Acoustic Vector-Sensor Array Processing

by

Jonathan Paul Kitchens

B.S., Duke University (2004)
S.M., Massachusetts Institute of Technology (2008)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

Author .....
Department of Electrical Engineering and Computer Science
April 26, 2010

Certified by .....
Arthur B. Baggeroer
Ford Professor of Engineering; Secretary of the Navy/
Chief of Naval Operations Chair for Ocean Science
Thesis Supervisor

Accepted by .....
Professor Terry P. Orlando
Chairman, Department Committee on Graduate Students

Acoustic Vector-Sensor Array Processing

by

Jonathan Paul Kitchens

Submitted to the Department of Electrical Engineering and Computer Science
on April 26, 2010, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Existing theory yields useful performance criteria and processing techniques for acoustic pressure-sensor arrays. Acoustic vector-sensor arrays, which measure particle velocity and pressure, offer significant potential but require fundamental changes to algorithms and performance assessment.

This thesis develops new analysis and processing techniques for acoustic vector-sensor arrays. First, the thesis establishes performance metrics suitable for vector-sensor processing. Two novel performance bounds define optimality and explore the limits of vector-sensor capabilities. Second, the thesis designs non-adaptive array weights that perform well when interference is weak. Obtained using convex optimization, these weights substantially improve conventional processing and remain robust to modeling errors. Third, the thesis develops subspace techniques that enable near-optimal adaptive processing. Subspace processing reduces the problem dimension, improving convergence or shortening training time.

Thesis Supervisor: Arthur B. Baggeroer
Title: Ford Professor of Engineering; Secretary of the Navy/
Chief of Naval Operations Chair for Ocean Science

Acknowledgments

Completing a doctoral program is a monumental task for which I have many to thank. I am grateful to God for providing the opportunity, guiding me, and giving me the strength to finish. He sustained me through three floods and countless other trials.

My family deserves special thanks. I am deeply indebted to my wife, Amy, for her sacrifice during the last four years. Her support and encouragement never wavered; this effort was as much hers as it is mine. She is the love of my life and my best friend. I appreciate the patience of my son, Jack. His laughter and joy made the work easier and always put things in perspective.

Thanks also goes to my parents. They always supported my interest in science, whether they shared it or not. My father taught me to give everything my best. I'm grateful for his wisdom, his friendship, and his example. Before she passed away in 1998, my mother encouraged me to pursue my dream of graduate study at MIT. This thesis is dedicated to her.

The thoughts and prayers of my relatives have lifted my spirits. They are more than a thousand miles away, but I know they are behind me. I am blessed to have them in my life.

I am pleased to thank my advisor, Professor Arthur Baggeroer, for his patience and instruction. He challenged me to make everything better. His time and attention have improved my research, understanding, and writing. I have learned a great deal from his guidance.

I am also thankful for the mentorship of the committee. Meeting with such a distinguished group is a rare privilege. Professor Vincent Chan provided a much needed reality check and encouraged me to think about the context of my research. His advice will help me throughout my career.

Dr. Stephen Kogon deserves extra recognition for the many ways he helped me along the way. First, he helped me learn signal processing at Lincoln Laboratory. Second, he gave me guidance as my Lincoln Scholar's Program mentor. Third, he sacrificed his time to serve on my doctoral committee.

I would also like to thank Dr. Jennifer Watson, my friend and colleague at Lincoln Laboratory. Her faith in me made this experience possible. I cannot thank her enough for standing by me and giving me advice. I will always be grateful for her support.

I appreciate the help of everyone at Lincoln Laboratory who contributed to this work. The sponsorship of the Lincoln Scholar's Program made this possible and provided an excellent resource. Assistance from the group and division never faltered

thanks to William Payne and Dr. James Ward. My discussions with Drs. Shawn Kraut and Jordan Rosenthal were invaluable; I thank them again for their time. I appreciate Dr. Nicholas Pulsone for introducing me to "eigenbeams." I am also thankful for my friend, Kevin Arsenault, with whom it has been a pleasure to work.

Ballard Blair deserves recognition for meeting with me almost weekly. I enjoyed our conversations about everything from adaptive processing to z-transforms. Our luncheons will be sorely missed.

