization problem when the acoustic field is composed of the SOI, interferences, and background noise. Interesting conclusions are drawn along this section.
2.6.1 Beamforming with Delay-and-sum Weights

For a narrowband element space beamformer, the processing applied to each sensor reduces to a scalar coefficient or weight. If we want to maintain the SOI undistorted, the beamformer should exhibit unity gain for the SOI arriving at its particular direction. If the SOI arrives from a DOA of \((\theta_0, \phi_0)\) the signal read at the \(k^{th}\) element would be from (2.12)

\[ u_s(p_k, a_0, \omega, t) = s(t)e^{-j(ka_T^0 p_k)}, \]  

(2.49)

Considering that the signal information is located in the term \(s(t)\), the ideal beamformer output should be \(z_s(t) = s(t)\). This is equivalent to read the SOI at the origin of coordinates or \(u_s(0, a_0, \omega, t) = s(t)\).

If we choose the coefficient that will be multiplying the signal at the \(k^{th}\) element with a complex value of \(v_k^* = \frac{1}{K}e^{j(ka_T^0 p_k)}\) and reminding that \(e^{j\alpha}e^{-j\alpha} = 1\), we can perform the operation \(v_k^* u_s(p_k, a_0, \omega, t) = \frac{1}{K}s(t)\). After adding all the weighted signals in all array elements, we obtain the beamformer output expression

\[ z_s(t) = \sum_{k=1}^{K} v_k^* u_s(p_k, a_0, \omega, t) = \sum_{k=1}^{K} \frac{1}{K}s(t) = s(t) \]  

(2.50)

which is what we wanted. The beamformer that uses this choice of weights is referred to as the delay-and-sum, the spatial matched, or the conventional beamformer. The weights were smartly chosen so they introduce a delay at each sensor signal such that the SOI presents the same phase shift after being weighted. By adding the weighted signals we obtain the desired beamformer output. That is why these choice of weights are termed delay-and-sum.

We can also obtain the beamformer output in vectorial form. The array input SOI vector from (2.13) and repeated here is

\[ u_s(a_0, \omega, t) = s(t)s_0. \]  

(2.51)
Arranging the delay-and-sum weights in a vector we have

\[
v = \frac{1}{K} \begin{bmatrix}
e^{j\kappa a_0^T p_1} \\
e^{j\kappa a_0^T p_2} \\
\vdots \\
e^{j\kappa a_0^T p_K}
\end{bmatrix} = \frac{1}{K} s_0. \tag{2.52}
\]

Therefore the delay-and-sum weight vector is nothing but the SOI array steering vector scaled by the number of array sensors \(K\). Arranging (2.50) in vectorial form, we have

\[
z_s(t) = v^H u_s(a_0, \omega, t) = s(t)v^H s_0. \tag{2.53}
\]

The product \(v^H s_0\) is indeed equal to unity. It is not difficult to prove that \(s_0^H s_0 = K\) using (2.10). Finally, putting (2.52) in \(v^H s_0\) yields one.

A) Remarks

There are infinite solutions to the equation \(v^H s_0 = 1\). However, in practice the acoustic field is composed of SOI, interferences, and background noise. The objective of the beamformer is not only to keep the SOI undistorted but to attenuate as best as possible the interference and the background noise. To attenuate the noise, we need to combine the weighted signals in a coherent way so the spatial uncorrelated noise is diminished by averaging. In order to attenuate the interferences, the array output should be minimal for different DOAs than that of the SOI. The SOI steering vector \(s_0\) is a vector that belongs to a \(C^K\) dimensional space, where \(C\) is the set of complex numbers. Other signals arriving from different DOAs will have different steering vectors. The delay-and-sum weight vector \(v\) has the same direction as \(s_0\) (2.52), therefore its internal product will be maximum. The internal product of \(v\) with any other steering vector should be less than the maximum and ideally should be zero. We would need to find the array output to all possible steering vectors to assess the performance of the beamformer for a particular choice of \(v\). Before calculating the array response to signals impinging from all DOAs, lets obtain the beamformer output for other choices of weight vectors.
2.6.2 Delay-and-sum Found as a Minimization Problem

At this point we could find the optimum weight vector that minimizes the beamformer output power due to the background noise only, meanwhile maintaining the SOI undistorted. This is a typical optimization problem expressed as

\[ \mathbf{v} = \arg \min \{ E \left[ |z_n(t)|^2 \right] \}, \quad (2.54) \]

subject to the constraint \( \mathbf{v}^H \mathbf{s}_0 = 1 \). \( z_n(t) \) has been defined in (2.36). The solution of this problem is solved by Lagrange multipliers in Appendix B as

\[ \mathbf{v} = \frac{s_0}{K}. \quad (2.55) \]

This expression is nothing but the delay-and-sum weight vector. Therefore the delay-and-sum weight vector is optimum in minimizing the noise power at the beamformer output meanwhile keeping the beamformer distortionless on the SOI [9, 10]. Interestingly enough, the minimization problem when using \( z_s(t) + z_n(t) \) in (2.54) yields the same solution.

2.6.3 Noise Output Power for Delay-and-sum Weights

For delay-and-sum weights (2.55) is not difficult to prove that \( \| \mathbf{v} \|^2 = \frac{1}{K} \). The beamformer output power due to the background noise from (2.41) is \( P_n = \sigma_n^2 \| \mathbf{v} \|^2 \). Therefore, the background input noise power \( \sigma_n^2 \) is attenuated \( K \) times at the beamformer output.

2.6.4 SNR\(_0\) for Delay-and-sum Weights

The SNR\(_0\) from (2.45) when using delay-and-sum weights becomes

\[ \text{SNR}_0 = K \text{SNR}_i. \quad (2.56) \]

Therefore, the SNR\(_0\) is \( K \) times the SNR\(_i\). The more elements in an array the more gain in SNR\(_0\).
2.6.5 Optimum Beamforming

The delay-and-sum weight vector, even when it is optimum for a field composed of the SOI and background noise, it is not enough when the acoustic field is composed of strong interferences. Also, the delay-and-sum weights are independent of the collected data and disregard the information about the interferences that can be estimated from the data correlation matrix.

If the statistics of the interferences and the noise are known, meaning we have the nominal correlation interference matrix $R_i$ and the power of the noise $\sigma_n^2$, we can obtain a beamformer weight vector that minimizes the beamformer output power subject to a SOI distortionless constraint. The problem is laid out as

$$v = \arg \min \left\{ E \left[ |z(t)|^2 \right] \right\},$$

subject to the constraint $v^H s_0 = 1$. $z(t)$ has been defined in (2.32) and $E \left[ |z(t)|^2 \right]$ in (2.42). The solution of this problem is solved by Lagrange multipliers in Appendix C as

$$v = \frac{R_u^{-1} s_0}{s_0^H R_u^{-1} s_0},$$

(2.58)

where each term of $R_u = R_s + R_i + R_n$ has been defined in subsection 2.5.3. Applying the matrix inversion lemma to $R_u$ and rearranging terms we find that the weight vector expression becomes

$$v = \frac{(R_i + R_n)^{-1} s_0}{s_0^H (R_i + R_n)^{-1} s_0},$$

(2.59)

The beamformer that finds the weight vector as (2.58) or (2.59) is referred to as the MVDR beamformer [9] also called Capon beamformer [25].

In practice we do not know the nominal correlation of the interference plus noise and we need to estimate it from the data. The weight vector is adaptively trained to adapt to the incoming data and the beamformers that use these adaptive weights are referred to as adaptive beamformers [9]. Among the different algorithms to find the
weight vector we can use the sample matrix inversion (SMI) to obtain the inverse of the spatial correlation matrix in (2.58). SMI is a block data processing, therefore the beamformer needs to acquire a block of snapshots before giving a correct output. Other algorithms that are updated at each snapshot are the least mean squares (LMS) and the recursive least squares (RLS) algorithms. The LMS algorithm is a gradient algorithm and computationally very attractive. The drawback is that the convergence rate depends on the eigenvalue spread of the correlation matrix. The RLS requires more computationally complexity than the LMS but the convergence rate does not depend on the correlation matrix eigenvalue spread.

2.7 Beampattern

This section introduces the beampattern or array response to all DOAs for a beamformer with a particular weight vector $\mathbf{v}$.

The beamformer output expression (2.30) is repeated here

$$z(t) = \mathbf{v}^H s(t).$$

(2.60)

The beampattern is defined as the beamformer output power response for a signal with constant amplitude $A(t) = 1$ arriving at any DOA ($\theta, \phi$). Thus, $s(t) = e^{j\omega t}$ and the beampattern expression is

$$B(\theta, \phi) = |\mathbf{v}^H s(\theta, \phi)|^2.\quad (2.61)$$

The beampattern is usually computed for all possible angles and plot in a figure. We now present two examples of beampatterns

2.7.1 Delay-and-sum Beampattern

As an example, for an ULA of $K = 24$ elements, $d = \frac{\lambda}{2}$ inter-element distance and delay-and-sum weights designed for an impinging signal with a DOA of $(\theta_0 = 90^\circ, \phi_0 = 60^\circ)$,
the 2-D and 3-D beampatterns are shown in Fig. 2.7. Fig. 2.7(a) shows the 2-D beampattern having unity gain at $\phi = 60^\circ$ degrees as we expected. Fig. 2.7(b) shows the 3-D beampattern with the DOA cone of uncertainty around the array axis $y$.

From the 2-D beampattern we find that there is a main lobe with a finite width and several sidelobes where the beamformer gain is much lower. The mainlobe width is dictated by the array aperture. The larger the aperture the narrower the mainlobe, therefore for a beamformer with a narrow mainlobe the SOI can be detected with more precision and better resolved for a interference with a close DOA.

Also we find that there are nulls in the beampattern. Those nulls represent orthogonal spatial locations with respect the spatial location represented by the weight vector. Any interference coming from any of those locations will have zero output in the beamformer. On the other hand, any interference coming at a sidelobe with the same power as the sidelobe attenuation will be output with unity gain, and taken as if it is part of the SOI.
2.7.2 MVDR Beampattern

The second example expands the previous one by including an interference with a $SIR_i$ of -30dB and a DOA of $(\theta_0 = 90^\circ, \phi_0 = 48^\circ)$ that arrives at the first sidelobe peak of the beampattern in Fig. 2.7(a). The $SNR_i$ is 0dB. After obtaining the MVDR weights with (2.58) we plot the 2-D and 3-D beampatterns in Fig. 2.8. The 2-D beampattern in Fig. 2.8(a) shows a null where the conventional beampattern has the first side lobe. For comparison purposes we also plot in a fine dashed trace the conventional beampattern from the previous example. The MVDR weights succeeded in placing a null at the interference DOA meanwhile maintaining the distortionless constraint. One of the potential disadvantages of the MVDR beamforming is that the mainlobe width is increased as well as the gain in some sidelobes. In some particular situations this increment in the sidelobes gain could result in a worse performance if for some reason a new interference appears at the same DOA where the sidelobe is placed.
2.8 Performance Measurements

This section introduces performance measurements to assess and compare beamformers in a quantitative aspect.

2.8.1 Array Resolution

The 3dB beamwidth (HPBW) and the distance to the first nulls (BW_{NN}) of the beam-pattern mainlobe are two measures to quantify the array resolution.

The HPBW happens for the angles where \( B(\theta, \phi) = \frac{1}{2} \). This measure is commonly used for the ULA where the beampattern is defined in 2-D. For 3-D beampatterns, the HPBW is the maximum distance where \( B(\theta, \phi) = \frac{1}{2} \) in any direction.

Analogous to the HPBW, the BW_{NN} is the angular distance that happens for the angles that make \( B(\theta, \phi) = 0 \).

2.8.2 Array Directivity and Directivity Index

The array directivity is defined as the ratio between the beamformer output power due to a signal arriving from a DOA: \((\theta_0, \phi_0)\) and the beamformer output power due to interferences distributed uniformly over a sphere. The mathematical expression is

\[
D = \frac{|B(\theta_0, \phi_0)|^2}{\frac{1}{4\pi} \int_0^\pi d\theta \int_0^{2\pi} d\phi \sin \theta |B(\theta, \phi)|^2}.
\]  

(2.62)

For a distortionless beamformer with weights steered to \((\theta_0, \phi_0)\) the numerator of (2.62) is one. In general, this expression must be evaluated numerically. In particular, for the SLA, \( D = \|v\|^{-2} \) and the directivity does not depend on the steering direction. for ULA where \( d < \frac{\lambda}{2} \) the directivity depends on the steering direction [9].

The directivity is maximum for a beamformer with delay-and-sum weights. For any other weights (i.e. MVDR) the directivity always decreases.
Frequently, the directivity is expressed in dB and is referred to as the directivity index \((DI)\)

\[
DI = 10 \log_{10} D.
\] (2.63)

2.8.3 Array Gain Against White Noise

As we already seen, one of the objectives of a beamformer is to improve the \(SNR_0\) by adding signals coherently and noise incoherently. The array gain measures this improvement. The array gain is defined as the improvement in \(SNR\) when using the beamformer. It is the ratio between the \(SNR_0\) and the \(SNR\) in the received signal from a sensor [9].

\[
A_w = \frac{SNR_0}{SNR_i}.
\] (2.64)

The subscript \(w\) refers to the spatially uncorrelated noise input. The \(SNR\) at the input of a sensor is always \(SNR_i = \frac{\sigma^2_s}{\sigma^2_n}\). For a distortionless beamformer we found in (2.45) that \(SNR_0 = \frac{SNR_i}{\|v\|^2}\). Therefore, the array gain for a distortionless beamformer is

\[
A_w = \|v\|^{-2},
\] (2.65)

which is independent of the geometry of the array as long as the noise is uncorrelated and the beamformer is distortionless.

For a SLA the array directivity is the same as the array gain. However, for a ULA with \(d < \frac{\lambda}{2}\) and for other geometries it is not.

2.8.4 Array Gain Against Interferences and Noise

The array gain against interferences and noise is defined as

\[
A_g = \frac{SINR_0}{SINR_i},
\] (2.66)

where the \(SINR_0\) was defined in subsection 2.5.5 and the \(SINR_i\) in subsection 2.5.3.
2.9 Chapter Summary

This chapter expands the beamforming concepts introduced in chapter 1 and uses a mathematical treatment on the signal models, array geometry, beamforming and performance measures.

The beamforming process is explained for conventional and optimum beamforming. The beampattern concept is laid out as well as the beamformer performance measures that will be used in the following chapters.
Chapter 3

An Improved Partial Adaptive Narrow-band Beamformer Using Concentric Ring Array

Partial adaptation is often used to reduce the computation and improve tracking ability of an adaptive array. In some practical situations, the received signal to be processed contains some interferences whose characteristics are known. The previously proposed partially adaptive CRA is not able to utilize the prior information of known interferences without sacrificing the number of DOFs, which will cause higher steady state error and smaller number of interferences that can be canceled. We propose in this paper an improved partially adaptive CRA that can utilize the prior knowledge to improve performance and maintain the same number of DOFs. The proposed method designs the non-adaptive weights to remove the known interferences, and is shown to provide much faster convergence speed and lower steady state error than the original method.

3.1 Introduction

Array signal processing is a popular technique in acoustic and radar applications because it adds the spatial dimension in addition to time, and hence improves the performance in signal acquisition and interference rejection. Perhaps the most popular array is the ULA
because of its simple structure that allows very efficient processing in DOA estimation and signal enhancement. ULA is very effective for 2-D beamforming. In 3-D space, ULA creates ambiguity with its beampattern warping around the array [11] and is not appropriate for 3-D beamforming. CRA [14], [16] has been proposed for 3-D beamforming which is able to eliminate DOA ambiguity and provides frequency invariant design for broadband beamformer. In practice, the direction of all the interferences may not be known \textit{a priori} and the weights in the beamformer are made adaptive to minimize the array output subject to a set of constraints including a unity gain in the look direction of the signal. 3-D beamforming typically requires a huge number of array elements (over one hundred) to achieve good performance. As a result, the convergence speed and tracking performance could be poor. Reducing the number of adaptive coefficients is necessary to improve performance and achieve real time processing. Partially adaptive approach has been proposed to reduce the number of adaptive coefficients. Partial adaptive array can be beamspace or element space. Recently, Li & Ho [50] proposed an element space partially adaptive array in which each ring is considered as a sub-array that performs fixed beamforming using the delay-and-sum weights [23], and the outputs from the concentric rings are combined using a set of adaptive weights. The partially adaptive array improves the convergence speed significantly compared to fully adaptive array, and the steady state residual interference and noise power is only increased slightly. One drawback of the previously proposed partially adaptive circular array is that the number of interferences that can be removed is limited by the number of adaptive weights which is equal to the number of rings. Furthermore, it does not make use of any prior knowledge about any interference whose directions may be known. Although constraints on the adaptive weights can be used to eliminate the interferences quickly from the known directions; however, this will unnecessarily reduce the number of DOFs in the adaptive weights which will increase the steady state residual error level and limits the extra
number of interferences that can be removed. In some scenarios, the characteristics and DOA of some interferences are known. In this paper, we propose an enhanced partially adaptive array that can make use of this prior knowledge to improve convergence speed, whilst at the same time maintaining the same number of DOF’s in the adaptive weights to assure good steady state behavior. The idea of the proposed method is not to restrict the intra-ring weights to be delay-and-sum, but rather select the weights so that it minimizes the output in each ring through the knowledge of the signal’s DOA and the known DOA’s and characteristics of some of the interferences. The output from every ring is then fed to a Generalized sidelobe canceler (GSC) based adaptive structure [50] to perform a second level of optimization to remove the interferences whose characteristics are not known. The proposed array reduces interferences much faster as compared to the original method and achieves smaller steady state residual error level. This paper is organized as follows. Section 3.2 reviews briefly the fully and partially adaptive CRA. Section 3.3 presents the proposed improved partially adaptive array. Section 3.4 contains the experimental results and analysis, the paper is summarized in Section 3.5 and acknowledgments are in section 3.6.

3.2 Previous Adaptive CRA Designs

3.2.1 Fully Adaptive CRA

We shall consider a concentric circular multi ring array that is composed of $M$ concentric rings as shown in Fig. 3.1. The number of receiving elements in ring $m$ is $K_m$ and the total number of elements is $K = K_1 + \ldots + K_M$. The output of the array at time $t$ is:

$$z(t) = \sum_{m=1}^{M} \sum_{k=1}^{K_m} v_{mk}^* u_{mk}(t) = v^H u(t),$$

(3.1)

where $u_{mk}(t)$ is the received signal at the element $k$ of ring $m$. $v_{mk}$ is the corresponding array weight, and (*) represents complex conjugate. In vector form $v$ is the weight
Figure 3.1: Concentric Ring Array (CRA).

vector and \( \mathbf{u}(t) \) is the input vector. In fully adaptive array the weighting coefficients are found adaptively by minimizing the output power subject to a set of constraints (Linearly Constrained Minimum Variance criteria, or LCMV). Usually, it includes one constraint to maintain unity gain at the DOA of the desired signal. This approach has the advantage of reducing the interferences from any DOA’s (different than the desired signal DOA) and achieving a low steady state residual error. The disadvantage of this approach is its slow convergence, poor tracking ability and high computational cost.

### 3.2.2 Partially Adaptive CRA

The partially adaptive CRA [50], [51] overcomes these deficiencies by adapting only one weight per ring, thereby reducing greatly the number of weights that need to be adapted. The technique considers each ring as a sub-array in which fixed beamforming using delay-and-sum weights is applied. The outputs from different rings are then combined
adaptively to form the final output. To be specific, let $h_m$ be the vector containing the delay-and-sum weights for ring $m$, whose $k^{th}$ element is:

$$h_{mk} = \frac{1}{K_m} e^{j\frac{2\pi}{\lambda} R_m \left[ \sin \theta_0 \cos (\phi_0 - \nu_k) \right]} \quad k = 1, ..., K_m,$$

(3.2)

where $\lambda$ is the wavelength, $R_m$ the radius of the ring, $\theta_0$ and $\phi_0$ are the desired signal’s elevation and azimuth angles respectively, and $\nu_k = 2\pi k / K_m$ is the azimuth angle of the $k^{th}$ element in the ring. Denote the received signal vector of ring $m$ as $u_m(t)$. Then, the output of the ring is:

$$y_m(t) = h_m^H u_m(t) \quad m = 1, ..., M.$$

(3.3)

Each ring output is then multiplied by an adaptive inter-ring weight to find the final output as:

$$z(t) = w^H y(t),$$

(3.4)

where $y(t) = [y_1(t), ..., y_M(t)]^T$ and $w$ the adaptive inter-ring weight vector. The inter-ring weight vector $w$ is found adaptively [50] subject to the same conditions as in the fully adaptive approach; i.e. by minimizing the output power $E[|z(t)|^2]$ subject to a set of constraints. This partially adaptive array has better tracking ability and faster convergence due to the small number of adaptive weights; however, the steady state error is somewhat higher than that in the fully adaptive array. Also, the number of interferences that can be canceled is limited by the number of adaptive weights minus the number of constraints.

### 3.3 Improved Partially Adaptive Concentric Ring Array

Fig. 3.2 shows the proposed design block diagram. In practice, the characteristics and DOA’s of some interferences may be available. Instead of using simply delay-and-sum
weights within each ring as in [14], the proposed method selects the intra-ring weights $g_m$ to minimize the known interferences in each ring. They are found by minimizing the output power of each ring $E[|y_{mC}(t)|^2]$ subject to the unity gain constraint at the desired DOA, where $y_{mC}(t)$ is the output of the ring $m$ when only the desired signal, the known interferences and the noise are present. Since the intra-ring weights $g_m$ are found by minimizing the output power of the $m^{th}$ ring over the known interferences only, the resultant beampattern from $g_m$ may create higher gain in some other directions in which the unknown interferences may come in. The consequence is putting extra burden on the adaptive weights and could lead to a higher steady state residual error level in the final array output. To keep this effect minimal, we modify the cost function to find $g_m$ as follows by adding a penalty term that constrains the distance between $g_m$ and the delay and sum weights:

$$g_m = \arg \min \left\{ (1-\alpha)E[|y_{mC}(t)|^2] + \alpha ||g_m - h_m||^2 \right\}, \quad (3.5)$$

subject to $K_m h_m^H g_m = 1$; where $\alpha$ is a penalty factor. Equation (3.5) provides a tradeoff optimization for the improved intra-ring weight vector. A smaller value of $\alpha$ allows larger deviation from the delay-and-sum vector and a larger reduction amount.

Figure 3.2: Block Diagram of the proposed partially adaptive array.
of known interferences. The solution reduces back to delay and sum if $\alpha$ is equal to unity. The minimization problem can be solved by the use of Lagrange multipliers. This method finds the minimum of a function subject to one or more constraints by introducing a Lagrange multiplier $\eta$, and creating an auxiliary function as:

$$J = (1 - \alpha)E[|y_mC(t)|^2] + \alpha ||g_m - h_m||^2 + \eta \left(1 - K_mg_m^Hh_m\right), \quad (3.6)$$

where

$$E[|y_mC(t)|^2] = g_m^HR_mCg_m, \quad (3.7)$$

$$||g_m - h_m||^2 = (h_m - g_m)^H(h_m - g_m), \text{ and } R_mC = E[u_mC(t)u_mC(t)^H].$$

Taking the derivative of $J$ with respect to $g_m^*$ yields:

$$\frac{\partial J}{\partial g_m^*} = (1 - \alpha)R_mCg_m - \alpha (h_m - g_m) - \eta K_mh_m. \quad (3.8)$$

Setting the gradient to zero and solving for $g_m$ gives:

$$g_m = (\alpha + \eta K_m) \left[(1 - \alpha)R_mC + \alpha I\right]^{-1}h_m. \quad (3.9)$$

Since $1 = K_mh_m^Hg_m$ from the constraint, pre-multiplying (3.9) by $K_mh_m^H$ forms:

$$1 = K_m(\alpha + \eta K_m)h_m^H[(1 - \alpha)R_mC + \alpha I]^{-1}h_m. \quad (3.10)$$

Finally combining (3.9) and (3.10) by eliminating $\alpha + \eta K_m$ yields:

$$g_m = \frac{[(1 - \alpha)R_mC + \alpha I]^{-1}h_m}{h_m^H[(1 - \alpha)R_mC + \alpha I]^{-1}h_m}K_m \quad m = 1, ..., M. \quad (3.11)$$

It is important to note that $R_mC$ is the covariance matrix from the essential input and can be theoretically found to include the information of the known interference characteristics and noise as:

$$R_mC = s_m^s_0s_m^H + \sum_{l=1}^{C} \sigma_l^2 i_mH_l + \sigma_n^2 I \quad m = 1, ..., M. \quad (3.12)$$
where $\sigma_i^2$ is the power of the $l^{th}$ interference from a total of $C$ known interferences, $\sigma_n^2$ is the noise power, and $s_{m0}$ is the delay vector for a signal coming from the desired DOA onto ring $m$ and has the elements:

$$s_{m0k} = e^{j\frac{2\pi}{\lambda}R_m[\sin \theta_0 \cos (\phi_0-v_k)]} \; k = 1, \ldots, K_m.$$  (3.13)

$i_{ml}$ is the delay vector for the $l^{th}$ interference coming from its known DOA$l$, and is calculated as in (3.13) by replacing $(\theta_0, \phi_0)$ by $(\theta_l, \phi_l)$. These delay vectors are only dependent on the ring’s geometry and on the DOA’s of the incoming signal and interferences. The penalty factor has to be carefully selected so that the array is still tuned to the desired DOA and at the same time rejects the known interferences implicit in the essential covariance matrix. If the penalty term is too close to zero, a negative side effect could appear in which a possible higher gain than the one from the delay-and-sum weights may occur in the directions of the unknown interferences. Once the intra-ring weights are obtained from (3.11), the output of each ring is calculated as:

$$y_m(t) = g_m^H u_m(t) \; m = 1, \ldots, M.$$  (3.14)

Each ring output is then multiplied by an adaptive inter-ring weight to find the form output as:

$$z(t) = w^H y(t).$$  (3.15)

The adaptive inter-ring weights $w$ are found using a GSC configuration as in [50]. It decomposes the adaptive weights into constrained and unconstrained components. The constrained part is not adaptive and the associated array response component is called the quiescent response. The unconstrained part consists of a blocking matrix which eliminates the desired signal, followed by the adaptive weights. The adaptive algorithm used is the normalized least mean squares (NLMS) [10] that minimizes the instantaneous output squared magnitude $|z(t)|^2$ through iteration.
3.4 Experimental Results

To demonstrate the performance of the improved partially adaptive array, we present a design example as shown in Fig. 3.2 for the processing of a narrowband signal at 1 kHz. The CRA has 68 elements arranged in 4 rings. The elements are equally spaced in each ring, and the number of elements in the rings, from the innermost are 12, 12, 20 and 24. Each ring is treated independently to calculate the $g_m$ weights. The output of each ring is then multiplied by the adaptive weights $w$ to obtain the final output.

The received array signal is simulated by a computer, which contains the desired signal coming from the DOA ($\theta = 90^\circ$, $\phi = 0^\circ$). The interference is composed of three narrowband signals of 1kHz coming from DOA’s: ($\theta = 75^\circ$, $\phi = 120^\circ$), ($\theta = 90^\circ$, $\phi = 150^\circ$) and ($\theta = 80^\circ$, $\phi = 220^\circ$). The background noise is Gaussian and omni-directional.
The signal to interference ratio (SIR) is -25dB, -35dB, and -30dB respectively, and the signal to background noise ratio is 0dB. The number of ensemble averages is 100. The penalty term $\alpha$ is set to 0.1. The processing results are presented in Fig. 3.3 and in Table 3.1.

Fig. 3.3 shows the convergence result of the residual interference and noise power in four situations. The first graph shows the results for fully adaptive array. The convergence speed is slower than that in the other situations; however it achieves the smallest steady state error. The second graph shows the original partially adaptive array results [50]. The convergence rate is faster; however, the steady state error is slightly higher than in the fully adaptive array situation. The lower two graphs show the convergence results with the new improved array for two cases. In case I, the interference coming at, DOA ($\theta = 80^\circ, \phi = 220^\circ$) is known. Case II is when two interferences coming at DOA's ($\theta = 90^\circ, \phi = 150^\circ$) and ($\theta = 80^\circ, \phi = 220^\circ$) are known. In both cases the initial residual interference and noise power is much smaller than those in the fully adaptive and the original partially adaptive method. This is due to the elimination of the known interferences at the intra-ring level from the better intra-ring weights design given in (3.11). Also, convergence rates are faster. Their steady state errors are smaller than in the original partially adaptive array and closer to the values from the fully adaptive array. Case I initially converges slower than case II, but achieves smaller steady state error. Case II has the fastest initial convergence; however, after this initial stage, the values become larger than that in case I. One possible explanation is that the reshaping effect on the beampattern formed from $g_m$ to cancel the two known interferences can force higher gain values at other spatial locations, thereby increasing the burden to the adaptive weights that now have to cancel the unknown interference and noise with higher power.

Table 3.1 shows the interference and noise power at different iterations. The second
Table 3.1: Comparison of residual interference and noise power, $\alpha = 0.1$

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Fully</th>
<th>Org. Partial</th>
<th>Case I</th>
<th>Case II</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11.5665</td>
<td>11.9432</td>
<td>2.0878</td>
<td>0.7087</td>
</tr>
<tr>
<td>400</td>
<td>0.0943</td>
<td>0.0604</td>
<td>0.0489</td>
<td>0.0923</td>
</tr>
<tr>
<td>1200</td>
<td>0.0324</td>
<td>0.0456</td>
<td>0.0359</td>
<td>0.0629</td>
</tr>
<tr>
<td>2000</td>
<td>0.0267</td>
<td>0.0385</td>
<td>0.0306</td>
<td>0.0505</td>
</tr>
<tr>
<td>4000</td>
<td>0.0183</td>
<td>0.0283</td>
<td>0.0210</td>
<td>0.0256</td>
</tr>
<tr>
<td>8000</td>
<td>0.0163</td>
<td>0.0257</td>
<td>0.0181</td>
<td>0.0210</td>
</tr>
<tr>
<td>12000</td>
<td>0.0160</td>
<td>0.0254</td>
<td>0.0177</td>
<td>0.0203</td>
</tr>
<tr>
<td>20000</td>
<td>0.0163</td>
<td>0.0252</td>
<td>0.0183</td>
<td>0.0198</td>
</tr>
</tbody>
</table>

The column is the results for fully adaptive array. The third column is the results for the original partially adaptive array. Columns four and five are the results for case I and II of the new improved partially adaptive array. The starting values for case I and case II are much lower than that in the original approaches. The effectiveness of the new intra-ring weights design is evident, and some of the interferences have been eliminated at this stage. In case II the starting value is smaller than that in case I because one more interference is known. The convergence in case I is faster than the convergence in the original partially adaptive array and at all times it maintains smaller values. Case II starts with a smaller value than all other cases; however, it cannot achieve smaller steady state error than the original partially adaptive array until the 4000th iteration. Then, the residual interference and noise error becomes smaller.

### 3.5 Summary

In this paper, we have proposed an improved partially adaptive CRA for the beamforming of a narrow-band signal in which the knowledge of some interferences are available. The partial adaptive array partitions the array weight vector into two components: intra-ring weights and inter-ring weights. The intra-ring weights are designed through LCMV over the interferences whose characteristics are known. The inter-ring weights are adaptive to
remove the unknown interferences. The intra-ring weight design also includes a penalty term to limit the deviation from delay-and-sum solution to avoid the possibility of having higher gain over DOA’s of the unknown interferences. The good performance of the proposed partially adaptive array is corroborated through simulations. The future plan is to investigate the choice of the penalty factor with respect to the number of known interferences and the signal-to-noise ratio. We also plan to compare our technique to the low rank dimension reduction approach for improving the adaptation speed.

\section*{3.6 Acknowledgments}

The authors would like to thank Willard Larkin, Jeffrey Short, Michael Wicks and Robert Bolia, for providing technical insight and support for the project. This research is supported by US AFOSR under contract FA9550-04-C-0058 and partially by the Polytechnic University of Puerto Rico.
Chapter 4

Optimizing The Performance of the Partial Adaptive Concentric Ring Array in The Presence of Prior Knowledge

The partial adaptive CRA has been successfully applied to 3-D beamforming because of its flexibility, faster tracking ability and reduced computation with respect to the fully adaptive CRA. In some cases, prior knowledge regarding some interferences is available so that better beamformers can be designed. The previous method that exploits prior knowledge by using a fixed penalty factor could not guarantee in maintaining a low residual interference and noise level. We propose in this paper an adaptive beamformer that automatically seeks out the optimum penalty factor to achieve the best performance. The proposed beamformer outperforms the previous design in maintaining a higher output signal to interference and noise ratio, even after the characteristics of the interferences have changed. The performance of the proposed beamformer is evaluated through simulations.
4.1 Introduction

Array signal processing is widely used to perform spatial filtering of signals that arrive from different directions. The array geometry and the weights applied to each sensing element form a beampattern that maintains the SOI undistorted and attenuates interferences coming from other DOAs. In the case of 3-D beamforming, the ring array is preferable to the Linear Array because it reduces the DOA cone of uncertainty to just two DOAs and it maintains a uniform beampattern independent of the azimuth angle [9, 12]. Among RAs, the CRA structure is quite popular for its flexibility in adaptive beamforming, its suitability for nested array design in broad band applications, and its ability to achieve frequency invariant characteristics [12, 13].

When the DOAs and powers of the interferences are unknown, adaptive methods are required to find the array weights that minimize the interferences and background noise in the array output. The beamformer that adapts all the weights is referred as the fully adaptive () beamformer.

In practice for remote signal acquisition, the number of weights required to achieve a desired performance could be over several hundreds [9]. It is necessary to decrease the number of weights to adapt in order to reduce the computational complexity and improve the tracking ability in non-stationary environments.

A partial adaptive CRA called Type I was proposed by Li et al. [50] where each ring is considered as a sub-array that uses delay-and-sum weights [9]. The output of each sub-array is passed to an adaptive GSC [9] that forms the final output. Cox et al. applied a similar approach to linear arrays [49]. The Type I beamformer converges much faster than the fully adaptive beamformer, at the expense of decreasing the array’s DOFs and as a result, limiting its ability to eliminate larger number of interferences and reduce the amount of background noise. The number of interferences that Type I array is able
to cancel is one less than the number of rings. If the number of interferences present is larger than this limit, the array will not be able to perform satisfactorily.

Prior information has been exploited in array processing to improve performance [64]. In certain scenarios the DOAs and powers of some interferences are known or can be estimated \textit{a priori}. In such a case, Vicente \textit{et al.} [71] proposed the Modified Type I beamformer that uses the intra-ring weights to cancel the known interferences and leaves the adaptive GSC weights to remove the unknown interferences. The result is an increase in the total number of interferences that can be eliminated because it does not rely on the adaptive weights to cancel all interferences. As in the Type I array, each ring is considered as a sub-array that performs fixed beamforming. The fixed weights are chosen to eliminate the known interferences, but at the same time restricted to not deviate too much from the delay-and-sum weights. A fixed value penalty factor is used to control this amount of deviation [71].

The Modified Type I array achieves in most cases a lower residual error than the Type I array. However, in some scenarios the performance was worse than the Type I array. The Modified Type I array could suffer from high sidelobe levels because the intra-ring weights are found by ignoring the unknown interferences. Note that Modified Type I array is essentially a generalization of Type I array because it reduces to Type I array if the penalty factor is set to a very large value. Hence, increasing the penalty factor will improve the performance of Modified Type I array if it is worse than that of the Type I array. Given a localization scenario, an optimum penalty factor may exist that improves its performance, and will make it not worse than, if not better than Type I array. The paper proposes an automatic method to obtain the penalty factor value that achieves the minimum attainable residual error level for the Modified Type I beamformer.

Next section formulates the problem. Section 4.3 reviews the fully adaptive, Type I and Modified Type I beamformers. Section 4.4 proposes a method to find the penalty
factor that optimizes performance. Section 4.5 is the simulation and section 4.6 is the conclusion.

4.2 Problem Formulation

The CRA structure is shown in Fig. 4.1. It is composed of $M$ rings located in the $(x, y)$ plane where ring $m$ has $K_m$ elements and there is a total of $K = \sum_{m=1}^{M} K_m$ array elements. The array input vector with respect time $t$ is modeled for a far field narrow-band scenario with expression:

$$u(t) = s(t)s + \sum_{l=1}^{L} i_l(t)i_l + n(t),$$

(4.1)

where $s$ and $i_l$ are the steering vectors of the SOI and the $l^{th}$ interference respectively. $s(t)$ and $i_l(t)$ are the corresponding amplitudes. The isotropic Gaussian noise vector is represented by $n(t)$. $L$ is the total number of interferences where the first $C$ of them are assumed to have known/estimated DOAs and powers.

The $(K \times 1)$ SOI steering vector $s$ is composed of $M$ sub-vectors $s_m$ (steering vector...
for ring $m$). $s_m$ is $(K_m \times 1)$ and its $k^{th}$ element is equal to:

$$s_m(k) = e^{j2\pi(x_{m,k}\cos\phi_0+y_{m,k}\sin\phi_0)\sin\theta_0} \quad k = 1, ..., K_m \quad m = 1, ..., M,$$

(4.2)

where $\lambda$ is the wavelength, $(\phi_0, \theta_0)$ are the azimuth and polar angles of the SOI, and $(x_{m,k}, y_{m,k})$ is the location of the $k^{th}$ array element at ring $m$ in Cartesian coordinates. The interference steering vectors $i_l$ are in the same form as (4.2) and defined by replacing $(\theta_0, \phi_0)$ with $(\theta_l, \phi_l)$. The array output at time $t$ is:

$$z(t) = v^H u(t),$$

(4.3)

where $v$ is the weight vector and $()^H$ is the Hermitian transpose.

### 4.3 Background

The fully adaptive beamformer adaptively finds the weights $v$ by minimizing the instantaneous output power $|z(t)|^2$ subject to a distortionless constraint at the DOA of the SOI.

The Type I beamformer [50] was designed to reduce the computational cost of the fully adaptive beamformer and increase the convergence rate. It considers each ring as a sub-array where the intra-ring weight vector for ring $m$ is the delay-and-sum $h_m = s_m/K_m$. One may consider choosing $h_m$ steered towards the interferences. However, we do not know the DOAs of all interferences in the problem at hand. The output of ring $m$ is $y_m(t) = h_m^H u_m(t)$ where $u_m(t)$ is the input signal vector of ring $m$. The output of each ring is collected as the vector:

$$y(t) = [y_1(t), y_2(t), ..., y_m(t), ..., y_M(t)]^T.$$

(4.4)

The vector $y(t)$ is then passed to an adaptive NLMS-GSC algorithm that forms the final output as:

$$z(t) = w_{gsc}^H y(t),$$

(4.5)
where $w_{gsc}$ is the overall GSC weight vector that is obtained adaptively by minimizing the instantaneous array output power $|z(t)|^2$ subject to a distortionless constraint at the DOA of the SOI.

The Modified Type I beamformer [71] was created to utilize the prior knowledge available for some interferences and increase the number of interferences that can be canceled with respect to the Type I array. This beamformer also considers each ring as a sub-array, but in this case the intra-ring weight vector for ring $m$, now called $g_m$, is obtained by minimizing the cost function:

$$J = \left\{ g_m^H R_{mc} g_m + \alpha ||g_m - h_m||^2 \right\}$$

subject to the constraint $g_m^H s_m = 1$, where $R_{mc}$ is the correlation matrix of ring $m$ obtained from the DOAs and powers available from the $C$ interferences, and $\alpha$ is the penalty factor that limits the deviation of $g_m$ from the delay-and-sum weights $h_m$. The larger the $\alpha$, the less deviation from delay-and-sum. The solution is:

$$g_m = \frac{[R_{mc} + \alpha I]^{-1} s_m}{s_m^H [R_{mc} + \alpha I]^{-1} s_m}. \quad (4.7)$$

The output of ring $m$ is now $y_m(t) = g_m^H u_m(t)$ and the array output is obtained using (4.4) and (4.5), where $w_{gsc}$ is found in the same manner as in Type I. The penalty factor used was $\alpha = 0.01 / \text{tr}(R_{mc})$. It is interesting to notice that the Type I array is a particular case of the Modified Type I array for large values of $\alpha$. That is, $\lim_{\alpha \to \infty} g_m = h_m$ in (4.7) and the Modified Type I becomes the Type I array. (4.7) has a Diagonal Loading (DL) form [9]. However, unlike the typical DL, (4.7) is not applied to the whole array weight vector but to each ring weight vector on an individual basis. Also, $\alpha$ is not applied to the data correlation matrix, but on a $R_{mc}$ built from the signal and known interferences only.
4.4 Optimizing the Performance Through the Penalty Factor

Although the Modified Type I array performs better than the Type I array in most cases, sometimes it has larger steady-state residual error. This is because the Modified Type I array finds the intra-ring weights $g_m$ based only on the interferences with prior knowledge and ignore the unknown ones. As a result, if the unknown interferences happen to be in the sidelobe of the beampattern created by the intra-ring weights, it would be difficult for the adaptive weights to remove the unknown interferences resulting in a larger residual error. In addition, the sidelobes created by $g_m$ could also increase the amount of the background noise that leaks through the beamformer to the output.

Since the Modified Type I array becomes the Type I array when $\alpha$ is very large, we can improve its performance by finding the optimum $\alpha$ that achieves the smallest steady-state residual error in any localization scenario. We then use the optimum $\alpha$ to obtain the intra-ring weights (4.7).

A physical interpretation of $\alpha$ with respect to the available prior knowledge follows. When none prior knowledge is available the optimum $\alpha$ value will be infinite, which makes (4.7) become a delay-and-sum intra-ring weight vector. When all the interferences are known, the optimum $\alpha$ will be zero, resulting in the MVDR solution. It is important
to note that a single $\alpha$ is used for all rings. Selén et al. [65] proposed an approach to automatically estimate a similar parameter in a fully adaptive beamformer. Applying Selén’s method in our case would yield different $\alpha$ values from different rings and does not guarantee the entire array output has minimum residual error.

Fig. 4.2 shows the theoretical steady-state residual error vs. the penalty factor for two scenarios that will be explained in the simulations. Although scenario 1 has better performance when $\alpha$ is large and scenario 2 gives better result when $\alpha$ is small, the residual errors in both cases are smallest at some $\alpha$ values that are neither too small nor too large.

For a given $\alpha$ we find $g_m$ from (4.7) and hence the output of ring $m$, $y_m(t)$. The input to the GSC is $y(t)$ and hence the weight vector of the GSC, $w_{gsc}$, is related to the penalty factor $\alpha$ in a complicated manner. When both $\alpha$ and $w_{gsc}$ are found through simultaneous adaptation of them, it is realized that typical adaptive algorithms like NLMS do not converge due to the complex nature of the error surface. We propose here to use the steady-state solution of $w_{gsc}$ instead.

The block diagram of the proposed beamformer is shown in Fig. 4.3. It is composed of $M$ intra-ring weight vectors $g_m$ that are applied to the input vectors $u_m$. For notation simplicity, we define the $(K \times M)$ partition matrix as:

$$
P = \begin{bmatrix}
g_1 & \cdots & \cdots & \cdots & 0_{(N_1 \times 1)} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0_{(K_m \times 1)} & \ddots & \ddots & \ddots & g_M
\end{bmatrix}.
$$

(4.8)

$P$ is dependent on $\alpha$ through the intra-ring vectors $g_m$ obtained from (4.7). Now, we can write the GSC input $y(t) = P^H u(t)$. The upper branch in the GSC uses the quiescent weights $w_q$ to form the quiescent response $z_q(t)$, and

$$
w_q = (P^H P)^{-1} P^H \hat{v}_q.
$$

(4.9) is the Least-Squares solution of $P w_q = \hat{v}_q$, where $\hat{v}_q$ is a desired quiescent weight.
vector (e.g. delay-and-sum). The lower branch has a blocking matrix \( B \) and the weight vector \( \mathbf{w}_{a,ss} \) to produce \( z_a(t) \). \( B \), of dimensions \( M \times (M - 1) \), is designed to eliminate the SOI component from \( y(t) \) and is obtained from the null space of the SOI. The weight vector \( \mathbf{w}_{a,ss} \) is:

\[
\mathbf{w}_{a,ss} = ((\mathbf{PB})^H \mathbf{R}_u (\mathbf{PB}))^{-1}\mathbf{Bw}_q, \tag{4.10}
\]

where \( \mathbf{R}_u \) is the input signal correlation matrix. (4.10) is the solution to the problem of minimizing the output power subject to a distortionless response at the DOA of the SOI [9]. The array output \( z(t) \) is \( z_q(t) - z_a(t) \). The GSC overall weight vector is:

\[
\mathbf{w}_{gsc} = \mathbf{w}_q - \mathbf{Bw}_{a,ss}. \tag{4.11}
\]

Using (4.5) with \( y(t) = \mathbf{P}^H \mathbf{u}(t) \), the array output at the discrete instant \( t = dT \) can be written as:

\[
z(t) = (\mathbf{Pw}_{gsc})^H \mathbf{u}(t), \tag{4.12}
\]

where \( T \) is the sampling period and \( d \) is the snapshot number.

To find (4.10) we need to estimate \( \mathbf{R}_u \) from the data. We propose three methods to do it. The first one (M1) obtains the estimate from non-overlapping blocks of \( D \) snapshots as:

\[
\hat{\mathbf{R}}_u = \frac{1}{D} \sum_{d=1}^{D} \mathbf{u}(d)\mathbf{u}^H(d). \tag{4.13}
\]

Here we use (4.13) to update \( \mathbf{w}_{a,ss} \) after each \( D \) snapshots. The second method (M2) uses overlapping blocks of data recursively as:

\[
\hat{\mathbf{R}}_u(D + d) = \hat{\mathbf{R}}_u(D + d - 1) + [\mathbf{u}(D + d)\mathbf{u}^H(D + d) - \mathbf{u}(d - 1)\mathbf{u}^H(d - 1)]/D. \tag{4.14}
\]

A third modification (M3) uses an exponentially weighted sample correlation matrix as in the RLS algorithm [9]:

\[
\hat{\mathbf{R}}_u(d) = \rho \hat{\mathbf{R}}_u(d - 1) + (1 - \rho)\mathbf{u}(d)\mathbf{u}^H(d), \tag{4.15}
\]
where $\rho$ is the forgetting factor. M2 and M3 use $\hat{R}_u$ to update $w_{a,ss}$ after each snapshot.

Because M1 only needs to update the $w_{gsc}$ weights each $D$ snapshots, it is computationally effective, but is not appropriate to handle the changes in interference characteristics. M2 and M3 updates $w_{gsc}$ after each snapshot, so it is computationally more expensive but is able to perform much better for changes in the characteristics of the interferences. The partial adaptive structure $P$ allows the reduction of computation over the fully adaptive method. One can notice that (4.10) involves an inversion of a much smaller $(M-1 \times M-1)$ matrix, compared to the inversion of a $(K-1 \times K-1)$ matrix if we were using the fully adaptive GSC array.

### 4.4.1 Obtaining the Optimum Penalty Factor

To find the optimum $\alpha$ that achieves the smallest steady-state residual error, we require an expression of the steady-state residual error as a function of $\alpha$. Ideally, the residual error is only obtained when the SOI is absent. When the SOI is present, the output power contains both the residual error and the power of the SOI. We will use the output power instead of the residual error to find the initial $\alpha$ because they differ only in a constant value equal to the power of the SOI.

An estimate of the array output power is:

$$\hat{P}_{z,ss} = (Pw_{gsc})^H \hat{R}_a (Pw_{gsc}).$$  \hspace{1cm} (4.16)

(4.16) relates the output power to the penalty factor $\alpha$ through $P$ and $w_{gsc}$. It has a global minimum with respect to $\alpha$ but its surface is not always unimodal (e.g. see Fig. 4.2). To find the minimum of (4.16), we shall use a gradient search algorithm that starts in the vicinity of the global minimum of $\hat{P}_{z,ss}$. A coarse grid search is applied to obtain an initial value of $\alpha$. This initialization will only be needed in the first block at the beginning of the process. For subsequent processing blocks we use the optimum $\alpha$ from the previous block as the initial $\alpha$. 
Since $\alpha$ can take on very small values, the gradient search is implemented using the Newton method over $\beta = \log \alpha$ with a damping factor of 0.5:

$$
\beta^{(r+1)} = \beta^{(r)} - \frac{0.5}{\nabla^2_\beta \hat{P}_{z,ss}} \nabla_\beta \hat{P}_{z,ss},
$$

(4.17)

where $r$ is the iteration number. The first and second derivatives $\nabla_\beta \hat{P}_{z,ss}$ and $\nabla^2_\beta \hat{P}_{z,ss}$ are found by using linear approximations.

### 4.4.2 Algorithm

When the beamformer uses M1 to generate $\hat{\textbf{R}}_u$, the algorithm is implemented by employing data blocks of length $D$. Once the first block of data is collected (i.e. $d = D$), several values of $\alpha$ in the range $(10^{-5}, 10^5)$ are used to find the respective steady-state output power values using (4.7)-(4.11), (4.13), and (4.16). The value of $\alpha$ that achieves the smallest $\hat{P}_{z,ss}$ is selected as the initial value to start the Newton algorithm given in (4.17). After convergence, the optimum $\alpha$ is used to find the intra-ring weights $g_m$ from (4.7), the partition matrix $\textbf{P}$ from (4.8), and the GSC weights $w_{gsc}$ from (4.13),(4.9)-(4.11). For subsequent data blocks, the array weights ($g_m$ and $w_{gsc}$) are updated in the same manner, but the initial $\alpha$ value is now obtained from the optimum $\alpha$ solution of the previous block. Because the proposed method cannot obtain an initial $\alpha$ value until the first $D$ snapshots are collected, we will use a fixed large value of $\alpha$ to generate the
array output when $d < D$.

When M2 and M3 are used to generate $\hat{R}_u$, the optimum $\alpha$, $P$, and $w_{gsc}$ are determined using the same approach as in M1. The difference is that (4.13) is replaced by (4.14) or (4.15) for M2 or M3 respectively, and that the optimum $\alpha$, $P$, and $w_{gsc}$ are updated after each snapshot.

### 4.5 Simulations and Results

The simulation example uses a CRA that has 68 elements arranged in 4 rings. The elements are uniformly spaced in each ring, and the number of elements from the innermost ring is 12, 12, 20, and 24. The number of ensemble averages is 100. Our results assume no steering vector mismatch or calibration errors. In practice those errors are present and their effect on the proposed technique is a subject for further investigation.

**Scenario 1** considers a narrowband 1kHz input composed of the SOI arriving from a DOA $(\theta_0 = 90^\circ, \phi_0 = 0^\circ)$ with a $SNR$ of 0dB, and four interferences arriving from DOAs $(\theta_1 = 65^\circ, \phi_1 = 230^\circ)$, $(\theta_2 = 75^\circ, \phi_2 = 120^\circ)$, $(\theta_3 = 60^\circ, \phi_3 = 160^\circ)$, and $(\theta_4 = 75^\circ, \phi_4 = 283^\circ)$ with signal to interference ratio (SIRs) equal to $-30$dB, $-25$dB, $-35$dB, and $-30$dB. The DOA $(\theta_1, \phi_1)$ and SIR of the first interference are known *a priori*. The background noise is Gaussian distributed and isotropic. The results are shown in Fig. 4.4. The three methods that estimate $R_u$ of the proposed beamformer perform similarly (upper plot). The output signal to interference and noise ratios ($SINR_0$) of them are several dBs higher (at least 7.5dB at iteration 3500) than those of Type I and Modified Type I arrays (lower plot), where they are still adapting and far away from their respective steady-state output $SINRs$.

**Scenario 2** examines the behavior of the proposed beamformer under changes in the interference characteristics. The SOI has DOA $(\theta_0 = 90^\circ, \phi_0 = 0^\circ)$ and 0dB $SNR$. The four interferences arrive from DOAs $(\theta_1 = 85^\circ, \phi_1 = 230^\circ)$, $(\theta_2 = 90^\circ, \phi_2 = 120^\circ)$,
(θ₃ = 70°, φ₃ = 150°), and (θ₄ = 70°, φ₄ = 283°) with SIRs −30dB, −25dB, −35dB, and −30dB. The DOA (ϕ₁, θ₁) and SIR of the first interference are known a priori. The fourth interference suddenly changes its characteristics at snapshot number d = 1500 to a different DOA of (θ₄ = 65°, φ₄ = 278°) and SIR of −40dB. The results are shown in Fig. 4.5. When d < 1500, M1, M2, and M3 perform comparably in achieving high output SINRs (upper plot). For the same period of time, the Type I and the Modified Type I arrays (lower plot) are still adapting and their output SINRs are at least 5dB below than those of the proposed beamformer. At snapshot d = 1500, the characteristics of the fourth interference change and the effect is noticeable in all beamformers. For M1 the SINR₀ falls to unacceptable low values until the weights are calculated from a block that is entirely composed of the snapshots that have new interference characteristics. For M2 and M3 the effect is a sudden drop of the output SINRs. Then, the output SINRs gradually increase as the Ĝₜ is recursively updated. Type I and Modified Type I arrays respond to the change similarly as M2 and M3. As time proceeds, their output SINRs quickly recover but always being lower (at least 3dB at iteration 3500) than those of the proposed method.

4.6 Conclusion

In this paper we have proposed a method to improve the performance of the Modified Type I partial adaptive CRA through the optimization of the penalty factor α, which achieves the maximum attainable output SINR. In the proposed beamformer, α is automatically chosen to allow an appropriate amount of prior knowledge used in the design of the beamformer weights. The result is an array with reduced sidelobe levels and higher output SINR with respect to the previously proposed beamformer.
Figure 4.4: Scenario 1: output $SINR$ vs. iterations for the proposed beamformer (upper plot) and the previous beamformers (lower plot).

Figure 4.5: Scenario 2: output $SINR$ vs. iterations for the proposed beamformer (upper plot) and the previous beamformers (lower plot).
Chapter 5

Combined Beamspace and Element Space Technique for Partial Adaptive Concentric Ring Array

Partial adaptive CRA are very attractive for beamforming in 3-D, because they substantially reduce the computation time and improve tracking ability with respect to a fully adaptive CRA. In some practical situations the impinging signal contains some interferences whose characteristics such as DOAs could be estimated *a priori*. Previous partial adaptive CRA methods that utilize prior knowledge were not able to always maintain a low residual interference and noise level in the beamformer output. Even when they could, the performance degrades quickly if the estimated characteristics of the interferences contain errors. In this paper we propose a combined beamspace and element space (CBSES) partial adaptive CRA that is able to maintain a low residual interference and noise level after beamforming, and at the same time, is robust under uncertainties in the estimated characteristics (DOAs) of some of the interferences.

5.1 Introduction

The use of arrays in signal processing is well known for its ability to separate uncorrelated signals with similar frequency contents impinging from different DOAs, and at the same
time attenuating the isotropic background noise. In particular, CRA is found particularly useful in 3-D beamforming for its ability to eliminate DOA ambiguities inherent in the ULA [9]. CRA can also be designed to provide frequency invariant characteristics for broad-band applications [14].

The number and characteristics of the interferences present are not completely known a priori, forcing the use of adaptive methods to find the beamformer coefficients (weights) associated with each array element. Quite often, the weights are found by minimizing the beamformer output power, subject to a set of constraints including, but not limited to, unity gain at the DOA of the SOI.