Finally, I am pleased to acknowledge Prof. Alan Oppenheim and the DSP Group for their imagination and insight. The one-of-a-kind DSPG brainstorming sessions are some of the most enjoyable experiences of my time at MIT. Special thanks goes to the members: Tom Baran, Sefa Demirtas, Zahi Karam, Joseph McMichael, Shay Maymon, and Dennis Wei. My interaction with them helped form this thesis.

This work was sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, recommendations, and conclusions are those of the author and not necessarily endorsed by the United States Government.

Contents

List of Figures

Chapter 1

Introduction

Because they are often reliable, easy to analyze, and straightforward to process, pressure sensor arrays have dominated sonar for decades. Recent advances in sensor quality and miniaturization have stirred interest in more complex devices, those that measure velocity or acceleration in addition to pressure. Each of these "vector-sensors" provides several measurements, offering significant potential and fresh challenges. Examining the use of vector-sensor arrays in passive sonar reveals the promise such arrays offer to the field of undersea surveillance.[1]

1.1 Passive Sonar Background

The principles that have historically driven passive sonar research are the same ones behind this work. Therefore, understanding the motivation for this research requires some background in passive sonar. This section provides a brief introduction to passive sonar and pressure-sensor arrays.

Passive sonar, which quietly listens for emitted sound, is effective at undersea surveillance for three reasons. First, sonar operates over great distances. Sound waves sometimes travel thousands of miles underwater. Electromagnetic waves, by contrast, generally travel short distances in saltwater before being absorbed. Second, emitted sound is exploitable. Machinery produces characteristic sounds which aid in

[1]This background material on sonar and vector-sensors is also covered in [1].

the detection, localization, and classification of vessels. Third, passive sonar is covert. Passive sonar systems are difficult to detect because they emit no energy. Active sonar, by contrast, emits energy which adversaries could use for counter-detection and avoidance.

The most common sensor employed for sonar is the hydrophone. A hydrophone measures only pressure, essentially forming an underwater microphone. Sound waves passing over a hydrophone introduce changes in pressure that are measured and used for detection. Omnidirectional hydrophones are common because their construction, maintenance, and analysis is well-understood. Decades of experience with hydrophones show they survive well in the corrosive ocean environment and are effective when assembled into arrays.

The most common array configuration is the uniformly spaced linear array. Linear arrays are often fixed to the side of a ship, mounted on the sea floor, or towed behind a moving vessel. When a vessel travels in a straight line, drag pulls a towed array into an approximately linear shape. The exact location and orientation of each sensor is usually unknown or subject to modeling errors.

1.2 Acoustic Vector-Sensors

Increasing the information measured by a sensor generally improves its performance. With acoustic measurements, particle velocity provides additional information about the direction of sound arrival. Acoustic vector-sensors each contain one omnidirectional hydrophone measuring pressure and three orthogonal geophones measuring the components of particle velocity.[2] Figure 1.2.1 illustrates a three-dimensional vector-sensor. The geophone in the figure contains a suspended coil which slides along the axis of a fixed magnet. Sound passing along the axis of the geophone vibrates the coil and induces a current. The induced current provides a measurement of velocity component along the geophone axis.

[2]Although velocity sensors are common, many vector-sensors equivalently use accelerometers, directional hydrophones, or pressure-gradient sensors.