In adaptive broad-band beamforming, we can directly adapt the filter coefficients for each individual sensor element signal, or we can decompose the received signal into many narrow-band components and apply an adaptive narrow-band beamformer for each individual component. When the time-window to apply FFT is sufficiently large, the second approach will give similar steady state results as the first [26] but improve convergence speed. In this paper, we focus on the second approach. In such a case, broad-band beamforming reduces to $N/2$ narrow-band beamforming where $N$ is the number of FFT bins. Each narrow-band beamformer has $K$ weights that are complex, where $K$ is the number of array elements.

Broad-band beamforming increases the number of weighting coefficients and computational cost with respect to a narrow-band beamformer. In addition, in order to achieve a fine angular resolution and a strong amount of noise reduction, the number of array elements required is huge, and can be in the order of several hundreds [14]. Consequently, there is a large number of weights to adapt, which will result in high computational cost, low convergence speed and poor tracking performance in a non-stationary environment. Partial adaptation methods are effective to reduce the deficiencies from the adaptation of hundreds of weights.
In a previous work Li & Ho [51] proposed an element space partial adaptive beamformer called Type I array, where each ring is considered as a sub-array that performs conventional beamforming using delay-and-sum weights [9]. The output of each ring is combined with adaptive weights to form the final output. This method substantially reduces the number of adaptive weights with respect to the fully adaptive array, leading to a much faster convergence and better tracking. The steady state error is only slightly larger than that in the fully adaptive array. Although Type I partial adaptive beamformer improves convergence by using a reduced number of adaptive weights, it limits the number of interferences that can be canceled. The Type I beamformer will not be able to cancel all interferences if the number of interferences exceeds the DOFs provided by the partial adaptive beamformer, which is given by the number of adaptive weights minus the number of constraints.

In practice, the DOAs of some interferences may be available. For instance, they can be estimated through DOA techniques before the desired signal appears. Several previous works [55, 64, 72] have used the prior DOA knowledge to improve the robustness of the beamformer. In partial adaptation, Vicente & Ho [71] proposed the Modified Type I array that uses the prior knowledge of some interferences to obtain the intra-ring weights by using a modified version of MVDR [25] approach that ignores the interferences with unknown DOAs. The beamformer is able to cancel the interferences with known DOAs within intra-ring without reducing the number of DOF, and effectively increase the total number of interferences that can be canceled. However, we found that the overall beampattern sometimes suffers from high sidelobe levels because the MVDR is within individual rings and the interferences with unknown DOAs are ignored. This leads to larger steady state error than in the Type I array. Also, if the interferences with known DOAs are not estimated accurately, the obtained intra-ring weights are not able to cancel effectively those interferences, causing a dramatic increase in the steady state
error.

In addition to element space, partial adaptive array can also be derived using beamspace techniques [9, 31]. The element space approach has the advantage of limiting the number of adaptive elements if the DOAs of the interferences are not known. On the other hand, the beamspace approach is particularly effective if the DOAs of the interferences are available; so that we can form beams steered toward them, and cancel them. In the problem at hand, some interferences have known DOAs and others do not. We therefore propose a CBSES approach to develop a partial adaptive beamformer, where the element space part, analogous to the Type I array, takes care of the interferences with unknown DOAs whilst the beamspace component handles the interferences with known DOAs.

The CBSES processing is applied to the array input signal, which is then fed to a GSC [73] that adaptively eliminates all interferences and generates the final output. On one hand, the GSC makes efficient use of the beamspace part to find a set of adaptive weights that eliminate those interferences of known DOAs. On the other hand, the available adaptive weights associated with the element space component are adaptively shaped to cancel the interferences with unknown DOAs.

The proposed beamformer maintains the Type I array structure that has small side-lobe levels, and at the same time makes use of the prior knowledge to remove interferences that have known DOAs. The proposed beamformer is found to be robust with respect to uncertainties in the prior knowledge about the DOAs of some interferences, and is more attractive for practical applications.

The rest of the paper is organized as follows. Section 5.2 is a review of Type I and modified Type I beamformers. Section 5.3 introduces the proposed CBSES beamformer. Section 5.4 contains the simulations and results. Conclusions are shown in Section 5.5 and acknowledgments in section 5.6.
5.2 Background

The CRA configuration is shown in Fig. 5.1. It is composed of a total of $K$ elements arranged in $M$ rings located in the plane $z = 0$. The number of elements in ring $m$ is $K_m$, and

$$K = K_1 + K_2 + \ldots + K_m + \ldots + K_M. \quad (5.1)$$

The output of the array at any time $t$ is:

$$z(t) = \sum_{m=1}^{M} \sum_{k=1}^{K_m} v_{mk}^* u_{mk}(t) = v^H u(t), \quad (5.2)$$

where $u_{mk}(t)$ is the signal received at the $k^{\text{th}}$ element of ring $m$, $v_{mk}$ is the weight associated to each element, $(\cdot)^H$ represents complex conjugation transpose, and $(\cdot)^*$ represents complex conjugation. $v$ and $u$ represent the compact vector notation of the weights and input signals respectively.

Under narrow-band input and far-field source assumptions, the received input signal
vector is modeled as:

$$\mathbf{u}(t) = s(t)\mathbf{s} + \sum_{c=1}^{C} i_c(t)\mathbf{i}_c + \sum_{l=1}^{L} i_l(t)\mathbf{i}_l + n(t)\mathbf{n}. \quad (5.3)$$

where \(s(t), i_c(t), i_l(t),\) and \(n(t)\) are the complex amplitudes of the SOI, interferences with prior DOA knowledge, interferences without prior DOA knowledge, and isotropic Gaussian noise signals respectively. \(\mathbf{s}, \mathbf{i}_c,\) and \(\mathbf{i}_l\) are the corresponding steering vectors. The components of the noise vector \(\mathbf{n}\) are random and spatially uncorrelated. \(C\) and \(L\) are the number of interferences with known and unknown DOAs respectively. The SOI array steering vector \(\mathbf{s}\) is a \(K \times 1\) vector whose elements are:

$$s_{m,k} = e^{j\frac{2\pi}{\lambda}(x_{mk}\cos\phi_0 + y_{mk}\sin\phi_0)\sin\theta_0} \quad m = 1, \ldots, M, \quad k = 1, \ldots, K_m, \quad (5.4)$$

where \(\lambda\) is the wavelength, \((x_{mk}, y_{mk})\) is the location of the array element \((m, k)\) in Cartesian coordinates, and \((\theta_0, \phi_0)\) are the polar and azimuth angles of the SOI. The interference steering vectors \(\mathbf{i}_c, \mathbf{i}_l\) are in the same form as (5.4) by replacing \((\phi_0, \theta_0)\) with \((\phi_c, \theta_c)\), and \((\phi_l, \theta_l)\) respectively.

A fully adaptive array finds the weight coefficients \(v_{mk}\) by minimizing the output power \(|z(t)|^2\) subject to a set of constraints.

### 5.2.1 Type I Partial Adaptive Beamformer

The Type I partial adaptive array proposed by Li & Ho [51] considers each ring in the array as a sub-array of \(K_m\) elements. The output of ring \(m\) is:

$$y_m(t) = \tilde{\mathbf{h}}_m^H\mathbf{u}_m(t), \quad (5.5)$$

where \(\mathbf{u}_m(t)\) is the received signal vector of ring \(m\), and \(\tilde{\mathbf{h}}_m\) is the vector containing the delay-and-sum weights of ring \(m\) defined as:

$$\tilde{h}_{m,k}(t) = \frac{1}{K_m}s_{m,k} \quad k = 1, \ldots, K_m. \quad (5.6)$$
Let
\[ y(t) = [y_1(t), y_2(t), \ldots, y_m(t), \ldots, y_M(t)]^T. \] (5.7)

The output of each ring is combined using \( w \) to obtain the final output as:
\[ z(t) = w^H y(t). \] (5.8)

The weight vector \( w \) is found adaptively, subject to some linear constraints.

### 5.2.2 Modified Type I Partial Adaptive Beamformer

Proposed by Vicente & Ho [71], the Modified Type I array replaces the intra-ring weights \( \tilde{h}_m \) in (5.5) by \( g_m \) defined as:
\[ g_m = \frac{[(1 - \alpha) R_{m,i}^e + \alpha I]^{-1} \tilde{h}_m/K_m}{\tilde{h}_m^H [(1 - \alpha) R_{m,i}^e + \alpha I]^{-1} \tilde{h}_m}. \] (5.9)

where \( \alpha \) is the penalty term, \( R_{m,i}^e \) is the correlation matrix of the interferences with known DOAs and power, and \( I \) is the \((K_m \times K_m)\) identity matrix. \( g_m \) is obtained by the use of MVDR, and at the same time by limiting deviation from delay-and-sum. The penalty term provides a tradeoff between them. The output of each ring,
\[ y_m(t) = g_m^H u_m \] (5.10)
is multiplied by the adaptive weight vector \( w \) to form the final output as in (5.8).

### 5.3 CBSES Partial Adaptive Beamformer

The Modified Type I adaptive array does not yield in all cases a smaller steady state error than the Type I adaptive array. There are some instances where the overall beampattern suffers from high sidelobe levels, causing an unacceptable amount of interference and noise in the beamformer output. This is a consequence of the constraint that forces the intra-ring output to be zero for the interferences of known DOAs, and ignores the
interferences of unknown DOAs when designing the intra-ring weights. The isotropic noise and other interferences could leak through the higher sidelobes, resulting in larger steady state error.

Another disadvantage of the Modified Type I array is that it requires the exact knowledge of the DOAs as well as the strength of the interferences in order to design the intra-ring weights $g_m$ to cancel them. In practice, they are estimated and will not be known exactly, and as a result, the Modified Type I array will have degraded performance. There is a need to derive another adaptive system that is able, in the presence of estimation errors, to maintain at least the performance of the Type I array, and effectively eliminate the interferences even though their DOAs are not exactly known.

Apart from the element space approach to reduce the number of adaptive elements as in the Type I array, beamspace is another alternative. The beamspace partial adaptive method uses beams steered to the SOI and the interferences [9], and transforms the input vector to a lower dimension space for processing. Quite often a beamspace beamformer can maintain a steady state error level similar to that of the fully adaptive beamformer. When the DOAs of all the interferences are not known, it is necessary to have a sufficient number of beams to cover all possible 3-D directions. A partial adaptive beamformer using beamspace alone will require too many weights to adapt.
The proposed CBSES beamformer is a combination of element space and beamspace, which will take the advantage of both methods. The proposed array will use the advantage of element space for performing conventional beamforming in each ring to reduce the isotropic noise power to a minimum. It will also use beamspace beams steered towards the known DOAs of the interferences to cancel them effectively through adaptation. The interferences with unknown DOAs, will be reduced adaptively using the DOF available from the adaptive weights assigned to the element space.

We shall formulate the CBSES technique using a partition matrix. The partition matrix of the proposed method is a $K \times (M + C)$ sparse matrix given by:

$$
P_b = [h_1 \cdots h_m \cdots h_M | b_1 \cdots b_c \cdots b_C].$$

The first $M$ columns represent the element space part and is formed by the vectors $h_m$. Each vector $h_m$ is composed of $K_m$ delay-and-sum weights from ring $m$ and $(K - K_m)$ zeros arranged in such a way that $y_m(t) = h_m^H u(t)$ is the same as the $m^{th}$ ring output (5.5) in the Type I array. The remaining $C$ columns represent the beamspace part and it is formed by the vectors $b_c$. Each vector $b_c$ is composed of a beam steered to each of the $C$ known DOAs of the interferences, and satisfies $b_c^H i_c = 1$. The elements of $b_c$ are defined as:

$$b_{c,mk} = \frac{1}{K} e^{j \frac{2\pi}{\lambda}(x_{mk} \cos \phi_c + y_{mk} \sin \phi_c) \sin \theta_c} m = 1, ..., M \quad k = 1, ..., K_m.$$  \hfill (5.12)

The partition matrix is applied to the array input vector to form the reduced element signal vector:

$$y_b(t) = P_b^H u(t),$$

where $y_b(t)$ is a $(M + C) \times 1$ signal vector that contains beamspace and element space array input signal. The partitioned signal vector $y_b(t)$ is processed by a GSC to obtain the final output $z(t)$. 

91
The block diagram of the proposed adaptive beamformer is shown in Fig. 5.2. The GSC structure has two branches. The first is the quiescent branch that performs fixed spatial filtering. The second is the adaptive branch that performs unconstrained optimization.

The quiescent branch is not adaptive. It produces the response $z_q(t)$, called quiescent response, by multiplying the partitioned signal vector $y_b(t)$ with the quiescent weights as:

$$z_q(t) = w_q^H y_b(t).$$  

The quiescent weights are chosen as:

$$w_q = (P_b^H P_b)^{-1} P_b^H s/K.$$  

The adaptive branch is formed by a blocking matrix $B$ and a vector of adaptive weights $w_a$. The blocking matrix has a size of $((M + C) \times (M + C - Q))$. Its purpose is to eliminate the signal component from the partitioned signal vector $y_b(t)$. It is created from the null space of the constraints where $Q$ is the number of constraints. The signal after the blocking matrix, is multiplied by the adaptive weights to form $z_a(t)$,

$$z_a(t) = w_a^H (B^H y_b(t)).$$  

The interferences are estimated by the adaptive weights in the adaptive branch; and it is subtracted from the quiescent response to generate the final output as:

$$z(t) = z_q(t) - z_a(t).$$  

The adaptive weights are found iteratively by an adaptive algorithm, such as NLMS [9], that minimizes the instantaneous output power $|z(t)|^2$.

5.3.1 Analysis

We now analyze the steady state performance of the proposed adaptive array. At steady state, the adaptive weights will converge to the optimum weights $w_{a, opt}$. The equivalent
optimum weight vector in the GSC with input $y_b(t)$ and output $z(t)$ is [9]:

$$w_{opt} = w_q - Bw_{a,opt}. \quad (5.18)$$

The theoretical residual interference and noise power at steady state, or simply called steady state residual error, is [9]:

$$P_{in,ss} = w_{opt}^H P_b^H R_{in} P_b w_{opt}, \quad (5.19)$$

where $R_{in}$ is the correlation matrix of the interferences plus noise.

The optimum weights can be theoretically found for the CBSES partial adaptive array as [9]:

$$w_{opt} = \frac{(P_b^H R_{in} P_b)^{-1} P_b^H s}{s^H P_b (P_b^H R_{in} P_b)^{-1} P_b^H s}. \quad (5.20)$$

Putting (5.20) into (5.19) and simplifying gives:

$$P_{in,ss} = \left(s^H P_b (P_b^H R_{in} P_b)^{-1} P_b^H s\right)^{-1}. \quad (5.21)$$

This expression can be used to evaluate the steady state residual error for any partial adaptive array with a particular partition matrix $P_b$. It will give the result for the fully adaptive case if $P_b$ is equal to an identity matrix. We will use this formula in the simulations section to validate our results.

### 5.4 Simulations and Results

To demonstrate the performance of the proposed CBSES beamformer and to compare results with those of Type I and Modified Type I arrays, we implemented a simulation example for the processing of a narrow-band component of 1kHz signal.

The input signal is generated with a computer, and is formed by the SOI, four interferences, and background random noise. The SOI has a signal to noise ratio ($SNR$) of 0dB, and a DOA of ($\theta_0 = 90^\circ, \phi_0 = 0^\circ$). There are four interferences coming at
Figure 5.3: Beampattern and residual error level vs. iterations of Fully Adaptive, Type I, Modified Type I, and CBSES arrays for scenario A.

Figure 5.4: Residual error level vs. iterations of Fully Adaptive, Type I, Modified Type I, and CBSES arrays for scenario B (upper plot) and scenario C (lower plot).
DOAs of \((\theta_1 = 75^\circ, \phi_1 = 120^\circ)\), \((\theta_2 = 90^\circ, \phi_2 = 150^\circ)\), \((\theta_3 = 80^\circ, \phi_3 = 220^\circ)\), and \((\theta_4 = 60^\circ, \phi_4 = 283^\circ)\). The signal to interference ratios (SIRs) are -25dB, -35dB, -30dB, and -30dB respectively, and the background random noise is Gaussian and isotropic. We will consider two different scenarios (A and B) where only one DOA of the interferences is known, and a third scenario (C), where there is non-negligible estimation error in the DOA of an interference.

The beamformer has a total of 68 elements arranged in 4 rings. The number of elements per ring from the innermost is 12, 12, 20, and 24. The partition matrix \(\mathbf{P}_b\) has five columns. The first four columns contain the element space part and is composed of four sparse vectors \(\mathbf{h}_1\) to \(\mathbf{h}_4\). The fifth column is implemented by a beam \(\mathbf{b}_1\) steered to the interference with a known DOA. The theoretical steady state residual error is found from (5.21). To validate the theoretical results we have implemented a NLMS algorithm to find the adaptive weights \(\mathbf{w}_a\), the equivalent GSC weights:

\[
\mathbf{w} = \mathbf{w}_q - \mathbf{Bw}_a,
\]

and the residual error:

\[
P_m = \mathbf{w}^H \mathbf{P}_b^H \mathbf{R}_m \mathbf{P}_b \mathbf{w}
\]

along 20,000 iterations. The number of ensemble averages is 100. The adaptive algorithm minimizes the beamformer output power subject to a linear constraint of unity gain at the DOA of the SOI.

The upper plot of Fig. 5.3 shows the theoretical beampattern of scenario A using (5.20), where an interference DOA \((\theta_1 = 75^\circ, \phi_1 = 120^\circ)\) is known. From the figure we see that Type I (narrow trace) is able to force nulls at only three of the four interferences (vertical lines) because its DOF is less than the number of interferences. Modified Type I (dashed trace) is able to cancel all interferences, but the increase in the sidelobe level is dramatic. The CBSES array (bold trace) is able to cancel all interferences and maintain
lower sidelobe levels similar to that of the Type I array. The theoretical values of the residual error at steady state computed using (5.21) are shown in the last row of Table 5.1. Their values are consistent with the findings in the beampattern.

The lower plot of Fig. 5.3 shows the convergence behavior of the residual interference and noise power vs. the number of iterations in the same scenario A. The four traces in the figure represent the fully adaptive, Type I, Modified Type I, and CBSES array. Type I (narrow trace) shows a large residual error after initial convergence because it is not able to cancel all interferences. Modified Type I (dashed trace) array performs worse than Type I in both convergence speed and residual error as a consequence of having higher sidelobe levels. However, the CBSES approach (bold trace) has a much smaller residual error level than the other partial adaptive arrays. The convergence speed is superior to that of Modified Type I and similar to that of Type I. The fully adaptive array trace is shown for reference as the lowest attainable residual level after adaptation.

Table 5.1 shows the residual error values at different iterations for scenario A. The CBSES array residual error levels are always smaller than those of the Modified Type I array and of the Type I array.

The second scenario B has a different interference with a known DOA of \((\theta_3 = 80^\circ, \phi_3 = 220^\circ)\). The upper plot of Fig. 5.4 shows the convergence behavior of the residual error vs. the number of iterations. The Type I array shows the same behavior as in scenario A, because it does not use prior knowledge. The Modified Type I array is able to attain slightly lower residual error levels than the CBSES array because in this particular case the beampattern happens not to suffer from high sidelobes as in the previous scenario. The CBSES array shows similar residual error behavior than in scenario A. Table 5.2 shows the steady state residual error values.

Finally, to show the consistent behavior and robustness of this design versus the Modified Type I array under uncertainty in the DOAs of some interferences, we simulated
Table 5.1: Comparison of residual error levels. Scenario A

<table>
<thead>
<tr>
<th>Iterat.</th>
<th>Fully</th>
<th>Type I</th>
<th>Mod. Type I</th>
<th>CBSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.4795</td>
<td>4.6266</td>
<td>11.3244</td>
<td>3.7599</td>
</tr>
<tr>
<td>50</td>
<td>4.3508</td>
<td>3.3778</td>
<td>9.9708</td>
<td>2.8982</td>
</tr>
<tr>
<td>1200</td>
<td>0.0459</td>
<td>0.1574</td>
<td>0.4700</td>
<td>0.1298</td>
</tr>
<tr>
<td>4000</td>
<td>0.0208</td>
<td>0.1115</td>
<td>0.1499</td>
<td>0.0436</td>
</tr>
<tr>
<td>8000</td>
<td>0.0173</td>
<td>0.1051</td>
<td>0.1307</td>
<td>0.0361</td>
</tr>
<tr>
<td>12000</td>
<td>0.0174</td>
<td>0.1048</td>
<td>0.1247</td>
<td>0.0353</td>
</tr>
<tr>
<td>20000</td>
<td>0.0175</td>
<td>0.1047</td>
<td>0.1188</td>
<td>0.0356</td>
</tr>
<tr>
<td>∞</td>
<td>0.0149</td>
<td>0.1013</td>
<td>0.1125</td>
<td>0.0295</td>
</tr>
</tbody>
</table>

Table 5.2: Steady state residual error levels. Scenario B

<table>
<thead>
<tr>
<th>Iterat.</th>
<th>Fully</th>
<th>Type I</th>
<th>Mod. Type I</th>
<th>CBSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>∞</td>
<td>0.0149</td>
<td>0.1013</td>
<td>0.0187</td>
<td>0.0257</td>
</tr>
</tbody>
</table>

Table 5.3: Steady state residual error levels. Scenario C

<table>
<thead>
<tr>
<th>Iterat.</th>
<th>Fully</th>
<th>Type I</th>
<th>Mod. Type I</th>
<th>CBSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>∞</td>
<td>0.0149</td>
<td>0.1013</td>
<td>0.1095</td>
<td>0.0256</td>
</tr>
</tbody>
</table>

A third scenario C, which is a slight modification of scenario B where the known DOA of an interference is now estimated to be \((\theta_{e,3} = 75^\circ, \phi_{e,3} = 225^\circ)\); meanwhile the true interference is arriving from \((\theta_3 = 80^\circ, \phi_3 = 220^\circ)\). The lower plot of Fig. 5.4 shows the convergence behavior of the residual error vs. the number of iterations. Modified Type I (dashed trace) is not able to maintain the low residual error as in scenario B and shows a behavior very close to that of Type I (narrow trace). However, the proposed CBSES array (bold trace) keeps almost the same low residual error as in scenarios A and B. Table 5.3 shows the steady state residual error values.
5.5 Conclusion

We presented in this paper a CBSES partial adaptive CRA for the processing of a narrow-band component of a broad-band signal, that takes advantage of the prior knowledge of DOAs of some interferences. The CBSES array uses both element space and beamspace processing to eliminate the interferences efficiently. The beamspace targets the interferences with known DOAs and the element space is used to cancel the interferences with unknown DOAs. The result is a beamformer that has consistent behavior in maintaining low residual interference and noise levels and at the same time is robust with respect to uncertainties in the interferences with known DOAs.

5.6 Acknowledgments

This research is supported by US AFOSR under contract FA9550-04-C-0058 and by the Polytechnic University of Puerto Rico.
Chapter 6

Beamforming Using the Spherical Array

This chapter is dedicated to beamforming using the SA. After a brief introduction, we describe the previous works and the techniques used on SA beamforming. The chapter continues explaining in detail the three sensor arrangements most used. Then, we suggest our proposed sensor arrangements. A study on the narrowband beamformer analysis follows with performance comparison between the fully and the partial adaptive beamformers. Computational complexity and convergence speed is also analyzed and examples are given. Next, broadband beamforming is presented and we propose a sub-array nesting technique to make the directivity and the number of array elements consistent among different sub-arrays. The chapter ends with a brief study on phase mode SA beamformers and a comparison to the element space beamformers.

6.1 Introduction

The SA is the preferred option for 3-D beamforming because it eliminates the DOA uncertainty present in the ring array meanwhile keeping an invariant beampattern both in azimuth and elevation and 360° coverage.

The element space processing for spherical arrays is able to maintain the noise uniform across all elements compared to the phase mode SA, which suffers of different noise
strength between the phase modes [74]. Also, the element space beamformer array gain against white noise, is constant with frequency (achieving frequency-independent robustness), meanwhile that of the phase mode SA reduces toward the lower frequencies [21].

Partial adaptation dramatically reduces the computational complexity of processing a large number of elements present in the fully adaptive beamformer. Moreover, it presents faster convergence rate and could yield better tracking for non-stationary environments. The optimum steady state $SINR_0$ in the partial adaptive beamformer is reduced but still closer to that of the fully adaptive beamformer.

The present work introduces two new sensor arrangements that optimize the inter-element distance and hence use the least amount of elements needed to satisfy the Nyquist spatial criterion and at the same time form parallel rings of sensors. Then, we propose a narrowband element space partial adaptive beamforming method with the sphere rings. We also introduce a broadband beamformer that uses a nesting technique to expand the beamformer frequency range. Finally, we compare the phase mode with the element space beamformer. Simulations assess the validity of the proposed methods as well as making comparison with other types of beamformers.

6.2 Previous Works on Spherical Array

6.2.1 Early works

Some of the earliest publications on SA include the works of DuHamel in 1952 [75]. He describes an approach using a spherical harmonic expansion to design beampatterns for SA apertures.


Results of theoretical analysis of the radiation patterns produced by a spherical an-
tenna array are reported by Sengupta et al., in 1965 [70]. They place the elements as uniformly as possible so the array presents a constant beampattern. Since the maximum number of elements that can be placed uniformly on a spherical surface is 20 (icosahedron), they use this same polyhedron to create a SA with 182 elements. They do that by placing the elements at equidistant positions on the surfaces of the icosahedron. A sphere is circumscribed about the icosahedron, and all the elements are then brought to the surface of the sphere. This arrangement loses part of the original symmetry but is still useful. The elements happen to form horizontal rings along the sphere for this particular arrangement of 182 elements. A later publication by Sengupta et al. arranges 162 elements using the same procedure.

MacPhie in 1968 [77] expands the phase factor as a truncated series of spherical harmonics or phase modes as DuHamel did for continuous spherical arrays. He also suggests a uniformly dense volume distribution within the sphere that would reduce the mutual coupling of the array elements. In his work he does not suggest any placement of the array elements.

Experimental Investigation on Spherical Arrays is performed by A. K. Chan and R. A. Sigelmann in 1969 [78] where they propose an arrangement of the elements located on parallel ring arrays along the z axis of the sphere.

6.2.2 Recent works

Since the late nineties, there was a been a renewed trend about SA processing. Most of the authors use spherical harmonics or phase modes. Other authors also consider the space domain processing where beamforming is done as a weighted sum of microphone signals.

Tawfik in 1997 [79] and 1999 [80] propose a generic beamforming structure allowing implementation of adaptive processing schemes for planar, circular and cylindrical arrays.
that can be extended to spherical arrays.

In 2002 J. Meyer and G. W. Elko [81, 82] published two articles on spherical microphone array for spatial sound recordings. The beamformer is based on the spherical harmonic decomposition of the soundfield, as MacPhie did. They arrange 32 array elements located at the center of the faces of a truncated icosahedron.

DeWitte et al. in 2003 [83], extend the concept of phase mode excitation of planar circular arrays in to spherical arrays. They arrange the elements in a equiangular grid.

In 2005 Rafaely [20] presents a theoretical and simulation analysis of the measurement errors of spherical microphone arrays. The analysis uses spherical harmonics decomposition on three different element arrangements; the equiangle, the Gaussian sampling and the near uniform sampling. He also compares rigid versus open sphere configurations. In another publication (2007) [21] he compares the phase-mode versus the delay-and-sum processing on spherical arrays. Rafaely address the excessive noise problem suffered for frequencies corresponding to the nodal values of the spatial spherical modes in SA beamforming. He solves the problem using a dual-sphere array. Next year, Rafaely [74] improves the dual-sphere configuration arranging the microphones in the volume of a spherical shell reducing in half the number of sensors compared to the dual-sphere configuration. Within the same year, Rafaely [22] presents an overview of beamforming methods for spherical arrays that includes the delay-and-sum, Dolph-Chebishev, and other methods based in the spherical harmonics domain. In 2008, Agmon and Rafaely [84] published an article about spherical aperture microphone, or continuous array. They use the phase mode method. An article from Parthy in 2008 et al. [85] perform a study on a co-centered two spherical arrays. They also use phase mode. The same year Lin and Qingyu [86] use phase mode to localize multiple acoustic sources using SA.

With respect concentric spherical arrays, we already named the Rafaely work on dual-spheres. Other authors Chan & Chen [18, 87, 88] use a concentric spherical array
to achieve an improved frequency invariant property with respect that of a single SA and avoiding the nulls of the Bessel function that appear in their transformation. It is important to note that Chan & Chen method, even when is a modal transform, is not the same transform as the spherical harmonic mode transform, they are quite different. Chan & Chen method only works for the equiangle sensor arrangement method. They claim that this arrangement method simplify the steering vector expression such is amenable to apply a similar modal transformation used for ring arrays (the Davies transformation). Abhayapala, in 2008 combines two sets of orthogonal functions, the spherical harmonics and the spherical Bessel functions. They claim to solve the numerical problems due to the zeros of the Bessel functions that appear in some of the phase modes.

6.3 Spherical Array Sensor Arrangements

This section study some of the most used sensor arrangements in SA beamforming. It will serve as a starting point to the methods we will propose in the next section.

6.3.1 Equiangle Sensor Arrangement

The equiangle sensor arrangement places the sensors at equal angles in azimuth and elevation. This sensor arrangement can be generated by rotating around the y axis a ring array lying on the $xy$ plane, or rotation around the z axis a ring array lying on the $yz$ plane (or $xz$ plane). Fig. 6.1(a) shows a 288 elements equiangle SA originated from 24 elements of a ring array rotating around the y axis 12 times. Fig. 6.1(b) shows the beampattern using delay-and-sum weights for a narrowband signal of $\kappa = 12.0343$ and inter-element distance of $\lambda/2$. The beampattern presents a $DI$ of 19dB. Fig. 6.1(c) shows the inter element distance between every element and its closest four neighbors. Two of the neighbors are located in the same ring and the two remainder are located in adjacent rings. We notice that the inter-element distance is constant among adjacent
elements in the same ring (\(\lambda/2\)), but is not constant among different rings. That distance is reduced as the elements get closer to the y axis.
6.3.2 Sengupta Sensor Arrangement

Sengupta et al. [70] arrange 182 sensors in 13 parallel rings to the \( xy \) plane with polar and azimuth angles as,

\[
\theta_n = \frac{\pi}{2} - \frac{\pi n}{12}, \quad n = -6, \ldots, 6.
\]

\[
\phi_{n,m} = \frac{2\pi}{5(6-|n|)} m, \quad m = 0, \ldots, 5(6 - |n|) - 1,
\]

(6.1)

where \( n \) represents the ring number and \( m \) is the element number in ring \( n \). The number of elements in each ring is not constant as we can see in Fig. 6.2(a). Fig. 6.2(b) shows

Figure 6.2: 182 elements, Sengupta spherical array.
the beampattern that presents lower sidelobes compared to those of the equiangle SA and a $DI$ of 20.2dB. Also the beampattern is more symmetric around the mainlobe. This is due to the fact that the variation about the inter element distances shown in Fig. 6.2(c) is less spread. In particular, the distance between the elements of the same ring is constant but it is not the same for all rings. The distance between adjacent elements in different rings is near constant in contrast to that of the equiangle SA. We observe that some distances are slightly larger than the Nyquist spatial criterion limit of $\lambda/2$. In any case the beampattern still does not present grating lobes or aliases.

6.3.3 Gaussian Sampling Sensor Arrangement

The Gaussian sampling arranges the sensors in parallel rings with respect to the $xy$ plane. The inter-element distance in each ring is constant and the polar angle that separates each ring is found as the solution of the Legendre polynomials of order $N_e/2$ where $N_e$ is the number of elements in each ring. Fig. 6.3(a) shows the element arrangement for a SA of 288 elements arranged in 12 rings where the each ring is composed of 24 elements. Fig. 6.3(b) shows the beampattern that is less symmetric that of Sengupta and a $DI$ of 19.5dB. Fig. 6.3(c) shows the inter element distance. Is interesting to notice that the distance between adjacent elements in different ring is kept constant, however the distance between adjacent elements in the same ring is not constant from ring to ring since all rings have the same number of elements.

6.4 Proposed Spherical Array Sensor Arrangements

6.4.1 First Proposed Sampling Sensor Arrangement

The sampling arrangement that we propose keeps as uniform as possible the inter-element distance of the SA meanwhile keeping the elements arranged in ring arrays. This last constraint will allows us to create partial array processing based on ring arrays. The
proposed method arranges the elements in parallel rings as those of the Sengupta and Gaussian arrangements, but having a constant inter-element distance of $\lambda/2$ between the adjacent elements in the same ring for all rings. Therefore, the different rings will have different number of elements for different radius. The distance between adjacent rings will also be kept constant to $\lambda/2$. The objective is to maintain the inter-distance elements to be close to $\lambda/2$ in all directions meanwhile maintaining the ring sub-arrays structure.
Fig. 6.4(a) shows the element arrangement for a SA with 186 elements distributed in 13 rings where the larger ring has 24 elements. Fig. 6.4(b) shows the beampattern that shows a good symmetry, even better than that of Sengupta and a $DI$ of 20.4dB, the largest of all discussed methods. Fig. 6.4(c) shows the inter element distance. Is interesting to notice that the distance between adjacent elements in same rings is kept constant and very close to $\lambda/2$. The distance between adjacent elements from other
rings is also close to this number and always smaller than those of Sengupta. The larger variations happen because the distance between elements in different rings is not the distance between the rings. Each ring has a different number of elements, so they are not aligned in elevation. This effect is more visible for the rings closer to the poles of the array.

6.4.2 Second Proposed Sampling Sensor Arrangement

![Diagram of a spherical array](image)

(a) Second proposed spherical array.

![Beampattern image](image)

(b) Beampattern $\theta = 60^\circ, \phi = 0^\circ$.

![Inter-element distance](image)

(c) Inter-element distance.

Figure 6.5: 200 elements, second proposed spherical array.
Alternatively to the first proposed arrangement we also used the Legendre polynomials to calculate the ring separation meanwhile keeping the inter-element distance close to $\lambda/2$ on each ring. The result is a SA with 200 elements arranged in 14 rings as shown in Fig. 6.5. The performance of this arrangement have a better $DI$ than that of the previous proposed arrangement. Also, the inter-element distance is more consistent and closer to $\lambda/2$. The drawback is that uses 14 elements more than the previous method.

### 6.4.3 Comparison Between the Different Sensor Arrangements

Table 6.1 summarizes the different sensor arrangement performance measurements. The first data row represents the array number of elements $K$. The objective is to have the fewer possible elements for the maximum aperture (array physical dimension) meanwhile avoiding spatial aliasing. The second data row is the $A_w$. The larger the value, the better the array rejection against spatially uncorrelated noise. Also, the beamformer will be more robust against steering vector mismatch as it will be explained in Chapter 7. The last data row of the table shows the $DI$ that represents the amount of array gain against isotropic spatially correlated noise. This is a quantitative measurement of the beampattern shape. The narrower the mainlobe and the lower the sidelobes, the higher the value of the $DI$ will be.

<table>
<thead>
<tr>
<th>$K$ (Array elements)</th>
<th>Equiangle</th>
<th>Sengupta</th>
<th>Gaussian</th>
<th>Proposed 1</th>
<th>Proposed 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>288</td>
<td>182</td>
<td>288</td>
<td>186</td>
<td>200</td>
</tr>
<tr>
<td>$A_w$(dB) (Array gain vs. white noise)</td>
<td>24.5939</td>
<td>22.6007</td>
<td>24.5939</td>
<td>22.6951</td>
<td>23.0103</td>
</tr>
<tr>
<td>$DI$ (dB) (Directivity index)</td>
<td>18.9971</td>
<td>20.1924</td>
<td>19.4562</td>
<td>20.3999</td>
<td>20.4341</td>
</tr>
</tbody>
</table>

From the results in the table we observe that the proposed arrangements use less array
elements than those of the Equiangle and Gaussian. These arrangements suffer of a high density of elements in the smaller rings closer to the rotation axis. The directivity is also inferior to the proposed arrangements. The array of Sengupta achieves very close $A_w$ and $DI$. However, from the beampattern Fig. 6.2 the radial symmetry around the main lobe is worse than that of the proposed arrangements. The proposed beamformers have a good balance between maintaining high $A_w$, $DI$, and beampattern radial symmetry properties.

6.5 Study of Narrowband Beamforming Using the Spherical Array

This section analyze the element space narrowband beamformer using SA. We will study and compare the fully adaptive versus the partially adaptive beamformer. The study will be performed by investigating performance parameters like the $SNR_0$ and the $SINR_0$ both in the steady state and along adaptation. Then, we will study the computational complexity. Finally, we will study and compare the convergence rate. The objective is to determine in what situations the partial adaptive beamformer is the preferred option against the fully adaptive beamformer.

6.5.1 Beamformer Output Signal to Noise Ratio

We will obtain the $SNR_0$ mathematical expressions for the fully and the partial adaptive beamformers for a field composed of the SOI and spatially uncorrelated noise only.

A) Fully Adaptive Beamformer $SNR_0$

The output of a fully adaptive beamformer of any geometry is

$$z(\kappa, t) = v^H u_s(\kappa, t) + v^H n(t),$$  \hspace{1cm} (6.2)
where $u_s(\kappa, t)$ is the SOI input vector and $n(t)$ the spatially uncorrelated noise. $\kappa$ is the wavenumber magnitude and $t$ represents the snapshot number. $v$ is the beamformer weight vector.

The expected output power is

$$P_s + P_n = v^H R_s v + \sigma_n^2 v^H I v,$$

where $R_s$ is the spatial correlation matrix of the SOI and $\sigma_n^2 I$ that of the noise.

The SA composed of $K$ elements, performs beamforming assuming delay-and-sum weights as in (2.55). The beamformer output power due to uncorrelated noise only is

$$P_{n-FA-DAS} = \frac{s_0^H}{\|s_0\|^2} (\sigma_n^2 I) \frac{s_0}{\|s_0\|^2} = \frac{\sigma_n^2}{K},$$

where $v = \frac{s_0}{\|s_0\|^2}$ is the delay and sum weight vector, $\sigma_n^2$ is the noise power at any array element, and $s_0^H s_0 = \|s_0\|^2$. For far field we used the equivalence $\|s_0\|^2 = K$.

The $SNR_0$ expression defined in (2.43) becomes,

$$SNR_{0-FA-DAS} = \frac{P_s}{P_{n-FA-DAS}} = \frac{K}{\sigma_n^2},$$

since the delay-and-sum beamformer satisfy the distortionless constraint and $P_s = 1$.

B) Partially Adaptive Beamformer $SNR_0$

The same analysis is done for partial adaptive SA. To find the beamformer weights, we use a partition matrix expression to arrange the weight vectors from each of the $M$ rings. The partition matrix is a sparse matrix whose non zero elements of each particular column produce a sub-array weight vector. The number of sub-arrays is the number of partition matrix columns. The locations of the non-zero elements in a particular column select the array elements that belong to that particular sub-array.

$$P = \begin{bmatrix}
  h_1 & \cdots & \cdots & \cdots & 0_{(N_1 \times 1)} \\
  \vdots & \ddots & \ddots & \ddots & \vdots \\
  0_{(K_m \times 1)} & \cdots & \cdots & h_M
\end{bmatrix}.$$  (6.6)
The beamformer output power due to uncorrelated noise is now

\[
P_{n-PA-DAS} = \frac{s_0^H \mathbf{P}}{\| \mathbf{P}^H s_0 \|^2} \left( \mathbf{P}^H \sigma_n^2 \mathbf{P} \right) \frac{\mathbf{P}^H s_0}{\| \mathbf{P}^H s_0 \|^2},
\]

(6.7)

where \( \frac{\mathbf{P}^H s_0}{\| \mathbf{P}^H s_0 \|^2} \) is the delay and sum weight vector of the partitioned space and \( \left( \mathbf{P}^H \sigma_n^2 \mathbf{P} \right) \) the partitioned noise correlation matrix.

After some manipulation we obtain

\[
P_{n-PA-DAS} = \sigma_n^2 \sum_{m=1}^{M} \frac{1}{K_m} M^2,
\]

(6.8)

where \( K_m \) is number of elements of ring \( m \). We used the equivalences \( \mathbf{P}^H s_0 = 1 \), \( \| \mathbf{P}^H s_0 \|^2 = M \) and the property of the partition matrix columns \( h_m \) satisfy \( h_m^H h_m = \frac{1}{K_m} \).

The \( SNR_0 \) expression defined becomes,

\[
SNR_{0-PA-DAS} = \frac{M^2}{\sigma_n^2 \sum_{m=1}^{M} N^{-m}},
\]

(6.9)

C) \( SNR_0 \) Comparison Between the Fully and the PA Beamformers

The expression (6.9) is never larger than (6.5) and both expressions are equal when \( K_m \) is the same for all rings \( K_m = K/M \). Therefore, we expect the partial array to have a larger steady state output power due to spatially uncorrelated noise as the number of sensor elements in each ring becomes more different.

When using the MVDR criterion to obtain the weight vectors, the same result will be obtained as it is shown in Appendix B.

6.5.2 Beamformer Output Signal to Interference Plus Noise Ratio

The same mathematical general result is not possible when there are interferences in the field. The most we can do is to obtain the mathematical expressions as function of the spatial correlation matrix of the interference plus noise and the array weight vector

113
at steady state. This happens because the correlation matrix is not diagonal and it is scenario dependent. Therefore it will be different depending on the number, DOAs, and powers of the incoming interferences and noise. The analysis is next.

**A) Fully Adaptive Beamformer $SINR_0$**

In our analysis we will consider that the field is composed of the SOI, $L$ interferences whose characteristics we do not know, plus the spatially uncorrelated noise. The expected residual interference plus noise output power is

$$P_{\text{in-FA}} = v^H R_{\text{in}} v,$$  \hspace{1cm} (6.10)

where $R_{\text{in}}$ is the spatial correlation matrix of the input signal of the interferences plus noise,

$$R_{\text{in}} = \sum_{l=1}^{L} R_{il} + \sigma_n^2 I.$$  \hspace{1cm} (6.11)

The expression $R_{il} = \sigma_i^2 i_i^H i_l$ was defined in section 2.5.3.

One of the methods to reduce the interference to the level of the uncorrelated background noise is the MVDR criterion. The beamformer weight vector solution (2.59) is repeated here,

$$v_{\text{MVDR}} = R_{\text{in}}^{-1} s_0 s_0^H R_{\text{in}}^{-1},$$  \hspace{1cm} (6.12)

where $s_0$ is the array steering vector when tuned to the SOI. Replacing (6.12) and (6.11) in (6.10) and after some manipulations we obtain the expression for the steady state residual interference plus noise output power of the fully array as,

$$P_{\text{in-FA-MVDR}} = \left(s_0^H R_{\text{in}}^{-1} s_0\right)^{-1}.$$  \hspace{1cm} (6.13)

The $SINR_0$ general expression is defined in (2.46) as

$$SINR_0 = \frac{P_s}{P_{\text{in}}}.$$  \hspace{1cm} (6.14)
The expression (6.14) particularized for the fully adaptive array that uses the MVDR criterion weights becomes

\[
SINR_{0-FA-MVDR} = s_0^H R_{in}^{-1} s_0,
\]  

(6.15)
since the MVDR criterion satisfy the distortionless beamformer constraint and \( P_s = 1 \).

B) Partially Adaptive Beamformer \( SINR_0 \)

A partial adaptive beamformer can be modeled with a partition matrix. When applying the partition matrix to the array input signal we obtain a reduced space vector as,

\[
y(\kappa, t) = P^H u(\kappa, t).
\]  

(6.16)
The reduced space vector signal is combined with a weight vector \( w \) to obtain the beamformer output

\[
z(\kappa, t) = w^H y(\kappa, t).
\]  

(6.17)
Therefore, the residual interference plus noise output power is,

\[
P_{in-PA} = w^H (P^H R_{in} P) w,
\]  

(6.18)
where \( P^H R_{in} P \) is the spatial correlation matrix of \( y(\kappa, t) \) analog to (6.10). If we use the MVDR criterion to obtain \( w \) we have,

\[
w_{MVDR} = \frac{(P^H R_{in} P)^{-1} (P^H s_0)}{s_0^H P (P^H R_{in} P)^{-1} P^H s_0},
\]  

(6.19)
which is analog to (6.12) where \( P^H s_0 \) is the SOI steering vector in the reduced space.

Replacing (6.19) in (6.18) we obtain the steady state residual interference plus noise power for a partial array that uses the MVDR criterion weights,

\[
P_{in-PA-MVDR} = \left( s_0^H P (P^H R_{in} P)^{-1} P^H s_0 \right)^{-1}.
\]  

(6.20)
The \( SINR_0 \) is

\[
SINR_{0-PA-MVDR} = s_0^H P (P^H R_{in} P)^{-1} P^H s_0,
\]  

(6.21)
C) $SINR_0$ Comparison Between the Fully and the PA Beamformers

To compare the performance of the fully array with respect the partial in terms of steady state residual interference and noise power we need to evaluate (6.15) and (6.21). Both have similar expression except that in the expression for the partial array, the spatial correlation matrix is projected onto the space spanned by the partition matrix and then expanded to the original space. In general the projection expansion will make (6.21) to be smaller or at least equal to (6.15) (if $P = I$, the identity matrix). Therefore the performance of the partial array with respect the fully array will decrease in general. For a particular scenario of $\mathbf{R}_{in}$ the choice of partition matrix $P$, the geometry, and the SOI DOA (the latter two determine $s_0$) will give us quantitative measurements of the residual interference plus noise output power. We can compare the loss of performance from a fully array to a partial array, or we can compare performances between partial arrays that use different sub-array configurations and/or geometries [89].

In our examples, the expectation of $P_m$ is performed with ensemble averages. For a more general comparison we would need to perform Monte Carlo simulations with different $\mathbf{R}_{in}$. That is, the interference DOAs, powers, and noise power.

6.6 Computational Complexity Between the Fully and the Partial Adaptive Beamformers

This section obtains the computational complexity of the fully adaptive beamformer when implementing the NLMS and RLS algorithms. We will repeat the computations for the partial adaptive beamformer. Mathematical results are given and some examples as well. The computational reduction for the SA that uses the first proposed sampling arrangement is dramatic as well as achieving faster convergence.
6.6.1 Fully Adaptive Normalized Least Mean Squares

A fully adaptive beamformer composed of \( K \) elements using the NLMS algorithm in a GSC configuration is composed of the quiescent weight vector, the blocking matrix and the low branch adaptive weight vector that adapts with the NLMS algorithm as it is shown in Fig. 6.6. Each part would need the following complex multiplications per snapshot:

- **Quiescent weight vector** \( \mathbf{w}_{q-FA} \): \( K \)
- **Blocking matrix** \( \mathbf{B}_{FA} \): \( K(K-Q) \)
- **Low branch adaptive weight vector** \( \mathbf{w}_{a-FA} \): \( K-Q \)
- **NLMS algorithm**: \( 2(K-Q) \)

The total number of complex operations is, after some manipulations

\[
\text{Ops}_{FA-NLMS} = K + (K+3)(K-Q), \quad (6.22)
\]

where \( K \) is the number of elements in the SA and \( Q \) the number of constraints. For the SA that uses the first proposed sampling arrangement where \( K = 186 \) and \( Q = 1 \) the number of complex multiplications is 35,151.
6.6.2 Fully Adaptive Recursive Least Squares

A fully adaptive array composed of $K$ elements using the RLS algorithm in a GSC configuration is composed of the same blocks as before. The low branch adaptive weight vector adapts now with the RLS algorithm. Each part would need the following complex multiplications per snapshot:

- Quiescent weight vector $w_{q-FA}$: $K$
- Blocking matrix $B_{FA}$: $K(K - Q)$
- Low branch adaptive weight vector $w_{a-FA}$: $K - Q$
- RLS algorithm: $3(K - Q)^2 + 4(K - Q)$

The total number of complex operations is

$$O_{PS_{FA-RLS}} = K + (K + 5)(K - Q) + 3(K - Q)^2.$$  \hspace{1cm} (6.23)

For the SA that uses the first proposed sampling where $K = 186$, $M = 13$ and $Q = 1$ the number of complex multiplications is 138,196.

6.6.3 Partial Adaptive Normalized Least Mean Squares

A partial adaptive array composed of $K$ distributed in $M$ rings using the NLMS algorithm is composed of the partition matrix followed by the same GSC configuration as before, which now operates on the reduced $M$ dimensional space. The block diagram is shown in Fig. 6.7 Each part would need the following complex multiplications per snapshot:

- Partition matrix $P$: $K$
- Quiescent weight vector $w_{q-PA}$: $M$
Figure 6.7: Partial Adaptive Beamformer, block diagram.

- Blocking matrix $B_{PA}$: $M(M - Q)$
- Low branch adaptive weight vector $w_{a-PA}$: $M - Q$
- NLMS algorithm: $2(M - Q)$

The total number of complex operations is

$$\text{Ops}_{PA-NLMS} = K + M + (M + 3)(M - Q), \quad (6.24)$$

For the SA that uses the first proposed sampling where $K = 186$, $M = 13$ and $Q = 1$ the number of complex multiplications is 391.

6.6.4 Partial Adaptive Recursive Least Squares

A partial adaptive array composed of $K$ elements distributed in $M$ rings using the RLS algorithm is composed of the partition matrix followed by the same GSC configuration as before operating on the reduced $M$ dimensional space. The low branch adaptive weight vector adapts with the RLS algorithm. Each part would need the following complex multiplications per snapshot:

- Partition matrix $P$: $K$
- Quiescent weight vector $w_{q-PA}$: $M$
• Blocking matrix $B_{PA}$: $M(M - Q)$

• Low branch adaptive weight vector $w_{a,PA}$: $M - Q$

• RLS algorithm: $3(M - Q)^2 + 4(M - Q)$

The total number of complex operations is

$$\text{Ops}_{PA-RLS} = K + M + (M + 5)(M - Q) + 3(M - Q)^2.$$  \hspace{1cm} (6.25)

For the SA that uses the first proposed sampling where $K = 186$, $M = 13$ and $Q = 1$ the number of complex multiplications is 847.

6.7 Convergence Speed

Convergence speed is crucial in beamforming. In this section we will study the convergence speed of the beamformer that uses the GSC configuration. We will compare the fully and the partial adaptive beamformer when using the NLMS and the RLS adaptation algorithms.

6.7.1 Convergence Speed of the NLMS Algorithm

The Normalized Least Mean Squares is a gradient algorithm famous for its simplicity and low computational cost. The convergence rate of this algorithm is dictated by the number of adaptive weights and by the eigenvalue spread of the correlation matrix used to obtain the weights.

The fully adaptive beamformer using the GSC configuration has $K - Q$ adaptive coefficients as shown in Fig. 6.6, being $Q$ the number of constraints. On the other hand, the partial adaptive beamformer has $M - Q$ adaptive coefficients. Since $M << K$ we expect the partially adaptive beamformer to achieve faster convergence speed than the fully adaptive beamformer.
However, not only the speed of the NLMS algorithm depends on the number of adaptive weights, but also on the eigenvalue spread of the correlation matrix. For the fully adaptive beamformer, the correlation matrix is

$$R_{y_{a-FA}} = B_{FA}^H E[y(t)y_H^H(t)]B_{FA}.$$  \hspace{1cm} (6.26)

For the partial adaptive beamformer is

$$R_{y_{a-PA}} = B_{PA}^H P^H E[y(t)y_H^H(t)]PB_{PA}.$$  \hspace{1cm} (6.27)

From (6.26) and (6.27) it is not possible to obtain a general expression that compares the eigenvalue spread between both expressions. The only common term is $E[y(t)y_H^H(t)]$, meanwhile the blocking matrices are different.

### 6.7.2 Convergence Speed of the RLS Algorithm

Contrary to the NLMS, the convergence speed of the RLS algorithm does not depend on the eigenvalue spread of the correlation matrix. It only depends on the number of adaptive coefficients. We expect the partial adaptive RLS beamformer to perform faster than that of the fully adaptive RLS algorithm.

### 6.8 Narrowband Beamforming Simulations

This section illustrate with examples the analysis of the previous section. We show steady state Monte Carlo trials to show the general beamformer behavior under different conditions. We also show adaptation examples using Monte Carlo trials. Finally we show two examples of DOA change in one of the interferences.

The experiments implemented in this section use the first proposed SA with 186 microphone sensors arranged in 13 rings. The equatorial ring is composed of 24 elements. The 6 rings of the upper hemisphere are composed of 1, 6, 12, 17, 21, and 24 elements.
The 6 rings of the lower hemisphere have the same number of elements. The inter-element distance between adjacent elements is set to $\lambda/2$ where $\lambda$ is the wavelength corresponding to a 1kHz frequency.

### 6.8.1 Steady-State $SINR_0$ Monte Carlo Simulations

This sub-section shows the simulation results of the steady state $SINR_0$ from 5000 Monte Carlo trials. We compare the steady state $SINR_0$ for fully adaptive and partial adaptive beamformers using delay-and-sum and the optimum weights (MVDR criterion). The results corroborate the results found in the analysis section.

The steady state $SINR_0$ results from 5000 Monte Carlo trials when using one, three, and six interferences are shown in Tables 6.2 to 6.4. The interferences have uniform random DOAs outside the beampattern mainlobe. The SOI DOA is ($\phi = 0^\circ, \theta = 90^\circ$) with a SNR of 0dB. The interferences and noise input powers are kept constant to the values indicated in the tables.

From the results we observe that when beamforming with delay-and-sum weights the $SINR_0$ is very low because the beamforming do not target the interferences. When using the MVDR criterion to target the interferences, the $SINR_0$ is close to the maximum value of 22.6951 dB that correspond to $10\log_{10}(K)$ as indicated in (2.56) where $K = 186$ is the number of array elements. As the number of interferences increases, the partial adaptive beamformer decreases its performance, but still is close to the maximum $SINR_0$ of the array.

### 6.8.2 Monte Carlo Simulations of $SINR_0$ Along Adaptation

Fig. 6.8 shows the results of 100 Monte Carlo trials for a fully and partial adaptive beamformers in a GSC configuration that use the NLMS and RLS adaptation methods. The sampling scheme is the first one proposed by Vicente on a SA with 186 elements.
Table 6.2: $SINR_0$ Monte Carlo results 5000 trials, 1 interference. SNR=0dB, SIR=-25dB.

<table>
<thead>
<tr>
<th>Method</th>
<th>$SINR_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max $SINR_0$</td>
<td>22.6951</td>
</tr>
<tr>
<td>Fully adaptive, delay-and-sum</td>
<td>-0.6864</td>
</tr>
<tr>
<td>Partial adaptive, delay-and-sum</td>
<td>-5.9992</td>
</tr>
<tr>
<td>Fully adaptive, MVDR</td>
<td>22.6789</td>
</tr>
<tr>
<td>Partial adaptive, MVDR</td>
<td>22.4223</td>
</tr>
</tbody>
</table>

Table 6.3: $SINR_0$ Monte Carlo results 5000 trials, 3 interferences. SNR=0dB, SIRs=-25, -35, -30dB.

<table>
<thead>
<tr>
<th>Method</th>
<th>$SINR_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max $SINR_0$</td>
<td>22.6951</td>
</tr>
<tr>
<td>Fully adaptive, delay-and-sum</td>
<td>-12.1358</td>
</tr>
<tr>
<td>Partial adaptive, delay-and-sum</td>
<td>-17.4718</td>
</tr>
<tr>
<td>Fully adaptive, MVDR</td>
<td>22.6466</td>
</tr>
<tr>
<td>Partial adaptive, MVDR</td>
<td>21.7491</td>
</tr>
</tbody>
</table>

Table 6.4: $SINR_0$ Monte Carlo results 5000 trials, 6 interferences. SNR=0dB, SIRs=-25, -35, -30, -35, -30, -25dB.