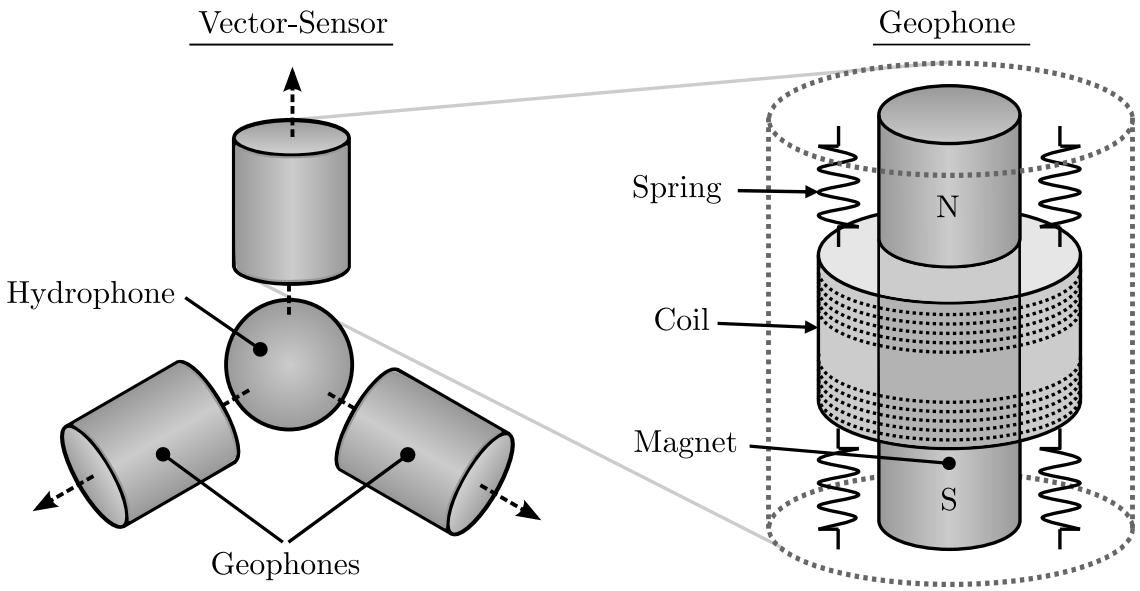

Figure 1.2.1: Notional diagram of a vector-sensor

Although geological vector-sensors have existed for decades, recent advances in geophone design have increased their utility for sonar applications. Because vector-sensors include directional information, they have the potential to improve the performance of passive sonar systems.

1.3 Vector-Sensor Processing

Vector-sensor measurements provide more information than pressure-sensor measurements. Using this additional information to improve performance is the role of vector-sensor processing. This subsection illustrates the primary benefit of vector-sensor processing: resolving ambiguous pressure-sensor measurements. Similar, more detailed analyses are provided in [1, 2, 3, 4].

The benefit of vector-sensors is first apparent when comparing a three-dimensional vector-sensor to an omnidirectional pressure-sensor. By definition, the response of the omnidirectional pressure sensor is equal in all directions. But because the vector-sensor also measures particle velocity, a three-dimensional vector, it yields information about the direction of a sound source. Put another way, *all directions are ambiguous to the pressure-sensor, but no directions are ambiguous to the vector-sensor.* This lack

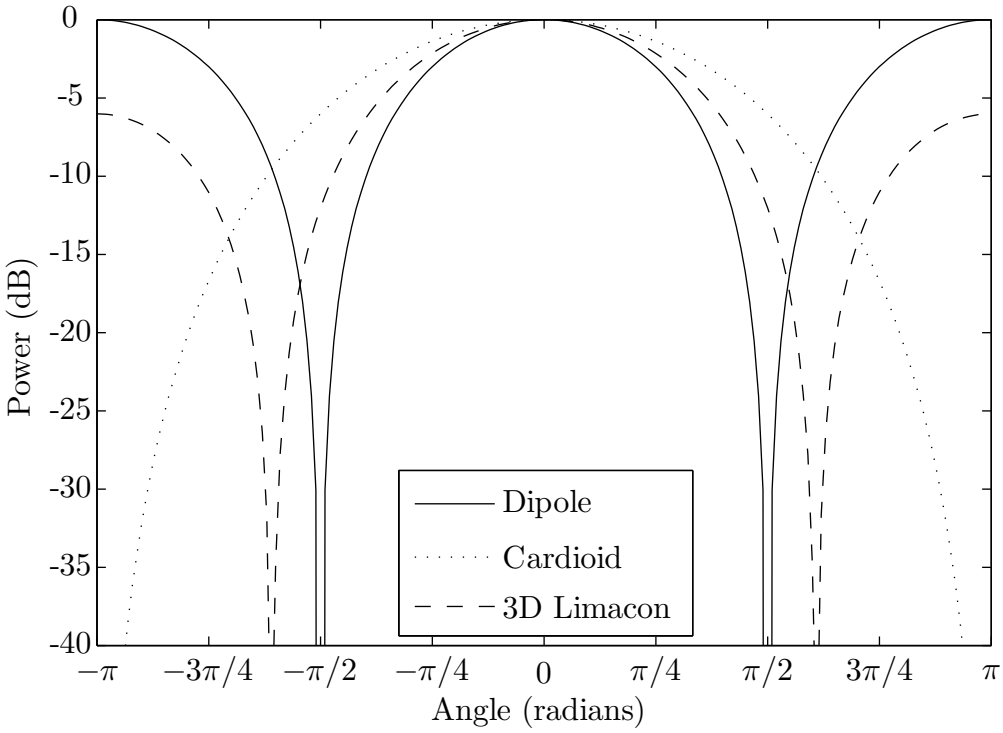

Figure 1.3.1: Example vector-sensor response patterns

of ambiguity means a single vector-sensor is inherently directional. Vector-sensors are also tunable: linear combinations of the four elements forms a "pseudo-sensor" with many different response patterns [3]. A few of these patterns are shown in Figure 1.3.1. By choosing appropriate weights, these patterns are easily rotated to emphasize or null any arbitrary direction.