<table>
<thead>
<tr>
<th>Method</th>
<th>$SINR_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max $SINR_0$</td>
<td>22.6951</td>
</tr>
<tr>
<td>Fully adaptive, delay-and-sum</td>
<td>-15.1165</td>
</tr>
<tr>
<td>Partial adaptive, delay-and-sum</td>
<td>-20.4838</td>
</tr>
<tr>
<td>Fully adaptive, MVDR</td>
<td>22.5984</td>
</tr>
<tr>
<td>Partial adaptive, MVDR</td>
<td>20.1949</td>
</tr>
</tbody>
</table>

arranged in 13 ring arrays. The number of interferences is six with fixed SIR values indicated in Table 6.4. The interferences are randomly located outside the beampattern mainlobe. The SOI DOA is ($\phi = 0^\circ, \theta = 90^\circ$) with a SNR of 0dB. The upper figure shows the adaptation when using the NLMS algorithm, the lower figure shows adaptation using the RLS algorithm. The fully adaptive beamformer $SINR_0$ result is shown in bold trace, and that of the partial adaptive beamformer in fine trace.

In both figures, the $SINR_0$ from partial adaptation have faster initial speed rate.
being outperformed by the fully adaptive beamformer only after getting closer to the steady state. When using NLMS, both fully and partial adaptive beamformers suffer of lower $SINR_0$ values with respect the theoretical maximum at the steady state. This is caused from the eigenvalue spread of the correlation matrices. Another cause of the longer time to achieve convergence is because the variable NLMS step size. The step size is decreased along the iterations in both fully adaptive and partial adaptive beamformers by the use of a coefficient factor $c_{\text{fact}}$ in order to reduce the excess mean square error (EMSE) at steady state [90]. The expression for the NLMS step size is

$$
\mu = \frac{1}{c_{\text{fact}}} \frac{1}{\mathbf{y}_a^H(\kappa, t) \mathbf{y}_a(\kappa, t)},
$$

where $\mathbf{y}_a(\kappa, t)$ is the lower signal vector after the blocking matrix. The decreasing of the step size is performed to reduce the EMSE. The price to pay is a larger adaptation time.

The results from the RLS method are shown in the lower figure. The $SINR_0$ from the partial adaptive beamformer, in fine trace, has a faster speed rate than that of the fully adaptive beamformer, in bold trace, being outperformed around the snapshot 600. Both curves get closer to the optimum $SINR_0$ values at steady state.
Table 6.5: Excess $SINR_0$ error, 100 Monte Carlo trials

<table>
<thead>
<tr>
<th></th>
<th>RLS $SINR_0$ (dB)</th>
<th>Fully adaptive</th>
<th>Partial adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimum</td>
<td>22.5915</td>
<td>20.1614</td>
<td></td>
</tr>
<tr>
<td>From adaptation (average)</td>
<td>21.4000</td>
<td>18.0000</td>
<td></td>
</tr>
<tr>
<td>EMSE</td>
<td>1.1915</td>
<td>2.1614</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NLMS $SINR_0$ (dB)</td>
<td>Fully adaptive</td>
<td>Partial adaptive</td>
</tr>
<tr>
<td>Optimum</td>
<td>22.5915</td>
<td>20.1614</td>
<td></td>
</tr>
<tr>
<td>From adaptation (average)</td>
<td>13.0000</td>
<td>11.0000</td>
<td></td>
</tr>
<tr>
<td>EMSE</td>
<td>9.5915</td>
<td>9.1614</td>
<td></td>
</tr>
</tbody>
</table>

To quantify the $SINR_0$ error at steady state, we obtain the difference between the optimum $SINR_0$ value and an average of the $SINR_0$ around the snapshot 6000. This difference is referred to as the EMSE. The results are shown in Table 6.5. The EMSE is small for RLS because the curves achieve the steady-state much earlier than snapshot 6000. This is because the adaptation does not depend on the eigenvalue spread of the correlation matrix. For NLMS, the EMSE is large because the curves are still adapting at snapshot 6000. This happens because both the correlation matrix eigenvalue spread and the variable step size as explained earlier. Appendix D shows the eigenvalue analysis of a particular example justifying the beamformer behavior.

### 6.8.3 Simulations With Change of interference DOAs

This subsection shows the $SINR_0$ results when there is a DOA change in one of the interferences in the middle of the adaptation. We use the same narrowband adaptive beamformer and the same sampling scheme as those used in last subsection.

In this particular case, the field scenario is composed of one SOI, six interferences, and isotropic spatial-time uncorrelated gaussian noise. The SOI and interferences are narrowband with a frequency of 1kHz with Gaussian random amplitudes at each snapshot. The SOI has SNR = 0dB, and a DOA$_{(θ,\phi)}$=$(90^°,0^°)$, where $θ$ and $ϕ$ are the polar and azimuth angles respectively. The six interferences have the DOAs and SIRs shown.
in Table 6.6. The experiments start with scenario SA6, and the DOA of one interference changes at a particular snapshot number.

### Table 6.6: Scenarios DOAs and SIRs

<table>
<thead>
<tr>
<th>Int #:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario</td>
<td>Interference DOA (θ, φ)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA6</td>
<td>75° 120° 129.4° 162° 65° 230° 125° 260° 50° 220° 90° 170°</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA6b</td>
<td>65° 132° 129.4° 162° 65° 230° 125° 260° 50° 220° 90° 170°</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA6 2</td>
<td>75° 120° 119.4° 172° 65° 230° 125° 260° 50° 220° 90° 170°</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario</td>
<td>Interference SIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>-25</td>
<td>-35</td>
<td>-30</td>
<td>-35</td>
<td>-30</td>
<td>-25</td>
</tr>
</tbody>
</table>

As before, we use an element space fully adaptive beamformer, which combines all 186 inputs with adaptive weights and adding the result to obtain the beamformer output. Also, we also use a partial adaptive beamformer that process each ring with fixed weights (delay-and-sum) and combine the rings outputs with adaptive weights to obtain the beamformer output. The adaptation algorithms are the NLMS and the RLS in a GSC configuration.

![SINR adaptation results](image)

(a) Scenario change SA6 to SA6b.  
(b) Scenario change SA6 to SA6_2.

Figure 6.9: $SINR_0$ adaptation results, 20 ensembles.

Fig. 6.9(a)(b) show the $SINR_0$ adaptation when using NLMS (upper figures) and
RLS (lower figures). Bold lines represent the $SINR_0$ from fully adaptive beamformers and fine lines that of the partial adaptive beamformers. Fig. 6.9(a) Shows a scenario change from SA6 to SA6b at snapshot $t = 3000$ where the DOA of the first interference changes from $(75^\circ, 120^\circ)$ to $(65^\circ, 132^\circ)$. Fig. 6.9(b) Shows a scenario change from SA6b to SA6a at snapshot $t = 1000$ where the DOA of the second interference changes from $(129.4^\circ, 162^\circ)$ to $(119.4^\circ, 172^\circ)$.

The initial behavior is consistent to that found in previous simulations (Fig. 6.8). In all cases the partial adaptive has a faster convergence rate but a slightly smaller steady state $SINR_0$. Also, all beamformers drop the $SINR_0$ due to the change of interference DOA and resume adaptation. We observe that for different scenario change there is a different $SINR_0$ drop. We investigated the cause of this behavior and we found that it depends on the location of the interference DOA with respect the beampattern sidelobes. If the interference suddenly changes from a low sidelobe value to a high sidelobe value, the $SINR_0$ drop is more dramatic since is equivalent to suddenly increasing the power of the interference. Therefore, the beamformer has to start adapting again to achieve the same $SINR_0$ as that before the change. This behavior happens both for partial or fully adaptive beamformers. With respect the $SINR_0$ recovery, in general for partial adaptive beamformers the $SINR_0$ recovers quicker, both in NLMS and RLS. The RLS recovers much faster than the NLMS as we expected.

### 6.8.4 Computational Complexity Example

<table>
<thead>
<tr>
<th>Method</th>
<th>NLMS</th>
<th>RLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully adaptive</td>
<td>35151</td>
<td>138196</td>
</tr>
<tr>
<td>Partial adaptive</td>
<td>391</td>
<td>847</td>
</tr>
</tbody>
</table>

The number of complex multiplications is shown in Table 6.7. We notice that the
NLMS is less computational expensive than the RLS in both fully and partial adaptive beamformers. Also, both partial adaptive beamformers that use NLMS or RLS show huge reduction in computational cost compared to those of the fully adaptive beamformer.

From the experiments we corroborate that the use the partial adaptive SA for narrowband applications is very attractive. The performance in $\text{SINR}_0$ of the RLS partial adaptive beamformer has a faster convergence speed to that of the fully adaptive beamformer. Also, the computational savings are dramatic as seen in Table 6.7. The trade off is to have a larger EMSE at steady state as shown in Table 6.5.

6.9 Broadband Beamforming Using the Spherical Array

6.9.1 The Frequency Domain Broadband Beamformer

Broadband beamforming will be performed using the frequency domain technique where the signal read at each sensor is decomposed in to several frequency bins with a DFT. Narrowband beamforming is performed at each frequency bin. The broadband beamformer output is obtained by applying the IDFT to each narrowband beamformer output. Fig. 6.10 shows the frequency domain broadband beamformer composed of $K$ array elements and $N$ frequency bins where $[u]_k$ represents the element $k$ of the array input vector and $\kappa_n = \omega_n/c$ the wavenumber corresponding to the $n^{th}$ frequency bin. Therefore, the broadband beamforming is reduced to $N$ narrowband beamformers whose implementation was shown in previous sections.

6.9.2 Using Nesting on Spherical Array

For broadband beamforming using element space weights, the directivity is reduced (the beampattern mainlobe widens) as the frequency bin decreases. For signals with
bandwidth that spans more than one octave, the directivity will be dramatically reduced at the lowest frequencies. This behavior is inherent to the element space beamformers. To force a constant directivity for the lower frequencies is necessary to increase the array aperture. This solution is not practical since we would need to implement multiple concentric spherical arrays, one for each frequency bin. The increment in elements would be very costly.

One way to alleviate this effect is to perform array nesting. This technique reuses different array sensors for different frequency sub-bands creating sub-arrays. First, we divide the frequency bandwidth into sub-bands. For the lowest frequency sub-band we will design the SA aperture (spatial area) to achieve certain directivity and arrange the array elements to avoid spatial aliasing. For higher frequency sub-bands we will reuse the smaller rings in a sub-array that spans a reasonable aperture to maintain similar directivity than that of the previous sub-band. In order to avoid spatial aliasing the
inter-element distance should be reduced. Therefore, the reused rings should have more elements. Also, it would be necessary to add extra rings between those of the previous sub-band to reduce the inter-element distance between the elements in adjacent rings. There is a trade off between the necessary number of rings (and the number of elements) used for the larger sub-bands and the minimum acceptable directivity. We will use the minimum number of rings that maintains the directivity inside a certain range. That constraint could be difficult to satisfy, since each new added ring becomes the largest of the sub-array. Thus, increasing the array elements dramatically, which could unbalance the consistency with respect the other sub-arrays.

In the next subsection we give a solution for broadband beamforming of acoustic signals that spans from 250Hz to 4kHz.

6.9.3 Design of a Nested SA for Acoustic Signals

A) Sensor Arrangement and Sub-bands

![Image of nested spherical array]

Figure 6.11: nested spherical array.

This subsection shows the implementation of a broadband beamformer using nesting on a SA. The sensor arrangement is the first proposed sampling arrangement. We
consider three sub-arrays arranged as shown in Fig. 6.11. Each sub-array will perform broadband beamforming in the three frequency bands in Table 6.8. The first sub-array is composed of the whole array of 186 elements and 13 rings and its frequency range spans from 250Hz to 1kHz. The second sub-array is composed of 165 elements distributed in 8 rings with elements arranged as shown in Table 6.8. Its frequency range spans from 1kHz to 2kHz. The third sub-array is composed of 175 elements distributed in 8 rings as shown in Table 6.8. The frequency range spans from 2Khz to 4kHz.

B) Steady State Performance Measures

Table 6.9: Steady state performance results. 1kHz sub-array. 186 elements.

<table>
<thead>
<tr>
<th>Freq.</th>
<th>1kHz</th>
<th>0.5kHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fully</td>
<td>PA</td>
</tr>
<tr>
<td>$SINR_0$(dB)</td>
<td>22.6708</td>
<td>22.2327</td>
</tr>
<tr>
<td>$DI$(dB)</td>
<td>20.4105</td>
<td>20.1669</td>
</tr>
</tbody>
</table>

Table 6.10: Steady state performance results. 2kHz sub-array. 165 elements.

<table>
<thead>
<tr>
<th>Freq.</th>
<th>2kHz</th>
<th>1kHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fully</td>
<td>PA</td>
</tr>
<tr>
<td>$SINR_0$(dB)</td>
<td>22.1044</td>
<td>17.6661</td>
</tr>
<tr>
<td>$DI$(dB)</td>
<td>20.7233</td>
<td>18.2167</td>
</tr>
</tbody>
</table>

Tables 6.9 to 6.11 show the steady state performance measures for this sub-array. The scenario used to find the $SINR_0$ and the directivity index is composed of the SOI
Table 6.11: Steady state performance results. 4kHz sub-array. 175 elements.

<table>
<thead>
<tr>
<th>Freq.</th>
<th>4kHz</th>
<th>2kHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param.</td>
<td>Fully</td>
<td>PA</td>
</tr>
<tr>
<td>$SINR_0$ (dB)</td>
<td>22.4155</td>
<td>21.5386</td>
</tr>
<tr>
<td>$DI$ (dB)</td>
<td>21.6131</td>
<td>20.8941</td>
</tr>
</tbody>
</table>

with a SNR of 0dB and DOA of $(\theta = 45^\circ, \phi = 0^\circ)$. There are five interferences with SIRs of -15 -20 -17 -12 -16dB and DOAs randomly distributed outside the mainlobe with 100 Monte Carlo trials.

From the tables, we observe that there is not appreciable $SINR_0$ reduction per octave at the same sub-array either in fully or partial adaptive beamformers. The $SINR_0$ reduction from fully to partial adaptive in the same frequency is not too large either except at the 2kHz sub-array where it is reduced some dBs. This is because we either design the sub-array with 165 elements and 8 rings, or with 207 elements and 9 rings (42 elements in the new rings). If using 9 rings the directivity is better, but at a cost of dramatically increasing the number of elements.

The directivity also experiments a small reduction in all sub-bands except in the 2kHz sub-array where the difference is of around 2dB when using the partial adaptive beamformer.

For comparison purposes, Table 6.12 shows the $SINR_0$ and $DI$ for a SA with the same aperture as the one used in the previous example. Here, nesting is not used. To avoid spatial aliasing at the largest frequency, the array would need to have 739 elements. The results show that the $DI$ spans more than 12dB from the smallest to the largest frequency.

Summarizing, implementing nesting for broadband beamformers that span their bandwidth in more than one octave is a better option to keep a consistent $DI$ among sub-bands, reuse the array elements and rings, and finally keeping small and consistent
Table 6.12: Steady state performance results. 500-4kHz. 739 elements.

<table>
<thead>
<tr>
<th>Freq.</th>
<th>4kHz</th>
<th>2kHz</th>
<th>1kHz</th>
<th>500Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param.</td>
<td>Fully</td>
<td>PA</td>
<td>Fully</td>
<td>PA</td>
</tr>
<tr>
<td>DI(dB)</td>
<td>27.307</td>
<td>26.077</td>
<td>23.919</td>
<td>22.0883</td>
</tr>
</tbody>
</table>

the number of sub-array elements in all sub-bands.

6.10 Phase Mode Beamforming with Spherical Arrays

This section study the spherical harmonics or phase mode beamforming technique. This technique applies the spherical Fourier transform to the array inputs and perform beamforming in the Fourier domain. The resulting beamformer has the property of being frequency invariant. Therefore, the array directivity is constant for the same array aperture and a large range of frequencies. The cost to pay is a reduction in the array gain against the spatially uncorrelated noise. Next subsections show the signal model, the beamformer analysis, and the comparisons with the element space beamformer of the previous sections.

6.10.1 Spherical Harmonics

Lets consider a sound far field with pressure expression $u(\kappa, r, \theta, \phi, \theta_0, \phi_0)$, where $\kappa$ is the wave number magnitude $\kappa = 2\pi/\lambda$, $(r, \theta, \phi)$ is the spatial location on the sphere where we are measuring the field, and $(\theta_0, \phi_0)$ is the DOA of the sound field. Both the DOA and the spatial locations are expressed in spherical coordinates (as explained in section 2.1.1). The $o^{th}$ order and $m^{th}$ degree spherical Fourier transform of $u(\kappa, r, \theta, \phi, \theta_0, \phi_0)$ is defined as [91]

$$u_{o,m}(\kappa, r, \theta_0, \phi_0) = \int_{\theta=0}^{\pi} \sin \theta d\theta \int_{\phi=0}^{2\pi} u(\kappa, r, \theta, \phi, \theta_0, \phi_0) Y_o^m(\theta, \phi) d\phi, \quad 0 \leq o < \infty, \quad (6.29)$$
where \( m = -o, \ldots, o \), \( Y^m_o(\theta, \phi) \) is the spherical harmonic of order \( o \) and degree \( m \) with expression [91]

\[
Y^m_o(\theta, \phi) = \sqrt{\frac{(2o + 1)(o - m)!}{4\pi(o + m)!}} P^m_o(\cos \theta)e^{im\phi},
\]

(6.30)

and \( P^m_o \) is the Legendre function of order \( o \) and degree \( m \).

The inverse Fourier transform expression is found as

\[
u(\kappa, r, \theta, \phi, \theta_0, \phi_0) = \sum_{o=0}^{\infty} \sum_{m=-o}^{o} u_{o,m}(\kappa, r, \theta_0, \phi_0) Y^m_o(\theta, \phi).
\]

(6.31)

The spherical harmonics coefficients of a plane wave \( u(\kappa, r, \theta, \phi, \theta_0, \phi_0) = e^{i\kappa a T_o p} \) of wavenumber \( \kappa \) arriving at a sphere or radius \( r \) from a DOA: \((\theta_0, \phi_0)\) are [92, 93]

\[
u_{o,m}(\kappa, r, \theta_0, \phi_0) = b_o(kr) Y^{m*}_o(\theta_0, \phi_0),
\]

(6.32)

where for open spheres

\[
b_o(kr) = 4\pi i^o j_o(kr)
\]

(6.33)

and \( j_o \) is the spherical Bessel function\(^1\) of order \( o \).

This beamforming method could be thought as a natural way of analyzing the 3-D space sound field, because the sound field is decomposed into an orthogonal set of eigen-functions of the acoustic wave equation in spherical coordinates, \( i.e. \) the spherical harmonics [94, 95].

In practice we sample the sphere by arranging an array of \( K \) microphone sensors on the sphere surface and (6.32) is not exact anymore. Therefore, we need to approximate (6.29) with a summation along the \( K \) array elements as [74]

\[
\hat{u}_{o,m}(\kappa, r, \theta_0, \phi_0) = \sum_{k=1}^{K} c^k_{o,m} u(\kappa, r, \theta_k, \phi_k, \theta_0, \phi_0),
\]

(6.34)

where the coefficient \( c^k_{o,m} \) includes the term \( Y^{m*}_o(\theta, \phi) \) and is chosen to approximate the orthogonality of the transformation as it will be shown in detail on the next section.

\(^1\)This section uses \( j \) for the spherical Bessel function and \( i \) for the imaginary quantity \( \sqrt{-1} \)
Also we approximate (6.31) by limiting the order of the spherical harmonics to $O \leq \sqrt{K} - 1$ [20] as we will explain in section 6.10.3. The zeros of the spherical Bessel functions also impose a constraint on the SA radius, $\kappa r < O$ as we will explain in section 6.10.5.

### 6.10.2 Obtaining the $c_{o,m}^k$ Coefficients

The $c_{o,m}^k$ coefficients are found by forcing the orthogonality property of the spherical harmonics transformation that is

$$\int_{\theta=0}^{\pi} \sin \theta d\theta \int_{\phi=0}^{2\pi} Y_{o}^{m*}(\theta, \phi) Y_{o'}^{m'}(\theta, \phi) d\phi = \delta_{nn'} \delta_{mm'},$$

(6.35)

where $\delta_{nn'}$ is the Kronecker delta function. For a sphere sampled on $K$ points that uses spherical harmonics up to order $O$, the equivalent expression becomes

$$\sum_{k=1}^{K} c_{o,m}^k Y_{o'}^{m'}(\theta_k, \phi_k) = \delta_{nn'} \delta_{mm'}, 0 \leq o \leq O, 0 \leq o' \leq O,$$

(6.36)

where the coefficient $c_{o,m}^k$ includes the term $Y_{o}^{m*}(\theta_k, \phi_k)$ [74]. Therefore, the obtention of the $c_{o,m}^k$ coefficients is found by solving the following equation system [74]

$$\mathbf{Y} \mathbf{C} = \mathbf{I}.$$  

(6.37)

$\mathbf{Y}$ is the $(O + 1)^2 \times K$ matrix formed by the total number of spherical harmonics $Y_{o'}^{m'}(\theta_k, \phi_k)$ arranged as follows

$$\mathbf{Y} = \begin{bmatrix} Y_0^0(\theta_1, \phi_1) & Y_0^0(\theta_2, \phi_2) & \cdots & Y_0^0(\theta_K, \phi_K) \\ Y_1^{-1}(\theta_1, \phi_1) & Y_1^{-1}(\theta_2, \phi_2) & \cdots & Y_1^{-1}(\theta_K, \phi_K) \\ \vdots & \vdots & \ddots & \vdots \\ Y_O^0(\theta_1, \phi_1) & Y_O^0(\theta_2, \phi_2) & \cdots & Y_O^0(\theta_K, \phi_K) \end{bmatrix}.$$  

(6.38)

Each row indicates a particular spherical harmonic of order $o$ and degree $m$ along the $K$ sphere locations where we sample the soundfield. Since for a particular order $o$, $Y_o^m(\theta, \phi)$
has several degrees, from \( m = -o, \ldots, o \), the total number of rows is \((O + 1)^2\) and the number of columns is \(K\).

The matrix \( C \) of dimension \( K \times (O + 1)^2 \) is formed by all \( c_{o,m}^k \) coefficients arranged as

\[
C = \begin{bmatrix}
    c_{0,0}^1 & c_{1,-1}^1 & c_{1,0}^1 & c_{1,1}^1 & c_{2,-2}^1 & c_{2,-1}^1 & \cdots & c_{O,O}^1 \\
    c_{0,0}^2 & c_{1,-1}^2 & c_{1,0}^2 & c_{1,1}^2 & c_{2,-2}^2 & c_{2,-1}^2 & \cdots & c_{O,O}^2 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
    c_{0,0}^{K-1} & c_{1,-1}^{K-1} & c_{1,0}^{K-1} & c_{1,1}^{K-1} & c_{2,-2}^{K-1} & c_{2,-1}^{K-1} & \cdots & c_{O,O}^{K-1} \\
    c_{0,0}^K & c_{1,-1}^K & c_{1,0}^K & c_{1,1}^K & c_{2,-2}^K & c_{2,-1}^K & \cdots & c_{O,O}^K
\end{bmatrix}, \tag{6.39}
\]

where each column represents the coefficients of a particular order and degree (also referred to as mode) for all possible locations on the sphere where we sample the soundfield. \( I \) is the \((O + 1)^2 \times (O + 1)^2\) identity matrix.

The solution to this system that minimizes the norm of \( c_{o,m}^k \) is found as

\[
C = Y^H (YY^H)^{-1}. \tag{6.40}
\]

### 6.10.3 Order Limit of the Phase Mode Beamformer

The inverse spherical Fourier transformation in (6.31) becomes more precise when increasing the order number \( O \). However, we cannot infinitely increment the order for a finite sampling on the sphere as it happens when using \( K \) sensor elements.

The coefficients found as solution of (6.40) are valid as far as \( YY^H \) presents a condition number not too high. To ensure a \((O + 1)^2\) rank, we need to satisfy \((O + 1)^2 \leq K\); that is, the total number of modes should not be larger than the number of array elements. This limitation ensures that the coefficients \( c_{o,m}^k \) do not suffer of numerical errors.
6.10.4 Beamforming of a Single Plane Wave

To perform beamforming of a single plane wave with an SA of \( K \) elements, we first obtain the spherical harmonics coefficients \( \hat{u}_{o,m}(\kappa, r, \theta_0, \phi_0) \) by applying (6.34) to

\[
u(\kappa, r, \theta_k, \phi_k, \theta_0, \phi_0) = e^{i \kappa a_0 \cdot p_k}, \tag{6.41}
\]

where \( a_0 \) is the unit vector that indicates the DOA of the plane wave defined in (2.7), and \( p_k \) the cartesian coordinates of \((r, \theta_k, \phi_k)\), that is, the location \((k)\) where we measure the plane wave in the sphere.

Then, we apply beamformer weights \( v_{o,m}^*(\kappa, r, \theta_0, \phi_0) \) to each coefficient \( \hat{u}_{o,m}(\kappa, r, \theta_0, \phi_0) \) obtaining the array output as

\[
z(\kappa) = \sum_{o=0}^{O} \sum_{m=-o}^{o} \hat{u}_{o,m}(\kappa, r, \theta_0, \phi_0) v_{o,m}^*(\kappa, r, \theta_0, \phi_0). \tag{6.42}
\]

The beamformer weights are selected to satisfy a particular performance criterion. One of the choices to select the weights [22] when using an open SA that achieves a symmetric beampattern around the SOI direction is [82, 96]

\[
v_{o,m}^*(\kappa, r, \theta_0, \phi_0) = 4\pi Y_{m}^{o}(\theta_0, \phi_0)/(O + 1)^2 b_0(\kappa r). \tag{6.43}
\]

6.10.5 Radius Limit in Phase Mode Beamforming

The wave number and the sphere radius also have an effect in the maximum number of modes we can use in the beamformer. For beamformer weights that use the general expression [22]

\[
v_{o,m}^*(\kappa, r, \theta_0, \phi_0) = d_0 Y_{m}^{o}(\theta_0, \phi_0)/b_0(\kappa r), \tag{6.44}
\]

where \( d_0 \) is a constant value for a particular \( o \), the zeros of the function \( b_0(\kappa r) \) will limit the number of modes that could be used for beamforming. For an open sphere, \( b_0(\kappa r) = 4\pi i^o j_0(\kappa r) \) as defined in (6.33), so the zeros of \( b_0(\kappa r) \) correspond to the zeros of the spherical Bessel function \( j_0(\kappa r) \). Fig. 6.12(a) shows that for a fixed \( \kappa r \), the
larger orders suffer of very low $b_o(\kappa r)$ values. Therefore, using more modes could result in very high magnitude weights, increasing the vector norm and suffering of numerical instabilities. Usually the limiting condition is $\kappa r < O$ [82, 97]. We find it very restrictive so we will use a maximum order $O_{\text{max}}$, so $10\log_{10}(b_{O_{\text{max}}}(\kappa r)) > -12\text{dB}$.