The same behavior extends to arrays of vector-sensors. Compare a uniformly spaced linear array composed of N vector-sensors to one composed of N omnidirectional pressure-sensors. Example directional responses or "beampatterns" for both array types are shown in Figure 1.3.2. Both arrays have $N = 10$ elements at frequency $f = 5/7 f_d$, where f_d is the design frequency (the frequency at which the inter-element spacing is one-half wavelength). By choosing weights and linearly combining array elements, the top and bottom plots are "steered" to $\pi/2$ and $-\pi/4$, respectively. The response of a linear pressure-sensor array (PSA) is symmetric about rotation around the array axis. This is evident in the symmetric PSA beampattern: arrivals from op-

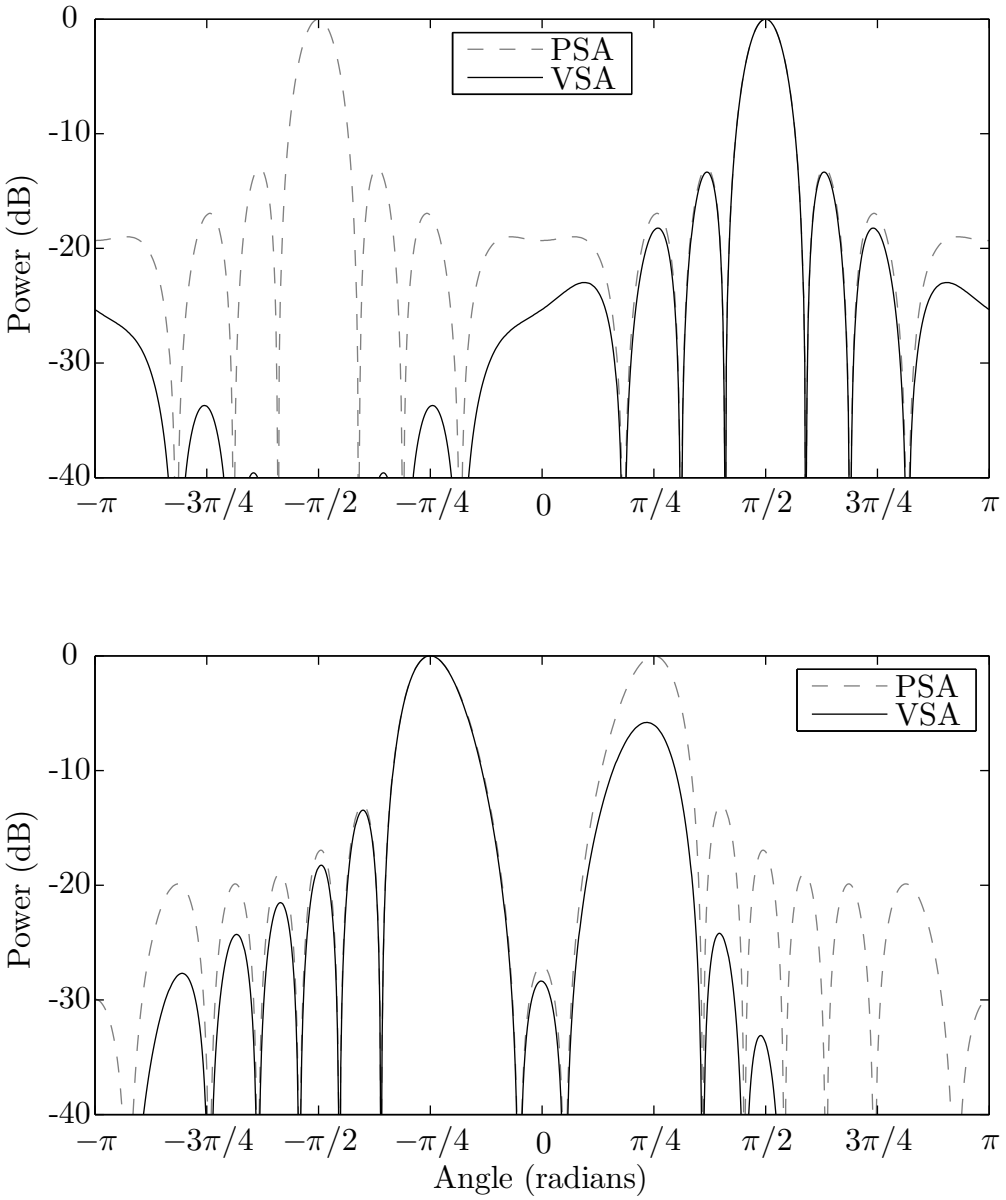

Figure 1.3.2: Example VSA and PSA response patterns

posite sides of the array yield identical pressure measurements. Changing the weights applied to each element alters the beampattern, but the response is always symmetric. The PSA beampattern always exhibits an ambiguous peak, or "backlobe," in the direction opposite the desired steering angle. In contrast, the vector-sensor array (VSA) utilizes unambiguous measurements from each sensor to reduce the backlobe. The level of backlobe reduction is determined by the choice of weights. In the top plot of Figure 1.3.2, the VSA backlobe is driven near zero; in the bottom plot of Figure 1.3.2, it is reduced by 6 dB.

Directional information makes VSA processing fundamentally different from PSA processing. Pressure-sensor processing exploits phase or time-delay measurements to resolve signals and reject noise. Vector-sensors provide little additional *phase* information because the sensor components are co-located; the directional components yield mostly *gain* information. VSA processing must exploit both gain and phase measurements to be effective.

1.4 Problem Statement

The additional measurements provided by vector-sensor arrays offer benefits and challenges. As the previous section shows, vector-sensor arrays are more versatile than arrays of only pressure-sensors. Exploiting this versatility raises a number of questions addressed in this work. These questions fall into two broad categories that serve to organize the research:

1. **How well can a vector-sensor array do?** How can the vector-sensor array "performance improvement" be quantified? By how much can vector-sensors potentially improve performance?

2. **How can a vector-sensor array do well?** How can vector-sensor arrays achieve good performance in practice? Without *a priori* information, what vector-sensor processing is best? How can vector-sensor processing adapt to incorporate new data? How can the computational cost required to process vector-sensor arrays be reduced?

Research suggests that existing results do not resolve these questions. Although vector-sensor array processing seems to be a straightforward extension to pressure-sensor array processing, it requires fundamental changes to analyses and algorithms.

The remainder of this section highlights the difficulty of answering the above questions with current approaches. Questions in the first category require metrics and bounds to quantify vector-sensor performance. Questions in the second category subdivide according to the two fields of array processing. The first field is nonadaptive processing, where the sensors are combined linearly using fixed weights. The second field is adaptive processing, where weights are allowed to change based upon observed data.

1.4.1 Performance Metrics and Limits

Two performance dimensions commonly used to quantify and bound improvements in array processing are array resolution and gain/directivity. The next two paragraphs briefly show that VSA improvements are not expressed along these performance dimensions.

Either beamwidth or angle estimation error is typically used to quantify array resolution. Figure 1.3.2 reveals that the VSA and PSA beamwidths are almost exactly the same. A wide class of beampatterns, including those in [3, 4], relies on pattern multiplication (see [5, §2.8]). The width of such beampatterns is virtually unchanged from that of a pressure-sensor array [1, §2.1]. Another metric that quantifies array resolution is the root mean squared error (RMSE) resulting from direction-of-arrival (DOA) estimation. Improving array resolution lowers the RMSE. Bounds on the RMSE are often derived and compared to the actual error of common direction-of-arrival algorithms; for vector-sensor arrays this analysis appears in [1, 2, 4, 6]. Representative plots are shown in Figure 1.4.1, derived from [1, §3.2] for the case of a single narrowband source in white noise and a $N = 13$ element, mismatched, linear VSA. The moderate mismatch scenario includes zero-mean Gaussian position, rotation, gain, and phase errors; the direction-of-arrival algorithm is a conventional beamscan technique. A detailed description of the parameters, algorithm, and bound is in [1]. Two