### 6.10.6 Equivalent Element Space Weights for Spherical Harmonics

The beamformer output from phase mode method is

$$z(\kappa) = \sum_{o=0}^{O} \sum_{m=-o}^{o} \hat{u}_{o,m}(\kappa, r, \theta, \phi)v_{o,m}^*(\kappa, r, \theta_0, \phi_0),$$  

where the weights are chosen to satisfy a particular design characteristics. The spherical harmonics coefficient are found as

$$\hat{u}_{o,m}(\kappa, r, \theta, \phi) = \sum_{k=0}^{K} \epsilon_{o,m}^k u(\kappa, r, \theta_k, \phi_k, \theta, \phi),$$  

where $\epsilon_{o,m}^k$ is independent of the frequency ($\kappa = 2\pi f/c$) and only dependent on the array geometry. Replacing (6.46) in (6.45) we obtain
\[ z(\kappa) = \sum_{k=1}^{K} u(\kappa, r_{k}, \phi_{k}, \theta, \phi) \sum_{o=0}^{O} \sum_{m=-o}^{o} c_{o,m}^{k} v_{o,m}^{*}(\kappa, r_{0}, \theta_{0}, \phi_{0}). \]  

Therefore the element space beamformer equivalent weights are

\[ [v]_{k}^{*} = \sum_{o=0}^{O} \sum_{m=-o}^{o} c_{o,m}^{k} v_{o,m}^{*}(\kappa, r_{0}, \theta_{0}, \phi_{0}) \]  

and the beamformer output is

\[ z(\kappa) = \sum_{k=1}^{K} [v]_{k}^{*} u(\kappa, r_{k}, \phi_{k}, \theta, \phi) = v^{H} u, \]  

which express the input output relationship like an element space beamformer. From (6.49) we can obtain the beamformer performance parameters as the DI and the \( A_{w} \) as were defined in section 2.8 of Chapter 2.

6.10.7 Example of Phase Mode Beamforming With Different Sensor Arrangements

Fig. 6.13(a) shows the beampattern from a SA with \( K = 288 \) elements arranged with Gaussian sampling method when using order \( O=10 \) and \( \kappa r = 12.0343 \). The acoustic field

(a) Gaussian sampling spherical array.  (b) Second proposed sampling spherical array.

Figure 6.13: Beampatterns spherical array.

Fig. 6.13(a) shows the beampattern from a SA with \( K = 288 \) elements arranged with Gaussian sampling method when using order \( O=10 \) and \( \kappa r = 12.0343 \). The acoustic field
Table 6.13: Phase Mode Performance Comparison.

<table>
<thead>
<tr>
<th>Beamforming Method</th>
<th>$DI$ (dB)</th>
<th>$A_w$ (dB)</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase mode, Gaussian</td>
<td>20.3582</td>
<td>17.5262</td>
<td>288</td>
</tr>
<tr>
<td>Phase mode, first proposed</td>
<td>20.2822</td>
<td>16.9120</td>
<td>200</td>
</tr>
</tbody>
</table>

is composed of six interferences with DOAs and SIRs from the first data row in Table 6.6. The weight vector is found as in (6.43). The directivity of this beamformer is 20.3582 dB as shown in Table 6.13. Using the second proposed sampling arrangement, with the same $\kappa r$ and scenario, we obtain the beampattern shown in Fig. 6.13(b). This array shows a directivity of 20.2822 dB with an element saving of 30.5%.

6.10.8 Comparison Between Phase Mode and Element Space Beamforming Example

Let’s consider an SA with $\kappa r = 12.0343$ that could correspond to a SA with a working frequency of $f = 1kHz$ and radius $r = 0.6570$. Considering the first proposed sampling method for element beamforming and the second sampling method for phase mode beamforming we have the performance measurements indicated in Table 6.14.

From the table we observe that the element space beamformer have similar $DI$ than
Table 6.14: Performance measurements.

<table>
<thead>
<tr>
<th>Beamforming Method</th>
<th>$DI$ (dB)</th>
<th>$Aw$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Space</td>
<td>20.1378</td>
<td>22.6951</td>
</tr>
<tr>
<td>Phase mode</td>
<td>20.2822</td>
<td>16.9120</td>
</tr>
</tbody>
</table>

Figure 6.15: Directivity index and Array gain vs. $\kappa r$.

that of the phase mode beamformer. The $Aw$ difference is dramatic. The element space beamformer keeps the absolute gain to 186 \textit{i.e.} the number of elements (or 22.6951dB). However, the phase mode beamformer gain is reduced about 6dB because the norm of the phase mode weight vector is much larger.

For different sampling methods and different working frequencies we obtain the results in Fig. 6.15 where the continuous line represents $DI$ and the dashed lines the $Aw$. In Fig. 6.15(a) we observe that the element space beamformer keeps the array gain constant and only dependent on the number of elements (see inlet). The directivity is reduced with the frequency as we expected in these type of beamformers. When considering array nesting the beamformer is able maintain the $DI$ between reasonable bounds.

In Fig. 6.15(b) we observe the opposite behavior (see inlet). The directivity is kept constant and the array gain is reduced with the frequency. The constant directivity is caused by the frequency dependency of the phase mode weight vectors, which make
the beamformer to be frequency invariant. The reduction in the array gain is caused by the frequency dependency of the norm weight vector that dramatically increases for lower frequencies. Fig. 6.12 shows the large variation of $b_0(\kappa r)$ term versus the spherical harmonic orders. We observe that the larger orders suffer of very low $b_0(\kappa r)$ values for low frequencies. Since this term is located in the denominator of the phase mode weights, the norm of the weights is dramatically increased.

### 6.11 Summary

This chapter investigated the feasibility and performance for partial adaptation using SA. First, we investigated the most used sensor arrangements and we suggested two new sensor element arrangements that are amenable for partial adaptive beamforming using rings at different horizontal planes as sub-arrays. The new sensor arrangements keep a reduced number of sensors by maximizing the inter-element distance between sensors in the same ring and also among adjacent rings before spatial aliasing occurs. Also, they maintain high directivity and more symmetrical beampatterns compared to the other arrangements that suffer of high element density in some parts of the array. The high density of elements degrades the beampattern symmetry, does not improve directivity and also does not make efficient use of the number of elements, having more elements than needed to satisfy the Nyquist spatial criterion to avoid aliasing and larger inter-element distance spread.

Secondly, we proposed and analyzed a narrowband partial adaptive beamformer and compared its performance to that of the fully adaptive beamformer. The proposed beamformer considerably reduces the number of complex operations and features faster convergence speed because it has much less coefficient to adapt. The loss of $SINR_0$ is less than 2dB from Monte Carlo simulations, which gives a general behavior of the proposed beamformer.
We also proposed a nesting approach for broadband beamforming of acoustic signals that span its bandwidth several octaves. The beamforming is performed in the frequency domain, where the total bandwidth is separated in sub-bands. At each sub-band, the acoustic field is acquired by re-using particular rings that satisfy the spatial sampling criterion for that sub-band and have an adequate array aperture to consistently maintain the directivity across sub-bands. We apply DFT to each sub-band and perform narrow-band partial adaptive beamforming to each frequency components or bins using a bank composed of the proposed narrowband beamformers. Then, we apply IDFT to obtain the broadband beamformer output. The proposed broadband beamformer maintains the directivity inside 2dB among sub-bands, contrary to the non nesting approach that spans the directivity more than 13dB. Also, the number of elements in each sub-band is consistent and much smaller than that for a non nesting broadband beamformer.

Finally, we analyzed and compare the phase mode beamformer with the element space beamformer. We compare the phase mode beamformers when using different sensor arrangements. We found that the first proposed sensor arrangement uses fewer sensor elements and featuring similar directivity and $SINR_0$ than that of the Gaussian arrangement, which is one of the most used arrangements in phase mode beamforming. We also compared the phase mode with the element space beamformer and found that the phase mode beamformer, at low frequencies, suffers of a huge reduction in array gain compared to that of the element space beamformer. The advantage of the phase mode beamformer is that it maintains the directivity almost constant, featuring frequency invariant beamforming. The element space reduces the directivity with the frequency, so array nesting is necessary to maintain the directivity between reasonable bounds.
Chapter 7

Robustness Against Sensor Position Errors

This chapter is dedicated to study the beamforming robustness against sensor position displacement when using element space beamformers. The term sensor position errors or location errors will be interchangeable. Array location perturbations could happen because manufacturing errors when placing the sensors, because vibration of the array when mounted in planes, etc. The most common robustness method is the quadratic norm constraint, which bounds the squared norm of the weight vector. This technique is also called diagonal loading. In this work, we will study both deterministic unknown location errors as well as random location errors. For the latter case we will obtain a statistical modeling of the sensor perturbations and use it to improve the array robustness.

7.1 Literature Review

One of the first approaches to the design of optimum array with a robustness constraint is published in 1955 by Gilbert and Morgan [98] where they obtain the beampattern as a function of the location errors variance. In 1956 Uzsoky and SolymAr [99] introduce the sensitivity or tolerance factor. In 1967 Tseng and Cheng [100] use an statistical model to maximize the array directivity.
More recently, Jablon in 1986 [101], introduces a method to minimize the leaking in the GSC low branch using optimum injected noise power latter known as diagonal loading. The next year, Cox et al. [52] applied quadratically constrained beamforming and he is cited in several articles as the researcher that coined the term diagonal loading, other articles name Carlson in 1988 [102]. Cox worked together with Owsley that already published similar works some years earlier [103, 104].

In 1994 Feldman, and Griffiths [53] introduce a new method based on the projection of the presumed steering vector onto the observed signal-plus-interference subspace. Er and Ng in the same year [54] propose a iterative method to find the perturbed SOI steering vector by maximizing the array mean output power using a first-order Taylor series approximation. Also in 1994 Fudge and Linebarger [105] propose an array calibration method by deriving the optimal eigenvector-based blocking matrix to minimizes leakage of the desired signal into the noise canceling filter similar to [101]. Bell, Ephraim, and Van Trees in 1996 [55] propose a Bayesian approach with the minimum variance distortionless response (MVDR) criterion to obtain an adaptive beamformer with optimal performance under good conditions and robust to uncertainty in DOA under poor conditions. The same authors in 1997 [72] develop an adaptive beamformer that combines the data driven approach to correct the uncertainties with the robustness of the data independent approach that imposes additional constraints.

Tian et al. in 2001 [106] propose a technique for implementing a quadratic constraint with recursive least squares. A variable diagonal loading term is added at each step where the amount of loading has a closed-form solution. Chang and Chiang in 2002 [107] investigate the application of $H^\infty$ algorithm to imperfect array beamforming with GSC structure.

A Comparison of Robust Adaptive Beamforming Algorithms is performed by Ward et al. on 2003 [108]. In the same year Stoica et al. propose the robust Capon beam-
former [61]. They use the reformulation of the Capon beamforming problem and they append the uncertainty constraint to obtain a robust estimate the SOI power without any intermediate calculation of the steering vector. In 2004 Vincent and O. Besson propose a method using negative diagonal loading [109]. Also in 2004 Li et al. [62], provide a complete analysis of a doubly constrained Capon beamforming, which uses a norm constraint on the weight vector to improve the robustness against array steering vector errors and noise. In 2005 Yan and Ma [110] maximize the array gain against white noise using a second order cone programming (SOCP) to find a robust beamformer. They assume the array location perturbations are Gaussian. In 2006 Elnashar, Elnoubi, and El-Mikati [111] propose an alternative realization of the LCMV adaptive beamforming where the diagonal loading technique is integrated into the adaptive update thus providing a loading-on-demand mechanism rather than fixed loading.

In 2007 and following Li’s article [62], Beck and Eldar [112] propose a double robust Capon beamformer with ellipsoidal uncertainty sets. In 2008, Li et al. [113] proposed a fully automatic robust beamformer based on diagonal loading.

Regarding articles about calibration, Dorny and Meagher in 1980 [114], as well as Ashok and Schultheiss in 1984 [115] propose methods that require auxiliary sources in known locations. In 1987 Y. Rockah et. al, [116, 117] introduce methods that do not require the known location of the sources. in 1991 Chen et al. [118] propose two approaches for bearing estimation based on the Toeplitz and eigenstructure reconstruction of the covariance matrix without the need for calibration. Calibration data can be used to find a calibration matrix as it is found in [119–121]. In 1995, See [122] introduces a two-step calibration procedure that involves a multidimensional search for the sensor positions followed by the solution of a set of linear equations for the calibration matrix. The search space required by this approach is considerably smaller than the estimation of all the parameters simultaneously. In 2007 Lanne et al. [123] use a new method of
calibration correction matrices in the beam pattern optimization. This method leads to the lowest possible uniform side lobe level, for the chosen SNR, beamwidth and beam pointing direction. In 2008 Chung and Wan [124] present a novel procedure for array self-calibration using the Space Alternating Generalized EM algorithm (SAGE). This method is able to simplify the multi-dimensional search procedure required for finding maximum likelihood estimates.

7.1.1 Remarks

As we observed in the above references, the diagonal loading technique is applied to achieve robustness against random errors due to mismatch in the nominal SOI steering vector. This steering vector mismatch could be caused because the true SOI DOA is not the same as that of the nominal SOI. Another cause could be from the true frequency of the SOI being different than the nominal one. Speed propagation mismatch is another cause. Also the mismatch could be caused from random errors in the nominal position of the sensors, e.g. from vibrations. A different case of mismatch happens when the data correlation matrix is different from the true one since we are estimating it. This effect could be caused because low sample size when time constraints do exist.

The robustness from unknown but deterministic sensor error locations is in general approached by array calibration. The calibration is added in the beamformer as a calibration correction matrix that adjust the nominal sensor positions to the exact sensor positions of the array elements.

7.2 Objective

Our objective is to study the effects of array sensor position errors and investigate the possibility of implementing a robust beamforming method that improves the beamformer array gain. For this work we will be using the element space SA implemented in Chapter
7.3 Location Error Model for Element Space Beamformers

One of the causes that create a mismatch in the steering vectors is the sensor position errors. This section obtains the mathematical expressions of the array steering vectors, the array output, and the $SINR_0$ under location errors. These expressions will be used in the analysis on later sections.

7.3.1 Steering Vectors Under Location Errors

The element space beamformer output expression to a narrowband signal with wavenumber magnitude $\kappa$ is

$$z(\kappa, t) = v^H u(\kappa, t), \quad (7.1)$$

where $v$ is the beamformer weight vector and $u(\kappa, t)$ a narrow band input signal at instant $t$ that is composed of $L + 1$ plane waves with wavenumber $\kappa$ plus spatially uncorrelated Gaussian noise. The array input signal is defined in Chapter 2 as (2.19)

$$u(\kappa, t) = s(t)s_0 + \sum_{l=1}^{L} i_l(t)i_l + n(t), \quad (7.2)$$

where $s(t)$, $i_l(t)$ are the SOI and the $l^{th}$ interference complex amplitudes, $s_0$, $i_l$ the corresponding nominal steering vectors, and $n(t)$ the spatially uncorrelated noise vector. Therefore (7.1) becomes

$$z(\kappa, t) = s(t)v^H s_0 + \sum_{l=1}^{L} i_l(t)v^H i_l + v^H n(t). \quad (7.3)$$

The sensor position errors will modify the array nominal steering vectors $s_0$ and $i_l$.
whose mathematical expressions were defined in Chapter 2 (2.14) and (2.17) as

\[
\mathbf{s}_0 = \begin{bmatrix}
    e^{-j\kappa a_0^T p_1} \\
    e^{-j\kappa a_0^T p_2} \\
    \vdots \\
    e^{-j\kappa a_0^T p_K}
\end{bmatrix},
\]

(7.4)

\[
\mathbf{i}_l = \begin{bmatrix}
    e^{-j\kappa a_l^T p_1} \\
    e^{-j\kappa a_l^T p_2} \\
    \vdots \\
    e^{-j\kappa a_l^T p_K}
\end{bmatrix}, \quad l = 1, \ldots, L.
\]

(7.5)

From (7.4) and (7.5) we observe that the nominal steering vectors contain the nominal sensor positions \( \mathbf{p}_k \) defined in Chapter 2, (2.1). The components of \( \mathbf{p}_k = [x_k, y_k, z_k]^T \) indicate the \( k^{th} \) nominal sensor position in Cartesian coordinates. In the presence of location errors, an extra term is added to each of the nominal location coordinates. Here, we assumed the error is additive. The extra term represents the unknown displacement from the nominal position. Therefore, the true location could be expressed as the sum of the nominal position \( \mathbf{p}_k \) plus the displacement \( \Delta \mathbf{p}_k \)

\[
\hat{\mathbf{p}}_k = \mathbf{p}_k + \Delta \mathbf{p}_k, \quad k = 1, \ldots, K.
\]

(7.6)

Also, each coordinate could be expressed from the nominal one plus the displacement value in each coordinate as

\[
\begin{align*}
\hat{x}_k &= x_k + \Delta x_k, \quad k = 1, \ldots, K. \\
\hat{y}_k &= y_k + \Delta y_k, \quad k = 1, \ldots, K. \\
\hat{z}_k &= z_k + \Delta z_k, \quad k = 1, \ldots, K.
\end{align*}
\]

(7.7)

It is important to notice that the displacements \( \Delta x_k, \Delta y_k, \) and \( \Delta z_k \) have different values in the same sensor \( \mathbf{p}_k \) and also among different sensors. The distribution of the displacement could be modeled by a Gaussian random variable with zero mean and variance \( \sigma_p^2 \) [9],

\[
\begin{align*}
\Delta x_k &\sim N(0, \sigma_p^2), \quad k = 1, \ldots, K. \\
\Delta y_k &\sim N(0, \sigma_p^2), \quad k = 1, \ldots, K. \\
\Delta z_k &\sim N(0, \sigma_p^2), \quad k = 1, \ldots, K.
\end{align*}
\]

(7.8)
The true steering vectors are found by replacing (7.6) in (7.4) and (7.5) yielding

\[
\tilde{s}_0 = \begin{bmatrix}
e^{-j\kappa a_0^T \tilde{p}_1} \\
e^{-j\kappa a_0^T \tilde{p}_2} \\
\vdots \\
e^{-j\kappa a_0^T \tilde{p}_K}
\end{bmatrix}.
\] (7.9)

\[
\tilde{i}_l = \begin{bmatrix}
e^{-j\kappa a_l^T \tilde{p}_1} \\
e^{-j\kappa a_l^T \tilde{p}_2} \\
\vdots \\
e^{-j\kappa a_l^T \tilde{p}_K}
\end{bmatrix}, \quad l = 1, \ldots, L.
\] (7.10)

The steering vectors (7.9) and (7.10) include the location errors at each vector component defined in (7.6). After some manipulation, the steering vector components could be written as

\[
[\tilde{s}_0]_k = [s_0]_k e^{-j\kappa a_0^T \Delta p_k}, \quad k = 1, \ldots, K.
\] (7.11)

\[
[\tilde{i}_l]_k = [i_l]_k e^{-j\kappa a_l^T \Delta p_k}, \quad l = 1, \ldots, L, \quad k = 1, \ldots, K.
\] (7.12)

Therefore, each component of the true steering vectors is composed of the nominal steering vector components \([s_0]_k, [i_l]_k\), multiplied by a pure phase factor that is function of the particular displaced sensor position \(\Delta p_k\), the wavenumber magnitude \(\kappa\), and the DOA of the SOI \(a_0\) in (7.11) or that of the \(l^{th}\) interference \(a_l\) in (7.12).

### 7.3.2 Beamformer Input, Output Signals Under Location Errors

The true array signal vector is obtained by replacing the true steering vector in (7.2) yielding

\[
\tilde{u}(\kappa, t) = s(t)\tilde{s}_0 + \sum_{l=1}^{L} i_l(t)\tilde{i}_l + \tilde{n}(t),
\] (7.13)

The first term of (7.13) is the true array input vector from the SOI only

\[
\tilde{u}_s(\kappa, t) = s(t)\tilde{s}_0.
\] (7.14)
The two last terms of (7.13) are the true array input vector from the interference plus noise

\[ \tilde{u}_{in}(\kappa, t) = \sum_{i=1}^{L} i_{i}(t) \tilde{i}_{i} + \tilde{n}(t), \quad (7.15) \]

Using (7.14) and (7.15) in (7.13) and replacing this last expression in (7.1) we obtain the true beamformer output as

\[ \tilde{z}(\kappa, t) = v^{H} \tilde{u}(\kappa, t) = v^{H} \tilde{u}_s(\kappa, t) + v^{H} \tilde{u}_{in}(\kappa, t). \quad (7.16) \]

### 7.3.3 Beamformer Output Signal to Interference Plus Noise Ratio

The $SINR_0$ expression was defined in Chapter 2, (2.46). When including the location errors the expression becomes

\[ SINR_0 = \frac{v^{H} \tilde{R}_s v}{v^{H} \tilde{R}_{in} v}. \quad (7.17) \]

The true correlation matrices $\tilde{R}_s$ and $\tilde{R}_{in}$ have expressions

\[ \tilde{R}_s = \sigma_s^2 \tilde{s}_0 \tilde{s}_0^{H}. \quad (7.18) \]

\[ \tilde{R}_{in} = \sum_{l=1}^{L} \sigma_i^2 \tilde{i}_l \tilde{i}_l^{H} + \sigma_n^2 I. \quad (7.19) \]

This expression will be used in next sections to achieve robustness beamforming.

### 7.3.4 Two Different Types of Sensor Position Errors

In this study we will consider two different cases:

- The perturbation is an unknown but Time Invariant value in each coordinate.

- The perturbation is an unknown random number in each coordinate.

The first case assumes that each sensor in the array is perturbed in each coordinate by a different $\Delta x_k$, $\Delta y_k$, and $\Delta z_k$ values, and those values are invariant in time. Also
these values are different for different \( k \). Therefore the sensors are permanently displaced in the array, but we do not know the exact displacement of each particular sensor. The displacement is assumed to follow a Gaussian distribution \( N(0, \sigma^2_p) \). This type of errors could happen from manufacturing errors when positioning the array sensors, permanent deformations in the array, etc. In general, this case is solved with calibration methods by means of a correction matrix. In the present chapter we introduce a technique that does not use explicit calibration.

The second case assumes that each sensor in the array is perturbed in each coordinate by a different \( \Delta x_k \), \( \Delta y_k \), and \( \Delta z_k \) values and these values are different for different \( k \). However, contrary to the first case, the values change along time. We will assume that the values change at each discrete snapshot satisfying a Gaussian distribution \( N(0, \sigma^2_p) \). Therefore we have an array of sensors that randomly change their locations at each snapshot. This second case could happen when the array is subjected to vibrations, specially in the case of a SA built on an open sphere where the sensors are positioned with thin wires and the array is mounted on a moving vehicle, large pole rod, etc. In both cases we assume to know the amount of error from its variance value \( \sigma^2_p \).

### 7.4 Unknown Time Invariant Location Errors \( \Delta p_k \)

This section considers the sensor position errors \( \Delta p_k \) to be an unknown but time invariant coordinate vector. The proposed robust method is equivalent to maximize an estimate of the \( SINR_0 \) expression with no constraints. The estimate will be obtained from the array data samples. To assess and compare our method with other methods, we will measure the performance of the beamformer with the array gain \( A_g \) defined in subsection 2.8.4 of Chapter 2.
7.4.1 Classical Robustness Method

The classical robustness method, minimizes the beamformer output power defined in Chapter 2 (2.42) repeated here as

\[ P_{\text{sin}} = v^H R_u v. \]  

(7.20)

subject to a distortionless constraint \( v^H s_0 = 1 \) and a quadratic constraint \( \|v\|^2 < U \), where \( U \) is an upper bound on the squared norm of the weight vector. The cost function to minimize is

\[ J = v^H \tilde{R}_u v + \eta_1 (v^H s_0 - 1) + \sigma_{DL}^2 (\|v\|^2 - U), \]

(7.21)

where the first term represents the expected beamformer output power and the second is the distortionless constraint. Finally, the third term is the quadratic constraint on the norm of the weight vector. Forcing to zero the partial derivative of \( J \) with respect \( v^H \) yields

\[ \frac{\partial J}{\partial v^H} = \tilde{R}_u v + \eta_1 s_0 + \sigma_{DL}^2 v = 0, \]

(7.22)

since \( \|v\|^2 = v^H v \). Rearranging the expression we obtain

\[ v = -\eta_1 (\tilde{R}_u + \sigma_{DL}^2 I)^{-1} s_0, \]

(7.23)

The distortionless constraint could be written as \( s_0^H v = 1 \). Premultiplying (7.24) by \( s_0^H \) and rearranging we obtain

\[ -\eta_1 = \frac{1}{s_0^H (\tilde{R}_u + \sigma_{DL}^2 I)^{-1} s_0}. \]

(7.24)

Replacing the value of \( \eta_1 \) from (7.24) into (7.22) we obtain the solution to this minimization problem as

\[ v_{MVDR-DL} = \frac{s_0^H (\tilde{R}_u + \sigma_{DL}^2 I)^{-1}}{s_0^H (\tilde{R}_u + \sigma_{DL}^2 I)^{-1} s_0}, \]

(7.25)
which is the weight vector solution to the MVDR criterion plus a term referred to as injected noise [101] or better known as diagonal loading [9, 52, 102]. The term $\sigma_{DL}^2$ is the amount of diagonal loading and it is related to the quadratic constraint bound $U$. However, the optimal loading level cannot be directly expressed as a function of the $U$ and has to be solved numerically [9, 106]. There are a vast amount of articles searching for the best diagonal loading level as indicated in the first section of this chapter.

### 7.5 Proposed Robust Method for Time Invariant Location Errors

The sensor position errors will invalidate the distortionless beamformer property. In (7.25) the weight vector is designed to satisfy the distortionless constraint expression

$$v^H s_0 = 1. \quad (7.26)$$

However, the true SOI steering vector is now $\tilde{s}_0$. Therefore, the ideal situation would be to use the exact constrain $v^H \tilde{s}_0 = 1$ if we knew $\tilde{s}_0$. The MVDR criterion with quadratic constraint that minimizes the beamformer output power subject to an inexact distortionless constraint will obtain a weight vector that will distort the SOI.

Instead minimizing the beamformer output power subject to an inexact distortionless constraint, we propose a method that minimizes the beamformer output power due to the interference plus noise subject to a distortionless constraint against the true DOA of the SOI. Using Lagrange multipliers, the cost function to minimize is

$$J = v^H \tilde{R}_{in} v + \eta_1(v^H \tilde{R}_s v - \sigma_s^2). \quad (7.27)$$

The first term represents the true beamformer output power due to interference plus noise. The second term represents the true distortionless constraint since $\tilde{R}_s = \sigma_s^2 \tilde{s}_0 \tilde{s}_0^H$. Finding the partial derivative of $J$ with respect $v^H$ and setting it to zero we obtain

$$\frac{\partial J}{\partial v^H} = \tilde{R}_{in} v + \eta_1 \tilde{R}_s v = 0. \quad (7.28)$$
or
\[
\tilde{\mathbf{R}}_{\text{in}} \mathbf{v} = -\eta_1 \tilde{\mathbf{R}}_s \mathbf{v}. \tag{7.29}
\]
This expression is also the solution that minimizes the Rayleigh quotient expressed as
\[
\frac{\mathbf{v}^H \tilde{\mathbf{R}}_{\text{in}} \mathbf{v}}{\mathbf{v}^H \tilde{\mathbf{R}}_s \mathbf{v}}, \tag{7.30}
\]
which is nothing but the inverse of $SINR_0$ defined in (7.17). Therefore our method is equivalent to maximizing the $SINR_0$ unconstrained.

To find the weight vectors we need to solve the expression in (7.29). This expression is known as the generalized eigenvalue problem. In the problem at hand, the weight vector is found from the eigenvector that correspond to the minimum eigenvalue of (7.29).