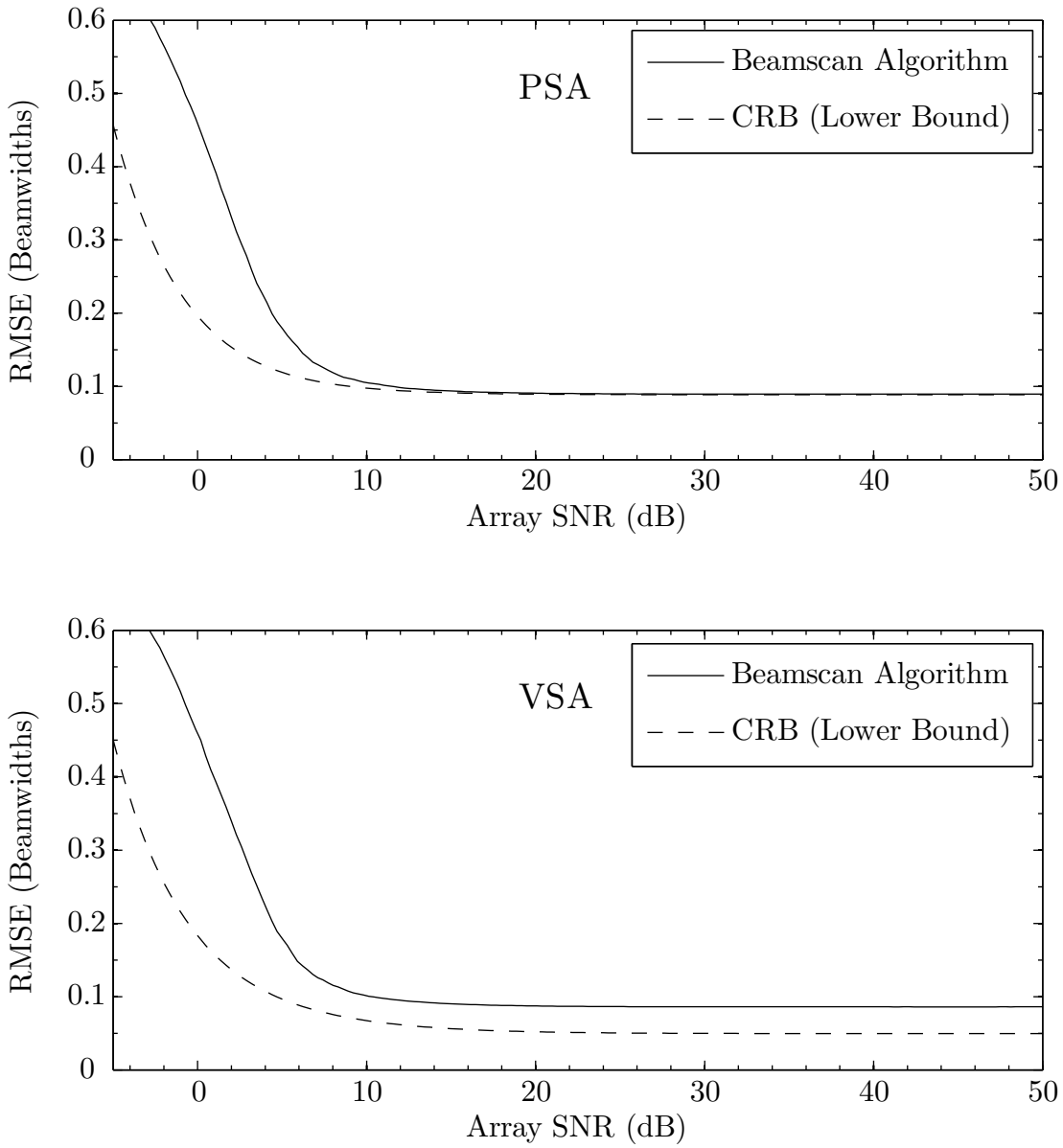

Figure 1.4.1: VSA vs. PSA resolution, measured by direction-of-arrival error

observations are clear from the figure: 1) the actual algorithm RMSE does not decrease significantly with a vector-sensor array, and 2) the lower bound indicates that only a modest improvement is possible. A key consideration is that vector-sensors increase the number of measurements from N to $4N$, but simply increasing the number of *pressure-sensors* to $4N$ (keeping the same inter-element spacing) yields a smaller beamwidth and a lower RMSE. Unlike increasing the number of pressure-sensors, however, using vector-sensors achieves improvement without altering the physical length of an array.

Vector-sensor arrays evidently do not substantially improve resolution, but research further reveals that VSAs do not improve directivity more than PSAs with comparable numbers of components. The directivity of vector-sensor arrays, presented in [7], is at most 6 dB higher than pressure-sensor arrays. As with array resolution this improvement is no better than that achieved by simply increasing the number of pressure-sensors from N to $4N$.

Because the benefits of vector-sensors are not reflected in measures such as resolution or directivity (considering the increased number of components), new measures are necessary to quantify VSA performance. Although existing bounds are useful for analyzing vector-sensor array configuration [6] and robustness [1], alternative bounds are required to understand how much improvement VSAs offer along the new performance dimensions.

1.4.2 Nonadaptive VSA Processing

Some of the most powerful nonadaptive processing techniques become difficult or impossible with vector-sensor arrays. Designing fixed weights for nonadaptive processing involves multiple objectives. Three of the most useful objectives are narrow beamwidth, low sidelobe level, and low sensitivity to modeling errors. Analytical methods enable joint optimization of the PSA beamwidth and sidelobe level, but these methods do not apply to VSAs. As a result, VSA beampatterns are often designed using alternative criteria.

Many existing approaches are similar to [3, 4] and effectively choose weights to maximize gain against some postulated noise field (see Section 4.2). One formulation of this problem is the mathematical program

$$\begin{aligned} \text{minimize} \quad & \mathbf{w}^H \widetilde{\mathbf{R}} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^H \mathbf{v}_0 = 1 \end{aligned} \tag{1.4.1}$$

for some postulated covariance matrix $\widetilde{\mathbf{R}}$ and signal replica $\mathbf{v}_0$. Choices for the postulated covariance matrix are often combinations of isotropic noise, white noise, and point sources. The resulting weights may have a simple closed form, and pattern multiplication may allow for spatial tapering. For instance, choosing

$$\widetilde{\mathbf{R}} = \mathbf{v}_b \mathbf{v}_b^H + \sigma^2 \mathbf{I}, \tag{1.4.2}$$

with $\mathbf{v}_b$ being a signal replica directed at the backlobe, gives "point null" beampatterns as shown in the top plot of Figure 1.4.2. Applying a 25 dB Taylor spatial taper ([5, §3.4.3]) to the weights yields the beampatterns shown in the bottom plot.

For a vector-sensor array, optimizing the important objectives of narrow beamwidth, low sidelobe level, and low sensitivity requires new techniques. Because existing techniques do not explicitly optimize over these objectives, the resulting weights are suboptimal with respect to each objective. For instance, the beampatterns in Figure 1.4.2 leave room for improvements in mainlobe width, sidelobe level, and robustness. Techniques that optimize these objectives are widely used for PSA processing, so equivalent techniques for VSA processing are important to develop.

1.4.3 Adaptive VSA Processing

A key problem in adaptive vector-sensor array processing is high dimensionality. Vector-sensor array data is four times the dimension of pressure-sensor array data because of the additional velocity measurements. This high dimension complicates adaptive processing in two ways.