The obtention of $\tilde{\mathbf{R}}_s$ and $\tilde{\mathbf{R}}_{\text{in}}$ is not trivial so lets manipulate (7.30) to our advantage. The weight vector that minimizes the expression (7.30) should not change if we add a constant as
\[
\frac{1}{SINR_0} + 1 = \frac{\mathbf{v}^H \tilde{\mathbf{R}}_{\text{in}} \mathbf{v}}{\mathbf{v}^H \tilde{\mathbf{R}}_s \mathbf{v}} + \frac{\mathbf{v}^H \tilde{\mathbf{R}}_s \mathbf{v}}{\mathbf{v}^H \tilde{\mathbf{R}}_s \mathbf{v}} = \frac{\mathbf{v}^H \tilde{\mathbf{R}}_{\text{in}} \mathbf{v}}{\mathbf{v}^H \tilde{\mathbf{R}}_s \mathbf{v}}. \tag{7.31}
\]
The numerator of (7.31) contains the array input signal correlation matrix that could be estimated from the data correlation matrix defined as
\[
\mathbf{C}_u = \frac{1}{T} \sum_{t=1}^{T} \tilde{\mathbf{u}}(\kappa, t) \tilde{\mathbf{u}}^H(\kappa, t), \tag{7.32}
\]
where $T$ is the number of snapshots considered in the estimation of $\mathbf{C}_u$. On the other hand, finding a good estimate of $\tilde{\mathbf{R}}_s$ is not as easy. In theory, we could obtain an estimate from
\[
\mathbf{C}_s = \frac{1}{T} \sum_{t=1}^{T} \tilde{\mathbf{u}}_s(\kappa, t) \tilde{\mathbf{u}}_s^H(\kappa, t), \tag{7.33}
\]
but $\tilde{\mathbf{u}}_s$ is the SOI under location errors and in general it is not available. This remains an open problem that would be solved in the next section.
Therefore, the Rayleigh quotient to minimize is
\[
\frac{v^H C_u v}{v^H C_s v},
\] (7.34)
which is solved by finding the generalized eigenvector that correspond to the minimum eigenvalue of the expression
\[
C_u v = -\eta_1 C_s v,
\] (7.35)

7.5.1 Example. Ideal case: Known \( C_s \)

A) Robustness Against \( \sigma_p^2 \)

To illustrate the validity of the proposed method to find a robust beamformer weight vector, we will find its array gain expression and compare it with other methods. In particular those that use the delay-and-sum weights, the MVDR criterion and the MVDR-DL. The latter being a robust method.

The weight vector expressions for each method are shown in Table 7.1 where \( s_0 \) is the nominal SOI steering vector, \( K \) is the total number of array elements, and \( C_u, C_s \) are the data correlation matrices obtained from \( T \) snapshots of data and they are used to estimate \( \tilde{R}_u, \tilde{R}_s \) respectively. Finally, \( \sigma_{DL}^2 \) is amount of diagonal loading.

To measure the beamformer performance we will find the array gain expression de-
fined in subsection 2.8.4 of Chapter 2 and repeated here as

$$A_g = \frac{\mathbf{v}^H \hat{\mathbf{R}}_s \mathbf{v}}{\mathbf{v}^H \hat{\mathbf{R}}_{in} \mathbf{v}} / \text{SINR}_i,$$

(7.36)

where $\hat{\mathbf{R}}_s$ and $\hat{\mathbf{R}}_{in}$ are the true correlation matrices of the SOI and interference plus noise defined in (7.18) and (7.19). $\text{SINR}_i$ is the signal to interference plus noise ratio at the array input. The array used is the first proposed SA. The scenario is composed of a SOI with DOA $(\theta_0 = 90^\circ, \phi_0 = 0^\circ)$ and $\text{SNR} = 0\text{dB}$, six interferences with DOAs, $(\theta_0 = 75^\circ, \phi_0 = 120^\circ)$, $(\theta_0 = 129.4^\circ, \phi_0 = 162^\circ)$, $(\theta_0 = 65^\circ, \phi_0 = 230^\circ)$, $(\theta_0 = 125^\circ, \phi_0 = 260^\circ)$, $(\theta_0 = 50^\circ, \phi_0 = 220^\circ)$, and $(\theta_0 = 170^\circ, \phi_0 = 90^\circ)$ with INR of 15, 14, 17, 12, 16, and 13dB respectively.

![Figure 7.1: Expected Array gain vs. $\sigma_p^2$.](image)

Fig. 7.1 shows the expected array gain versus the variance of location errors for the different beamforming methods in Table 7.1. The expectation of the array gain in (7.36) is performed with 50 ensemble averages where for each particular ensemble the sensor locations are fixed during all snapshots in a particular ensemble, but at each ensemble the array sensors are placed in different locations.

The data correlation matrices $\mathbf{C}_s$ and $\mathbf{C}_u$ are obtained from $T = 600$ data snapshots
in Fig. 7.1(a) and $T = 250$ data snapshots in Fig. 7.1(b). Where at each snapshot $t$ the complex amplitude of the SOI $s(t)$, those of the interference $i_l(t)$ and the noise vector $\tilde{n}(t)$ in (7.13) are Gaussian random values with variances $\sigma^2_s$, $\sigma^2_i$, and $\sigma^2_n$.

The trace with solid line and marker 'x' is the maximum array gain that would be obtained if we exactly knew the location errors.

The array gain from the beamformer that obtained the weights with the MVDR criterion is shown in solid line and marker 'o'. The mathematical expression is located at the second data row of Table 7.1. We observe in Fig. 7.1(a) that for small $\sigma^2_p$, the maximum array gain is around 1dB below the ideal array gain. This is because the number of snapshots $T$ to estimate $\tilde{R}_u$ is not very large and the estimator $C_u$ is somewhat different than $\tilde{R}_u$. For larger $T$ the array gain will approach the ideal array gain, and for smaller $T$ the array gain will be even smaller as shown in Fig. 7.1(b). We also observe that in both figures the degradation in the array gain starts very quickly even for small perturbation variances. This happens because the high sensitivity of the MVDR criterion caused from the estimation of the true correlation matrix $\tilde{R}_u$ and also from the inexact $s_0$ used to obtain the weight vector expression.

The array gain from the diagonal loaded MVDR (MVDR-DL) using $C_u$ and $s_0$ is shown in dashed line and marker 'o'. The mathematical expression is at the third data row of Table 7.1. We observe that there is robustness with respect the variance of the location errors as far as $\sigma^2_p = 1.4 \times 10^{-5}$ as we expected from this robust method [9]. Interestingly enough the diagonal loading method is also robust against low sample size scenario as shown in Fig. 7.1(b) where it keeps almost the same levels as in (a). The cost of the robustness is an initial loss of array gain in both cases.

The light dashed line with diamond markers shows the array gain from delay-and-sum weights that is almost as robust as the MVDR-DL but with smaller gain since it does not target the interferences.
Finally the dashed line with square markers shows the proposed method. We observe that both in Fig. 7.1(a) and (b) the array gain is similar to that of the MVDR method for small perturbations. This is because for small amount of perturbations, $C_s$ is not too different than the nominal $R_s$ and the minimization of (7.35) becomes the MVDR criterion. In Fig. 7.1(a) the proposed method outperforms the diagonal loaded robust method by 3.5dB. The most interesting behavior is that the array gain is almost independent of the location errors. This happens because the optimization is done solely from the data correlation matrices $C_s$ and $C_u$ that include the location errors, in contrast to the previous methods that used the nominal but incorrect $s_0$. Using different scenarios of interference and noise render similar results.

B) Robustness Against $SINR_i$

![Graph](image)

(a) $T = 600$. Medium sample size scenario. (b) $T = 250$. Low sample size scenario.

Figure 7.2: Array gain vs. $SINR_i$.

The array robustness is also measured against the signal to interference plus noise at the beamformer input [9]. The reason is that before beamforming we need to estimate the DOA of the SOI. These estimation methods work better for large $SINR_i$. Therefore it is important to achieve robustness for large $SINR_i$. 

159
Fig. 7.2(a) and (b) shows the expected array gain versus the $SINR_i$ where the sensor position error variance is fixed to $\sigma_p^2 = 10^{-4}$. The expectation is performed with 50 ensemble averages where each ensemble have unknown but constant sensor positions for all snapshots. We observe that the beamformer whose weights were obtained with the MVDR criterion degrades its array gain for very low values of $SINR_i$. With low sample size this effect is more dramatic. The robust beamformer that uses the MVDR-DL criterion is able to keep the array gain constant up to $SINR_i = -20$dB in both medium and low sample size. The array gain from the beamformer using delay-and-sum weights is constant since the weight vector does not depend on the $SINR_i$. Finally the proposed robust method also keeps the array gain constant against the $SINR_i$.

### 7.5.2 Example. Practical case: Known $s_0$ and $\sigma_p^2$ Using SA

<table>
<thead>
<tr>
<th>Method.</th>
<th>weight vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay-and-sum</td>
<td>$v = \frac{s_0}{R}$</td>
</tr>
<tr>
<td>MVDR from data correlation</td>
<td>$v = \frac{C_u^{-1}s_0}{s_0^H C_u^{-1}s_0}$</td>
</tr>
<tr>
<td>MVDR from data correlation with DL</td>
<td>$v = \frac{(C_u + \sigma_{DL}^2 I)^{-1}s_0}{s_0^H (C_u + \sigma_{DL}^2 I)^{-1}s_0}$</td>
</tr>
<tr>
<td>Proposed robust</td>
<td>$v : C_{in}v = \eta \tilde{\rho}_s v$</td>
</tr>
</tbody>
</table>

It is important to notice that the $C_s$ matrix is seldom available in most cases. If we repeat the last experiment but using an estimate of $\tilde{\rho}_s$ as $\tilde{\rho}_s = \varepsilon s_0 s_0^H + (1 - \varepsilon)I$ with $\varepsilon$ a known parameter, as it will be introduced in the next section, we obtain the results shown in Fig. 7.3 and Fig. 7.4 where the only changes with respect Fig. 7.1 and Fig. 7.2 are the draw traces of the array gain from the proposed robust beamformer. The above figures were obtained with $\varepsilon = e^{-0.5369}$. We observe in Fig. 7.3(a) that for $\sigma_p^2 \leq 10^{-4}$ the behavior of the proposed beamformer that uses the new estimator $\tilde{\rho}_s$ is similar to
the same in Fig. 7.1 that used the sample correlation matrix $C_s$ for larger variances the array gain degrades. In any case, the proposed beamformer outperforms by about 3dB the one that uses the MVDR-DL criterion and also extends the robustness from $\sigma_p^2 = 0.14 \times 10^{-4}$ to $\sigma_p^2 = 1.00 \times 10^{-4}$. For low sample size, shown in Fig. 7.3(b) the beamformer is outperformed by the one using the MVDR-DL about 2dB.

Figure 7.4: Array gain vs. $SINR_i$. 

(a) $T = 600$. Medium sample size scenario.

(b) $T = 250$. Low sample size scenario.
Interestingly enough Fig. 7.4 shows that the array gain of the proposed robust beamformer versus $SINR_i$ is constant. The explanation of this behavior resides in the constant norm of the weight vector with respect the $SINR_i$. The beamformer that uses MVDR and MVDR-DL criteria to obtain the weights do not have this property, i.e. the norm of the weight vectors is constant only up to certain $SINR_i$ value. For larger $SINR_i$ values, the norm starts increasing and the array gain starts decreasing. This is caused by the incorrect distortionless constraint that increases with the $SINR_i$ even for the diagonal loaded robust method.

7.5.3 Example. Practical case: Known $s_0$ and $\sigma_p^2$ Using CRA

Repeating the last examples using the CRA implemented in Chapter 3 section 3.4 we obtain the results shown in Fig. 7.5 where we observe similar behavior as in the last example considering that this array has a different number of elements so the MVDR ideal array gain is different.

Figure 7.5: $T = 600$. Medium sample size scenario.
7.6 Modeling of Time Variant Random Location Errors $\Delta p_k$

This section considers the sensor position errors $\Delta p_k : \Delta p_k(t)$ to be an unknown random vector that changes at every snapshot with a Gaussian distribution.

We consider each displacement coordinate with respect the time as a Gaussian distributed random variable, with same variance at each coordinate,

\[
\begin{align*}
\Delta x_k(t) & : N(0, \sigma^2_p), \quad k = 1, ..., K, \quad t = 1, ..., T. \\
\Delta y_k(t) & : N(0, \sigma^2_p), \quad k = 1, ..., K, \quad t = 1, ..., T. \\
\Delta z_k(t) & : N(0, \sigma^2_p), \quad k = 1, ..., K, \quad t = 1, ..., T.
\end{align*}
\] (7.37)

The sensors are subjected to a perturbation that will change its position at each snapshot. In this subsection we analyze the expectation of the beamformer output signal, the beampattern, and the $\tilde{SINR}_0$. The variable $t$ will be dropped from the upcoming expressions for simplicity.

7.6.1 Expected Beamformer Output Under Location Errors

Under random sensor position errors, expected value of the element space beamformer output is obtained by taking expectation of the expression in (7.16) as

\[
E[\tilde{z}(\kappa, t)] = v^H E[\tilde{u}(\kappa, t)],
\] (7.38)

where the expected beamformer input signal vector $E[\tilde{u}(\kappa, t)]$ incorporates the sensor position errors.

Considering the array input a single plane wave $\tilde{u}(\kappa, t) = u(t)\tilde{s}_0$ we have

\[
E[\tilde{u}(\kappa, t)] = u(t) \begin{bmatrix} E[\tilde{s}_0]\end{bmatrix}_1 \begin{bmatrix} E[\tilde{s}_0]\end{bmatrix}_2 \cdots \begin{bmatrix} E[\tilde{s}_0]\end{bmatrix}_K \end{bmatrix}^T,
\] (7.39)

since the expectation of a vector is the expectation of each vector component. In (7.39) $u(t)$ is the complex amplitude of the plane wave, and $[\tilde{s}_0]_k$ the $k^{th}$ component of the steering vector $\tilde{s}_0$. From (7.11) we obtain the expectation as,

\[
E[\tilde{s}_0]_k = [s_0]_k E[e^{-j\kappa a_0^T \Delta p_k}]. \; k = 1, ..., K.
\] (7.40)
The terms $-\kappa a_0^T \Delta p_k$, $k = 1, ..., K$, in (7.40) are assumed to be i.i.d. random variables with statistical distribution

$$-\kappa a_0^T \Delta p_k : N(0, \kappa^2 \sigma_p^2), \quad k = 1, ..., K, \quad (7.41)$$

as shown in Appendix E. Assuming $E[e^{-j\kappa a_0^T \Delta p_k}]$ is a constant term in all vector components we obtain the expectation of the steering vector as

$$E[\tilde{s}_0] = s_0 e^{-\kappa^2 \sigma_p^2}. \quad (7.42)$$

From (7.42) and (7.39) we obtain,

$$E[\tilde{u}(\kappa, t)] = u(t)s_0 E[e^{-j\kappa a_0^T \Delta p_k}] = u(\kappa, t)E[e^{-j\kappa a_0^T \Delta p_k}], \quad (7.43)$$

Indeed, the term $E[e^{-j\kappa a_0^T \Delta p_k}]$ is constant and could be computed using the characteristic function of the Gaussian distribution that satisfies

$$M_X(jt) = E[e^{jtX}] = e^{j(t\mu - \frac{\sigma^2}{2})}, \quad (7.44)$$

where $X : N(\mu, \sigma^2)$ [125]. For $t = 1$ we have $E[e^{jX}] = e^{(j\mu - \frac{\sigma^2}{2})}$. In the problem at hand the random variable $X$ is (7.41). Therefore (7.42) becomes

$$E[\tilde{s}_0] = s_0 e^{-\kappa^2 \sigma_p^2}. \quad (7.45)$$

Finally, (7.43) becomes

$$E[\tilde{u}(\kappa, t)] = u(\kappa, t)e^{-\kappa^2 \sigma_p^2}. \quad (7.46)$$

From (7.46) we observe that the expected value of the array input signal is affected by a positive real valued term that is frequency dependent through $\kappa$ and less or equal to one. For a particular variance in the location errors and a constant propagation speed, the term becomes smaller as the frequency increases.

The expected beamformer output expression from (7.38) and (7.46) becomes

$$E[\tilde{z}(\kappa, t)] = \mathbf{v}^H u(\kappa, t)e^{-\kappa^2 \sigma_p^2}. \quad (7.47)$$
From (7.47) we observe that a particular frequency component of the beamformer output will be differently attenuated with more emphasis to the higher frequencies.

7.6.2 Expected Beampattern Under Location Errors

The expected beampattern is found as

\[ E[\tilde{\mathbf{B}}(a_s)] = E[|\mathbf{v}\tilde{s}_s|^2], \]

where \( \tilde{s}_s \) is the perturbed steering vector towards the DOA \( a_s \) that scans the 3-D field. 

(7.48) could be expanded as [9]

\[ E[\tilde{\mathbf{B}}(a_s)] = \sum_{k=1}^{K} \sum_{k'=1}^{K} v_k^* v_{k'} e^{j\kappa a_s^T p_k} e^{-j\kappa a_s^T p_{k'}} e^{j\kappa a_s^T \Delta p_k} e^{-j\kappa a_s^T \Delta p_{k'}}. \]  

(7.49)

The summation has different values for \( k \neq k' \) and \( k = k' \). Therefore we end up with two terms as

\[ E[\tilde{\mathbf{B}}(a_s)] = \sum_{k=1}^{K} \sum_{k'=1}^{K} v_k^* v_{k'} e^{j\kappa a_s^T (p_k - p_{k'})} E[e^{j\kappa a_s^T (\Delta p_k - \Delta p_{k'})}] + \|v\|^2. \]  

(7.50)

Using the same relation as in (7.43) but considering that the phase of \( E[e^{j\kappa a_s^T (\Delta p_k - \Delta p_{k'})}] \) is the sum of two Gaussian variables we end up with

\[ E[\tilde{\mathbf{B}}(a_s)] = \sum_{k=1}^{K} \sum_{k'=1}^{K} v_k^* v_{k'} e^{j\kappa a_s^T (p_k - p_{k'})} e^{-\kappa^2 \sigma_p^2} + \|v\|^2. \]  

(7.51)

The first part of (7.50) is missing one term to become the nominal beampattern \( \mathbf{B}(a_s) \). Adding and subtracting that term we obtain the same result as Van Trees in [9]

\[ E[\tilde{\mathbf{B}}(a_s)] = \mathbf{B}(a_s) e^{-\kappa^2 \sigma_p^2} + \|v\|^2 (1 - e^{-\kappa^2 \sigma_p^2}). \]  

(7.52)

The effect of the first term in (7.52) is that the beampattern is attenuated similarly as the beamformer output was in (7.47). The second term adds a positive constant value to the beampattern, thus raising the whole beampattern and more important, the sidelobe levels. This could be a serious problem if there are strong interferences in the field, since
we expect the beampattern to be zero at the interferences DOAs. The norm of the weight vector is crucial to reduce this effect, we aim for a weight vector with minimum norm so the offset is reduced. From section 2.8.3 of Chapter 2, the array gain against spatially white noise expression (2.64) is $A_w = \|v\|^{-2}$. Therefore, the larger the array gain against spatially white noise, the smaller the sensitivity of the beampattern to location errors will be [9].

7.7 Proposed Robust Beamformer for Time Variant Location Errors

This section propose a robust beamforming method for sensor position errors that change each snapshot. First, we will obtain the spatial correlation matrices function of the statistical variance of the sensor position errors. Then, we will obtain the $SINR_0$ expression. The proposed beamformer applies a distortionless constraint that includes the sensor position errors, contrary to the classical diagonal loading technique that do not include the sensor errors in the distortionless constraint. We will find that the proposed robust method is equivalent to maximize the $SINR_0$ expression where the $\tilde{R}_i$ has been diagonal loaded. To asses and compare the performance of the proposed robust beamformer with other beamforming methods including the classical diagonal loading, we will use the array gain $A_g$ versus both the $\sigma_p^2$ and the $SINR_i$.

7.7.1 Correlation Matrices Under Location Errors

The correlation matrix expressions were defined in subsection 2.5.3 of Chapter 2. We will consider first that there is only one interference in the acoustic field and later we will consider the case of several interferences. For one interference only, the correlation matrices are

$$
\tilde{R}_s = E[\tilde{u}_s(\kappa, t)\tilde{u}_s^H(\kappa, t)],
$$

(7.53)
where $\tilde{\mathbf{u}}(\kappa, t) = s(t)\tilde{s}_0$ is the SOI and the interference is $\tilde{\mathbf{u}}_{i_1}(\kappa, t) = i_1(t)\tilde{i}_1$. The correlation matrices can be expressed as

$$\tilde{\mathbf{R}}_s = \sigma_s^2 E[\tilde{s}_0\tilde{s}_0^H], \quad (7.55)$$

$$\tilde{\mathbf{R}}_{i_1} = \sigma_{i_1}^2 E[\tilde{i}_1\tilde{i}_1^H]. \quad (7.56)$$

The expressions $E[\tilde{s}_0\tilde{s}_0^H]$ and $E[\tilde{i}_1\tilde{i}_1^H]$ are found similarly. Let’s find the first one as

$$E[\tilde{s}_0\tilde{s}_0^H] = E\left[ \begin{bmatrix} e^{-\kappa\alpha_0^T\tilde{p}_1} \\ e^{-\kappa\alpha_0^T\tilde{p}_2} \\ \vdots \\ e^{-\kappa\alpha_0^T\tilde{p}_K} \end{bmatrix} \begin{bmatrix} e^{\kappa\alpha_0^T\tilde{p}_1} & e^{\kappa\alpha_0^T\tilde{p}_2} & \ldots & e^{\kappa\alpha_0^T\tilde{p}_K} \end{bmatrix} \right]. \quad (7.57)$$

Operating the expression becomes

$$E[\tilde{s}_0\tilde{s}_0^H] = \begin{bmatrix} 1 \\ E[e^{-\kappa\alpha_0^T(\tilde{p}_2-\tilde{p}_1)}] \\ \vdots \\ E[e^{-\kappa\alpha_0^T(\tilde{p}_K-\tilde{p}_1)}] \end{bmatrix} \begin{bmatrix} 1 & \ldots & 1 \\ E[e^{-\kappa\alpha_0^T(\tilde{p}_2-\tilde{p}_1)}] & \ldots & E[e^{-\kappa\alpha_0^T(\tilde{p}_K-\tilde{p}_2)}] \\ \vdots & \ldots & \vdots \end{bmatrix}. \quad (7.58)$$

Each non-diagonal term of (7.58) is found as

$$E[e^{-\kappa\alpha_0^T(\tilde{p}_k-\tilde{p}_{n'})}] = e^{-\kappa\alpha_0^T(p_k-p_{n'})}e^{-\kappa^2\sigma_p^2}, \quad (7.59)$$

After some manipulations we find

$$E[\tilde{s}_0\tilde{s}_0^H] = e^{-\kappa^2\sigma_p^2}E[s_0s_0^H] + (1 - e^{-\kappa^2\sigma_p^2})I. \quad (7.60)$$

Then, expression (7.55) becomes

$$\tilde{\mathbf{R}}_s = e^{-\kappa^2\sigma_p^2}\mathbf{R}_s + \sigma_s^2(1 - e^{-\kappa^2\sigma_p^2})I. \quad (7.61)$$

Similarly, the expression (7.56) becomes

$$\tilde{\mathbf{R}}_{i_1} = e^{-\kappa^2\sigma_p^2}\mathbf{R}_{i_1} + \sigma_{i_1}^2(1 - e^{-\kappa^2\sigma_p^2})I. \quad (7.62)$$
When there is more than one interference, the expression (7.62) can be generalized to

$$\tilde{R}_i = e^{-\kappa^2 \sigma_p^2} R_i + \sum_{i=1}^{L} \sigma_{n_i}^2 (1 - e^{-\kappa^2 \sigma_p^2}) I_i$$  (7.63)

where we assumed the interferences are generated independently of each other, so $\sigma_i \sigma_{l'} = 0$ for $l \neq l'$.

### 7.7.2 $\text{SINR}_0$ Under Location Errors

The $\text{SINR}_0$ defined in Chapter 2 (2.48), under location errors situation becomes

$$\tilde{\text{SINR}}_0 = \frac{v^H \tilde{R}_s v}{v^H \tilde{R}_v v + \sigma_n^2 \|v\|^2}.$$  (7.64)

Replacing the correlation matrices, the expression (7.64) becomes

$$\tilde{\text{SINR}}_0 = \frac{e^{-\kappa^2 \sigma_p^2} v^H R_s v + \sigma_n^2 (1 - e^{-\kappa^2 \sigma_p^2}) \|v\|^2}{e^{-\kappa^2 \sigma_p^2} v^H R_i v + [\sigma_n^2 + \sum_{l=1}^{L} \sigma_{l_i}^2 (1 - e^{-\kappa^2 \sigma_p^2})] \|v\|^2},$$  (7.65)

or

$$\tilde{\text{SINR}}_0 = \frac{v^H \tilde{R}_s v}{v^H \tilde{R}_m v},$$  (7.66)

where $\tilde{R}_m$ is the true correlation of the interferences plus noise, which after some manipulation could be expressed as

$$\tilde{R}_m = e^{-\kappa^2 \sigma_p^2} R_m + (1 - e^{-\kappa^2 \sigma_p^2}) \left( \sigma_n^2 + \sum_{l=1}^{L} \sigma_{l_i}^2 \right) I.$$  (7.67)

In (7.67), $R_m$ is the nominal correlation of the interference plus noise defined in subsection 2.5.3 of Chapter 2.

The correlation matrix of the SOI plus interferences and noise is found by adding (7.67) and (7.61). After some mathematical manipulation we obtain

$$\tilde{R}_u = e^{-\kappa^2 \sigma_p^2} R_u + (1 - e^{-\kappa^2 \sigma_p^2}) \left( \sigma_s^2 + \sum_{l=1}^{L} \sigma_{l_i}^2 + \sigma_n^2 \right) I.$$  (7.68)

Lets analyze the bounds of (7.66). For low frequencies or low sensor perturbation variance, the term $e^{-\kappa^2 \sigma_p^2} \to 1$ so $\tilde{\text{SINR}}_0 \to \text{SINR}$. For high frequencies or high noise variance, the term $e^{-\kappa^2 \sigma_p^2} \to 0$ so $\tilde{\text{SINR}}_0 \to 0$. Therefore, the $\text{SINR}_0$ is bounded by 0 and 1.
power \( e^{-\kappa^2 \sigma_p^2} \to 0 \) and

\[
SI\tilde{N}R_0 \to \frac{\sigma_s^2}{[\sigma_n^2 + \sum_{l=1}^L \sigma_{n_l}^2]} = SINR_i,
\]

(7.69)

where we observe the beamformer becomes useless.

### 7.7.3 Proposed Method for Robustness

As already indicated in section 7.5, the MVDR-DL criterion to find the weights uses an inexact distortionless constraint \( \mathbf{v}^H \mathbf{s}_0 = 1 \). In this subsection we will explore the possibility of improving the diagonal loaded beamformer by imposing a statistical distortionless criterion.

The cost function to minimize is

\[
J = \mathbf{v}^H \tilde{\mathbf{R}}_u \mathbf{v} + \eta_1 (\mathbf{v}^H \tilde{\mathbf{R}}_s \mathbf{v} - \sigma_s^2) + \sigma_{DL}^2 (\|\mathbf{v}\|^2 - U),
\]

(7.70)

where the first two terms represents the expected beamformer output power due to interference plus noise and the expected distortionless constraint. The third term is the quadratic constraint on the norm of the weight vector, that is used in diagonal loading.

Forcing to zero the partial derivative of \( J \) with respect \( \mathbf{v}^H \) yields

\[
\frac{\partial J}{\partial \mathbf{v}^H} = \tilde{\mathbf{R}}_u \mathbf{v} + \eta_1 \tilde{\mathbf{R}}_s \mathbf{v} + \sigma_{DL}^2 \mathbf{v} = 0,
\]

(7.71)

since \( \|\mathbf{v}\|^2 = \mathbf{v}^H \mathbf{v} \). Rearranging the expression we obtain

\[
(\tilde{\mathbf{R}}_u + \sigma_{DL}^2 \mathbf{I}) \mathbf{v} = \eta'_1 \tilde{\mathbf{R}}_s \mathbf{v}.
\]

(7.72)

Likewise we did in (7.30) to (7.34) the minimization of (7.72) is equivalent to the minimization of the Rayleigh quotient

\[
\frac{\mathbf{v}^H (\tilde{\mathbf{R}}_u + \sigma_{DL}^2 \mathbf{I}) \mathbf{v}}{\mathbf{v}^H \tilde{\mathbf{R}}_u \mathbf{v}},
\]

(7.73)
which is solved by finding the generalized eigenvector that correspond to the minimum eigenvalue of the expression

\[(\tilde{R}_u + \sigma_{DL}^2 I)v = \eta \tilde{R}_s v, \quad (7.74)\]

The eigenvector corresponding to the smaller eigenvalue is weight vector solution for our proposed robust beamformer.

The correlation matrices in (7.74) have the expressions shown in (7.61) and (7.68). In particular (7.61) could be expressed as

\[\tilde{R}_s = \sigma_s^2 \left( e^{-\kappa^2 \sigma_P^2} s_0 s_0^H + (1 - e^{-\kappa^2 \sigma_P^2}) I \right), \quad (7.75)\]

where the nominal \(R_s\) has been replaced by its value \(R_s = \sigma_s^2 s_0 s_0^H\). Replacing (7.75) in (7.74) we notice that the term \(\sigma_s^2\) could be included in \(\eta\), yielding

\[(\tilde{R}_u + \sigma_{DL}^2 I)v = \eta' \left( \varepsilon s_0 s_0^H + (1 - \varepsilon) I \right)v, \quad (7.76)\]

where \(\varepsilon = e^{-\kappa^2 \sigma_P^2}\). The term in parenthesis is referred to as the normalized version of \(\tilde{R}_s\) with respect the SOI power \(\sigma_s^2\)

\[\tilde{\rho}_s = \left( \varepsilon s_0 s_0^H + (1 - \varepsilon) I \right). \quad (7.77)\]

This is a very interesting expression since it is entirely composed of known parameters \(i.e.,\) the nominal steering vector \(s_0\), the wavenumber \(\kappa\), and the variance of the sensor position errors \(\sigma_P^2\) through \(\varepsilon\). Therefore by solving (7.76) instead (7.74) we replaced the difficult problem of estimating \(\tilde{R}_s\) by the easier task of obtaining \(\tilde{\rho}_s\). We can also replace \(\tilde{\rho}_s\) instead \(\tilde{R}_s\) or \(C_s\) in the previous section expressions (7.31) and (7.35) to obtain the weight vector of the proposed robust beamformer for unknown Time Invariant sensor position errors.

With respect \(\tilde{R}_u\), we could obtain (7.68) if we knew the nominal correlation matrix of the beamformer input signal

\[R_u = \sigma_s^2 s_0 s_0^H + \sum_{t=1}^L \sigma_t^2 i_t i_t^H + \sigma_n^2 I. \quad (7.78)\]
This is a difficult task, since we would need to estimate the power of the SOI $\sigma_s^2$, those of all interferences $\sigma_l^2$, and that of the noise $\sigma_n^2$. Also, we would need to find a good estimate of the DOA of all the plane wave signals $a_0$ and $a_i$, to obtain the nominal steering vectors $s_0$ and $i_i$. A better option is to estimate $\tilde{R}_u$ with the data correlation matrix $C_u$. Therefore, the expression used to find the beamformer weights is

$$(C_u + \sigma_{DL}^2 I)v = \eta' \tilde{\rho}_sv,$$ (7.79)

### 7.7.4 Example. Robustness Against Time Variant Location Errors

Table 7.3: Weight vectors expressions for time variant location errors

<table>
<thead>
<tr>
<th>Method</th>
<th>Weight Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay-and-sum</td>
<td>$v = \frac{s_0}{\bar{K}}$</td>
</tr>
<tr>
<td>MVDR from data correlation</td>
<td>$v = \frac{C_u^{-1}s_0}{s_0^H C_u^{-1}s_0}$</td>
</tr>
<tr>
<td>MVDR from data correlation with DL</td>
<td>$v = \frac{(C_u + \sigma_{DL}^2 I)^{-1}s_0}{s_0^H (C_u + \sigma_{DL}^2 I)^{-1}s_0}$</td>
</tr>
<tr>
<td>Proposed robust</td>
<td>$v : (C_u + \sigma_{DL}^2 I)v = \eta' \tilde{\rho}_sv$</td>
</tr>
</tbody>
</table>

### A) Robustness Against $\sigma_p^2$

To explore the validity of the proposed method, we will find its array gain expression and compare it with other methods whose weight vector expressions are shown in Table 7.3. The scenario is the same as in the previous examples in subsection 7.5.1. The beamformer weights are found by solving the expression (7.79).

Fig. 7.6(a)(b) shows the array gain versus the variance of location errors for the different beamforming methods in Table 7.3. From Fig. 7.6(a) we observe that there has been almost 2dB improvement with respect to the MVDR beamformer but almost 3.5dB with respect to the robust MVDR-DL beamformer. In Fig. 7.6(b) the MVDR beamformer
gain is reduced because the low sample size. However, both the MVDR-DL and the proposed robust beamformer keep the same array gain as in Fig. 7.6(a). Therefore, the proposed robust beamformer maintains the same 3.5dB improvement level with respect the MVDR-DL beamformer.

B) Robustness Against $SINR_i$

Fig. 7.7 shows the array gain versus the $SINR_i$. We observe that in both cases, the proposed beamformer outperform the classical robust beamformer by 3dB, and maintains a consistent high array gain up to $SINR_i$ of $-3dB$ where the array gain starts decreasing. For $SINR_i$ between $-3dB$ and 2dB the classical robust beamformer maintains its array gain level. Therefore, for that high $SINR_i$ values the proposed beamformer is outperformed by the classical one.

7.8 Summary

This chapter investigated the problem of array sensor position errors in beamforming under two perspectives. The first perspective assumed that the location errors are time
invariant but unknown. In practice, each implemented array will have an unknown displacement value in each of the sensors, e.g. due to manufacturing defects. Different arrays will have different sensor locations. The manufacturing defects in the displacement of the sensors are assumed to have a Gaussian probability density function with known variance. We propose a beamformer that takes the sensor position errors into account by maximizing the $SINR_0$. The proposed beamformer shows a reasonable improvement (of 3dB in our simulations with the SA) and extends the robustness range ($\sigma_p^2$ about one order of magnitude) with respect the classical robust beamformer that uses diagonal loading technique.

The second perspective assumed that the array is subjected to perturbations in the sensor positions that vary from snapshot to snapshot. This is a more difficult problem to solve. We propose an improvement of the diagonal loading beamformer with a new distortionless constraint, which includes the statistical information about the sensor perturbations. The proposed new robust beamformer provides a good level of improvement with respect that of the classical diagonal loaded beamformer (about 3.5dB in our simulations with the SA).
Chapter 8

Summary and Future Research

8.1 Completed Research

The completed research of the thesis involves five main research topics. The first is the beamforming solution for a CRA element space narrowband partial adaptive beamformer that uses the available prior knowledge about the characteristics of some interferences to design the beamformer weights by the means of a penalty factor. The second topic optimizes the penalty factor by making it automatically adaptive that minimizes automatically the output power due to the interferences and background noise. The third topic introduces a novel beamforming technique that combines the element space and the beamspace to perform partial adaptive beamforming using the prior knowledge. The fourth topic investigates the partial adaptation using SA. We studied several sensor arrangements and suggested two new sensor arrangements for a SA. A narrowband element space partial adaptive beamformer that uses the spherical rings is also proposed as an extension of the research done with the CRA. Also, we propose a broadband element space partial adaptive beamformer that increases the frequency range by using array nesting. The last topic introduces two new algorithms for a robust element space beamformer that uses the statistical information about the sensor position errors to obtain the beamformer weights.
8.1.1 Using the Prior Knowledge in the Partial Adaptive Beam-former

We have proposed a novel partial adaptive beamformer based on the CRA for the beam-forming of a narrow-band signal where the prior knowledge about some of the interferences is available. The partial adaptive beamformer divides the CRA into sub-arrays composed of the individual rings. Hence, the beamformer weight vector is partitioned into two components, the intra-ring weights and the inter-ring weights. The intra-ring weights are designed using a fixed penalty factor that controls the amount of the interferences whose characteristics are known. The penalty factor limits the deviation of the weights from the delay-and-sum solution thus avoiding the possibility for the beampattern to suffer of high sidelobe gain at the unknown interferences DOAs. The inter-ring weights are adaptively found to remove the unknown interferences using a GSC. The performance of the proposed partial adaptive beamformer is better in most cases than the previously partial adaptive beamformer that does not use the prior knowledge in the design of the weights. In the rest of the cases, there is a need to find an optimum value of the penalty factor that should be automatically obtained to improve the $SINR_0$ performance.

8.1.2 Optimization of the Penalty Factor

We have introduced a method to improve the performance of the partial adaptive beamformer that uses the prior information about some of the interference characteristics through the optimization of the penalty factor, with the objective of achieve the maximum attainable $SINR_0$. In the proposed method, the penalty factor that controls the amount of prior knowledge applied to the design of the weights, is automatically chosen. The result is an array with reduced sidelobe levels and higher output $SINR$ with respect to the previously proposed beamformer that used a fixed value penalty factor.
8.1.3 Combined Beamspace Element Space Beamformer

The third research topic of this thesis introduces the CBSES partial adaptive beamformer, which is based on a CRA for the processing of narrow-band signals. The method to design the weights takes advantage of the prior knowledge about the characteristics of some interferences. The CBSES beamformer uses both the element space and the beamspace beamforming methods to efficiently eliminate the interferences present in the acoustic field. The beamspace targets the interferences whose characteristics were acquired by prior knowledge, and the element space is used to cancel the interferences with unknown characteristics. The resulting beamformer presents a consistent behavior in maintaining a low output interference and noise power levels, and at the same time is robust with respect to uncertainties in the interferences acquired with prior knowledge.

8.1.4 Partial Adaptive Beamforming With Spherical Array

We introduced a narrowband element space partial adaptive beamformer that uses the SA rings to perform a fixed beamforming and combine the rings outputs to perform adaptive beamforming. The proposed beamformer achieves faster convergence rate than that of the fully adaptive beamformer. Also, it dramatically reduces the computational complexity. The reduction in $SINR_0$ is small compared to that of the fully adaptive beamformer. We studied the narrowband beamformer with different sensor arrangements and we suggested two sensor arrangements for SA that maximize the distance between adjacent elements without suffering spatial aliasing, and at the same time maintain parallel rings of sensor elements.

We also introduced a frequency domain broadband element space partial adaptive beamformer that performs DFT to the broadband signal and then apply the proposed narrowband beamformer to every frequency component. The proposed broadband beamformer features array nesting, where the array rings are reused to form different sub-
arrays tailored for different frequency sub-bands. The result is a reduced number of elements for a increased range of array bandwidth. Moreover, the array directivity is kept inside the 2dB margin, contrary to the classical element space broadband beamformer whose directivity spreads more than 13dB and number of sensor elements needed to avoid aliasing at high frequencies is dramatically increased.

8.1.5 Robust Beamforming Against Sensor Position Errors

We proposed two robust beamforming methods that makes efficient use of the statistical knowledge about the sensor position errors. Contrary to the classical robust diagonal loading beamformer that uses a distortionless constraint disregarding the location errors, the proposed beamformers use a better distortionless constraint that utilizes the statistical information about the sensor position errors. The first beamforming method is applicable for sensor position errors that are time invariant but with Gaussian probability distribution among sensors and ensembles. This method does not need diagonal loading. The second beamforming method is used for sensor location errors that change with time, for example an open SA subject to vibrations. The proposed robust beamformer increases the diagonal loading robustness by using the novel distortionless constraint that includes the sensor position errors. The proposed robust beamforming methods achieve a reasonable improvement in performance with respect the classical robust methods.

8.2 Published Papers on Completed Research


L. M. Vicente and K. C. Ho, "Combined beamspace and element space technique for partial adaptive concentric ring array," in Proc. EURASIP, Poznan, Poland, Sept.
Future research goals include the following aspects:

1. Frequency invariant beamforming using element space SA.

2. Robust beamformer design against DOA uncertainty of the SOI.

3. Including the mutual coupling between sensors.

Frequency invariant beamforming usually is performed with phase mode transformations. Only a few articles investigated this topic with element space beamforming for CRA. It would be interesting to expand this research to achieve frequency invariant characteristics with the element space SA.

This thesis researched the robust beamforming against sensor position errors. The SOI DOA is assumed to be known. In practice the DOA is estimated and subject to errors. The classical robust beamformer uses the diagonal loading method. It would be very interesting to extend the current research to include the uncertainties in the SOI DOA.

Mutual coupling between the array sensors occur because the proximity of the sensing elements. This thesis assumed there are no mutual coupling. It would be very interesting
include the mutual coupling effect and investigate the robustness of the array against this effect.
Appendix A

The Discrete Fourier Transform (DFT)

The DFT is a mathematical linear transformation defined as

\[ X_k = \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \quad k = 0, \ldots, N - 1, \]  

(A.1)

where a set of discrete \( N \) complex time samples \( x_n \) with \( n = 0, \ldots, N - 1 \) are transformed on a set of \( N \) discrete samples \( X_k \) called the discrete frequency domain representation.

The DFT is an invertible transformation, the inverse DFT (IDFT) expression is

\[ x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j \frac{2\pi kn}{N}} \quad n = 0, \ldots, N - 1, \]  

(A.2)

The transform is based on the vectors composed by the elements \( e^{j \frac{2\pi kn}{N}}, \quad k = 0, \ldots, N - 1 \), that form an orthogonal basis over the set of \( N \)-dimensional complex vectors.

The DFT is used to represent a discrete set of frequency contents of a sampled signal \( x_n = x(T_s n) \) where \( T_s = \frac{1}{f_s} \) being \( f_s \) the sampling frequency. The discretized frequency contents are represented by

\[ X_k = X(f_k) \quad k = 0, \ldots, N - 1, \]  

(A.3)

where \( f_k = k \frac{f_s}{N} \) is the frequency resolution and \( X(f) \) is the frequency spectral content of the continuous time signal \( x(t) \). From its definition it can be proved that \( X_k \) is periodic.
Therefore the DFT is not able to represent $X(f)$ for all frequencies and only for those frequencies ($-\frac{f_s}{2} \leq f < \frac{f_s}{2}$). This fact agrees with the time sampling Nyquist criterion that limits the maximum frequency of a time signal $x(t)$ to be $f \leq \frac{f_s}{2}$ so it can be reconstructed from its samples $x_n = x(T_s n)$.

The discrete frequency values $f_k$ are referred to as frequency bins, and (A.3) represents the spectral value of $x_n$ in that frequency bin.
Appendix B

Delay-and-sum Weight Vector with Lagrange Multipliers

The objective is to minimize the beamformer output power from a field composed of the SOI plus spatially uncorrelated background noise as

\[ z_{s,n}(t) = v^H u(t) + v^H n(t). \]  

(B.1)

The minimization is written as

\[ v = \arg \min \left\{ E \left[ |z_{s,n}(t)|^2 \right] \right\}, \]  

(B.2)

subject to the distortionless constraint \( v^H s_0 = 1 \). This minimization problem is solved by the use of Lagrange multipliers. Introducing the Lagrange multiplier \( \eta \) in the auxiliary function

\[ J = E \left[ |z_{s,n}(t)|^2 \right] + \eta \left( 1 - v^H s_0 \right), \]  

(B.3)

where \( E \left[ |z_{s,n}(t)|^2 \right] = \sigma_s^2 v^H s_0 s_0^H v + \sigma_n^2 v^H v \) and taking the derivative of \( J \) with respect to \( v^H \) yields

\[ \frac{\partial J}{\partial v^*} = \sigma_s^2 s_0 s_0^H v + \sigma_n^2 v - \eta s_0. \]  

(B.4)

Setting the gradient to zero and solving for \( v \) gives:

\[ v = \eta \left( \sigma_s^2 s_0 s_0^H + \sigma_n^2 I \right)^{-1} s_0. \]  

(B.5)
We need to find \( \eta \). Since \( 1 = s_0^H v \), pre-multiplying (B.5) by \( s_0^H \) creates

\[
1 = \eta s_0^H \left( \sigma_s^2 s_0 s_0^H + \sigma_n^2 I \right)^{-1} s_0. \tag{B.6}
\]

Rearranging we have

\[
\eta = \frac{1}{s_0^H \left( \sigma_s^2 s_0 s_0^H + \sigma_n^2 I \right)^{-1} s_0}. \tag{B.7}
\]

Plugging in (B.7) in (B.5) we obtain

\[
v = \frac{\left( \sigma_s^2 s_0 s_0^H + \sigma_n^2 I \right)^{-1} s_0}{s_0^H \left( \sigma_s^2 s_0 s_0^H + \sigma_n^2 I \right)^{-1} s_0}. \tag{B.8}
\]

Using the Matrix inversion lemma to the term \( \left( \sigma_s^2 s_0 s_0^H + \sigma_n^2 I \right)^{-1} \) we have

\[
\left( \sigma_n^2 I + \sigma_s^2 s_0 s_0^H \right)^{-1} = \sigma_n^{-2} I - \frac{\sigma_n^{-4} s_0 s_0^H}{\sigma_n^{-2} I + \sigma_s^{-2} s_0 s_0^H}. \tag{B.9}
\]

Replacing (B.9) in (B.8), using the equivalence \( s_0^H s_0 = K \), after some mathematical manipulations, we obtain

\[
v = \frac{s_0}{s_0^H s_0} = \frac{s_0}{K}. \tag{B.10}
\]

Equation (B.10) is nothing but the delay-and-sum weight vector.

The same result will be obtained if we used \( z_n(t) = v^H n(t) \) in (B.1) instead \( z_{s,n}(t) \).
Appendix C

Optimum Weight Vector with Lagrange Multipliers

The objective is to minimize the beamformer output power from a field composed of the SOI, interferences and background noise as

\[ z(t) = s(t)v^Hs_0 + \sum_{l=1}^{L} v^Hi_l(t)i_l + v^Hn(t). \]  

(C.1)

The minimization is written as

\[ v = \arg \min \{ E[|z(t)|^2] \}, \]  

(C.2)

subject to the distortionless constraint \( v^Hs_0 = 1 \). This minimization problem is solved by the use of Lagrange multipliers. Introducing the Lagrange multiplier \( \eta \) in the auxiliary function

\[ J = E[|z(t)|^2] + \eta (1 - v^Hs_0), \]  

(C.3)

where \( E[|z(t)|^2] = v^HR_uv \) and \( R_u = R_s + R_i + R_n \). Taking the derivative of \( J \) with respect to \( v^* \) yields

\[ \frac{\partial J}{\partial v^*} = R_u v - \eta s_0. \]  

(C.4)

Setting the gradient to zero and solving for \( v \) gives:

\[ v = \eta R_u^{-1}s_0. \]  

(C.5)
We need to find $\eta$. Since $1 = {\mathbf{s}_0}^H{\mathbf{v}}$, pre-multiplying (C.5) by $\mathbf{s}_0^H$ creates

$$1 = \eta{\mathbf{s}_0}^H{\mathbf{R}_u}^{-1}\mathbf{s}_0.$$  

(C.6)

Rearranging we have

$$\eta = \frac{1}{{\mathbf{s}_0}^H{\mathbf{R}_u}^{-1}\mathbf{s}_0}.$$  

(C.7)

Plugging in (C.7) in (C.5) we obtain

$$\mathbf{v} = \frac{{\mathbf{R}_u}^{-1}\mathbf{s}_0}{{\mathbf{s}_0}^H{\mathbf{R}_u}^{-1}\mathbf{s}_0}.$$  

(C.8)

If we used $z_{in}(t) = \sum_{l=1}^{L} \mathbf{v}^H\mathbf{i}_l(t)\mathbf{i}_l + \mathbf{v}^H\mathbf{n}(t)$ in (C.1) we would obtained

$$\mathbf{v} = \frac{{\mathbf{R}_{in}}^{-1}\mathbf{s}_0}{{\mathbf{s}_0}^H{\mathbf{R}_{in}}^{-1}\mathbf{s}_0},$$  

(C.9)

where $\mathbf{R}_{in} = \mathbf{R}_i + \mathbf{R}_n$ is the correlation matrix due to the interferences and noise only. In fact, it can be proved using the matrix inversion lemma that (C.8) and (C.9) are equivalent.
Appendix D

Eigenvalue Analysis Example

D.0.1 Eigenvalue Spread

Table D.1: Eigenvalue spread

<table>
<thead>
<tr>
<th>Method</th>
<th>SA6</th>
<th>SA6b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully adaptive</td>
<td>$6.2529 \times 10^6$</td>
<td>$6.2521 \times 10^6$</td>
</tr>
<tr>
<td>Partial adaptive</td>
<td>$3.1576 \times 10^5$</td>
<td>$3.2040 \times 10^5$</td>
</tr>
</tbody>
</table>

This subsection investigates the correlation matrix eigenvalues to verify the slow behavior of the NLMS algorithm in the previous simulations. The correlation matrix is $R_{y_a}$, e.g. the lower branch correlation matrix. The eigenvalue spread is shown in Table D.1. For both scenarios SA6 and SA6b defined in Table 6.6, the eigenvalue spread of fully adaptive beamformers are similar. The same happens for partial adaptive beamformers. However, the eigenvalue spread of the partial adaptive beamformer is half of that in the fully adaptive beamformer. This justify why the former is faster. To have more information about the initial and final adaptation performance we would need to find the time constants.

D.0.2 Initial and Steady-state Time Constants

In order to assess the initial and steady-state convergence rate we need to find the maximum and minimum eigenvalues. The maximum and minimum eigenvalues are shown in
Table D.2: Eigenvalues (max, min)

<table>
<thead>
<tr>
<th>Method</th>
<th>SA6</th>
<th>SA6b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>min</td>
</tr>
<tr>
<td>Fully adaptive</td>
<td>6.2529 × 10^5</td>
<td>1.0000</td>
</tr>
<tr>
<td>Partial adaptive</td>
<td>1.5087 × 10^4</td>
<td>0.0478</td>
</tr>
</tbody>
</table>

Table D.2. We observe that both the minimum and the maximum eigenvalues for the fully adaptive beamformer are at least one order of magnitude higher than those of the partially adaptive beamformer. That will decrease both the initial and steady-state time constant obtained from Widrow and Stearns book [90] as,

\[
TC_{\text{init}} = \frac{1}{4\mu_{\text{init}}\lambda_{\text{max}}}, \quad TC_{\text{final}} = \frac{1}{4\mu_{\text{final}}\lambda_{\text{min}}}. \tag{D.1}
\]

However, the time constants also depend on the step size. In the problem at hand, the step size is different from fully adaptive to partial adaptive, and also the step size is larger at the beginning of the adaptation than at the end. The time constants in terms of algorithm iterations are shown in Table D.3. We observe that the partial adaptive beamformer have smaller initial time constants than those of the fully adaptive beamformer. Also, the final time constants are smaller in the partial adaptive beamformer. These results are in agreement with the fact that the partial adaptive beamformer has a faster initial convergence rate and achieves the steady state also faster. There is a behavior still to solve, why the partial adaptive beamformer have a larger EMSE.

Table D.3: Time constants (init, final)

<table>
<thead>
<tr>
<th>Method</th>
<th>SA6</th>
<th>SA6b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC_{init}</td>
<td>TC_{final}</td>
</tr>
<tr>
<td>Fully adaptive</td>
<td>5.9864</td>
<td>5.9891 × 10^7</td>
</tr>
<tr>
<td>Partial adaptive</td>
<td>0.4980</td>
<td>2.5162 × 10^6</td>
</tr>
</tbody>
</table>
D.0.3 Misadjustment

To verify why the EMSE of the partial adaptive beamformer is larger than that of the fully adaptive beamformer, we need to use the misadjustment expression found as [90],

$$Misadj = \mu \times tr(R_{y_a}),$$  \hspace{1cm} (D.2)

where $R_{y_a}$ is the low branch correlation matrix.

<table>
<thead>
<tr>
<th>Method</th>
<th>SA6</th>
<th>SA6b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully adaptive</td>
<td>0.0069</td>
<td>0.0069</td>
</tr>
<tr>
<td>Partial adaptive</td>
<td>0.0417</td>
<td>0.0417</td>
</tr>
</tbody>
</table>

Table D.4: Misadjustment

Table D.4 shows the misadjustment. We notice that is larger in the case of the partial adaptive beamformer. Therefore even when this beamformer has a faster convergence, at the steady-state will suffer of larger EMSE. This is an expected behavior of the NLMS algorithm [90].
Appendix E

Function of Gaussian Random Variables

Let’s consider i.i.d. random variables

\[ \Delta x_k(t) : N(0, \sigma_p^2), \quad k = 1, ..., K, \quad t = 1, ..., T. \]
\[ \Delta y_k(t) : N(0, \sigma_p^2), \quad k = 1, ..., K, \quad t = 1, ..., T. \]
\[ \Delta z_k(t) : N(0, \sigma_p^2), \quad k = 1, ..., K, \quad t = 1, ..., T. \]  
(E.1)

that represent the Cartesian coordinates misplacement in space and time of the sensor position with variance \( \sigma_p^2 \). The sensor position Cartesian coordinates are

\[ \tilde{x}_k = x_k + \Delta x_k, \quad k = 1, ..., K. \]
\[ \tilde{y}_k = y_k + \Delta y_k, \quad k = 1, ..., K. \]
\[ \tilde{z}_k = z_k + \Delta z_k, \quad k = 1, ..., K, \]  
(E.2)

where we eliminated the variable \( t \) for clarity, but is implicitly assumed. The sensor position vector is written as

\[ \tilde{p}_k = p_k + \Delta p_k, \quad k = 1, ..., K. \]  
(E.3)

The objective is to find the statistical distribution of the function of random variables

\[ X = -\kappa a_0^T \Delta p_k, \quad k = 1, ..., K, \]  
(E.4)

where \( \kappa \) is a constant number, \( a_0 \) is the DOA or direction cosines with expression

\[ a_0 = \begin{bmatrix} -\sin \theta_0 \cos \phi_0 & -\sin \theta_0 \sin \phi_0 & -\cos \theta_0 \end{bmatrix}^T, \]  
(E.5)
\[ \Delta p_k = \begin{bmatrix} \Delta x_k & \Delta y_k & \Delta z_k \end{bmatrix}^T. \quad (E.6) \]

Operating with (E.6) and (E.5) in (E.4), we obtain

\[ X = \kappa (\Delta x_k \sin \theta_0 \cos \phi_0 + \Delta y_k \sin \theta_0 \sin \phi_0 + \Delta z_k \cos \theta_0). \quad (E.7) \]

\(X\) in (E.7) is a function of three Gaussian random variables. Using the properties of function of random variables we obtain that \(X\) has a Gaussian distribution as

\[ X : N \left( 0, \kappa^2 \left( \sigma_p^2 \left( \sin \theta_0 \cos \phi_0 \right)^2 + \sigma_p^2 \left( \sin \theta_0 \sin \phi_0 \right)^2 + \sigma_p^2 \cos^2 \theta_0 \right) \right). \quad (E.8) \]

Manipulating (E.8) and applying the equivalence \(\sin^2 \theta_0 + \cos^2 \theta_0 = 1, \sin^2 \phi_0 + \cos^2 \phi_0 = 1\) we obtain

\[ X = -\kappa a_0^T \Delta p_k : N(0, \kappa^2 \sigma_p^2), \quad k = 1, \ldots, K. \quad (E.9) \]
Bibliography


192


203


Luis M Vicente was born November 26, 1964, in Salamanca, Spain. He received the Ingeniero de Telecomunicación and M.S.E.E degrees from the Universidad Politécnica de Madrid, Spain and from the Florida International University, USA in 1990 and 1996 respectively. From February 1990 to September 1993, he was with the Aerospace Division, SENER Group, Spain, as an Electrical Engineer. From September 1996 to May 1997, he was with Voyetra Inc., New York, as a Multimedia Software Engineer. From June 1997 to April 1998 he was with SIEMENS Corp., Madrid, as a Software Engineer. From May 1998 to until August 2002, he returned to Voyetra Inc. Since February 2003, he has been an Assistant Professor at the Universidad Politécnica de Puerto Rico. His research interest includes beamforming, array processing, digital signal processing, and adaptive filters. He is currently seeking his PhD. degree at the University of Missouri-Columbia where he will receive his Ph.D. in May 2009.