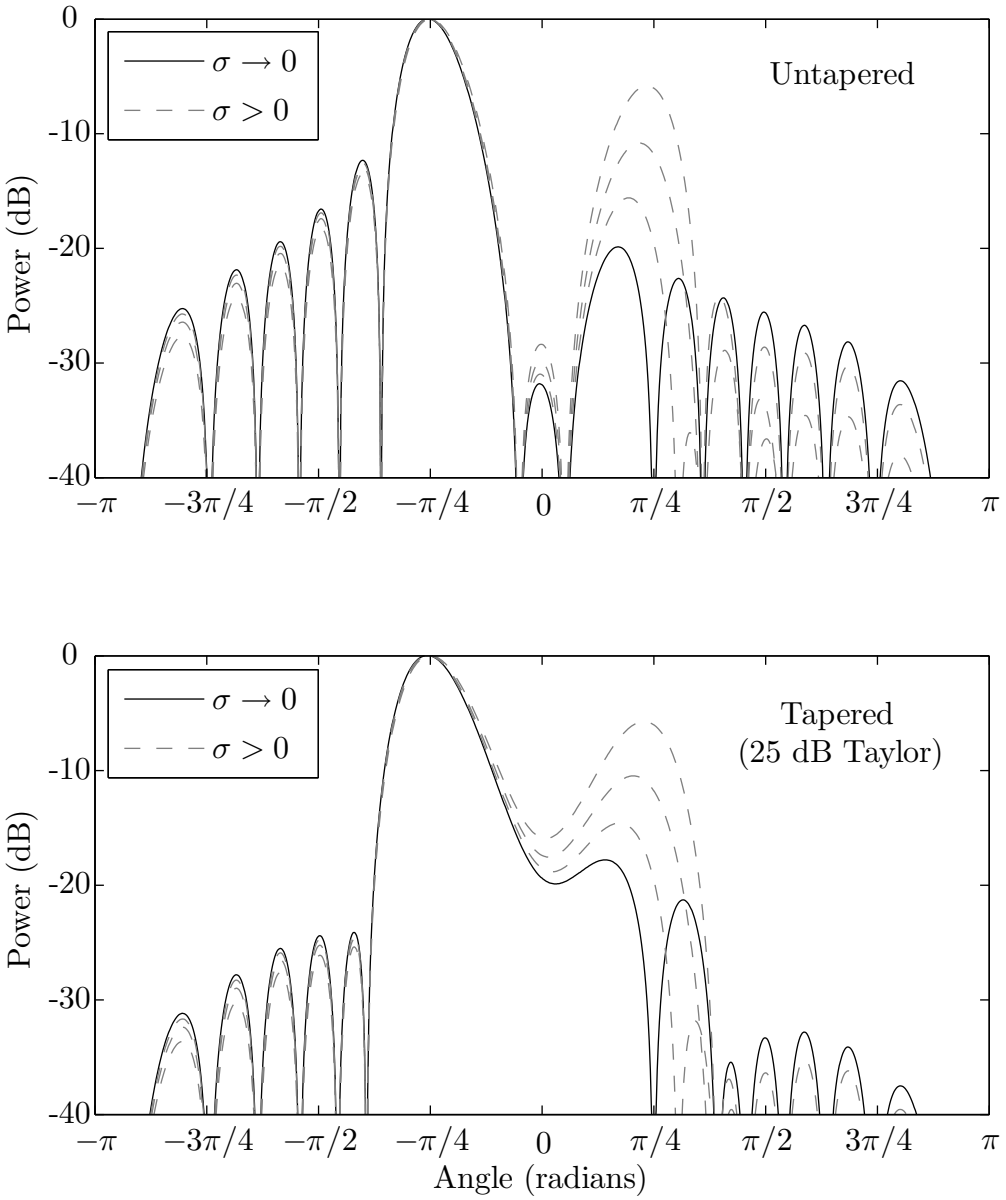

Figure 1.4.2: VSA beampatterns with "point nulls" and diagonal loading

First, it makes parameter estimation more difficult. Adaptive processing often requires estimating the second-order moments of the data, *i.e.* the covariance matrix. Logic similar to [8] quickly reveals the scope of this problem. The number of independent observations, or "snapshots," available is determined by the stationarity of the environment and the length of the array. The environment is effectively stationary if sources move less than a beamwidth during observation. The broadside beamwidths of vector and pressure-sensor arrays are almost the same, $\Delta\theta \approx 2/N$. Recall that N is the number of vector-sensors; the total number of measurements is $4N$ for a 3-D vector-sensor array. For an array of length L at the design frequency with wavelength λ, $N = 2L/\lambda$. The worst case (shortest) stationarity time is then given by a broadside source at range R moving tangentially at speed v:

$$\begin{aligned} \Delta T_{\text{stat}} &\approx \Delta\theta \cdot R/v \\ &= \lambda R/(Lv). \end{aligned} \tag{1.4.3}$$

The time required to preserve accurate phase estimates for a single snapshot is approximately $8\times$ the maximum travel time of sound waves across the array. This travel time is longest at array endfire, where it is

$$\Delta T_{\text{snap}} \approx 8L/c \tag{1.4.4}$$

for sound wave propagation speed c. The approximate number of snapshots available, K_{avail}, is then

$$\begin{aligned} K_{\text{avail}} &\approx \Delta T_{\text{stat}}/\Delta T_{\text{snap}} \\ &= \lambda cR/(8vL^2) \end{aligned} \tag{1.4.5}$$

The number of snapshots *desired* is determined by the dimension of the sample covariance matrix; a common rule-of-thumb is to use more than two or three times the dimension for a well-estimated matrix [9]. Using fewer snapshots produces weights that are not robust and are sensitive to noise. Assuming an optimistic factor of two

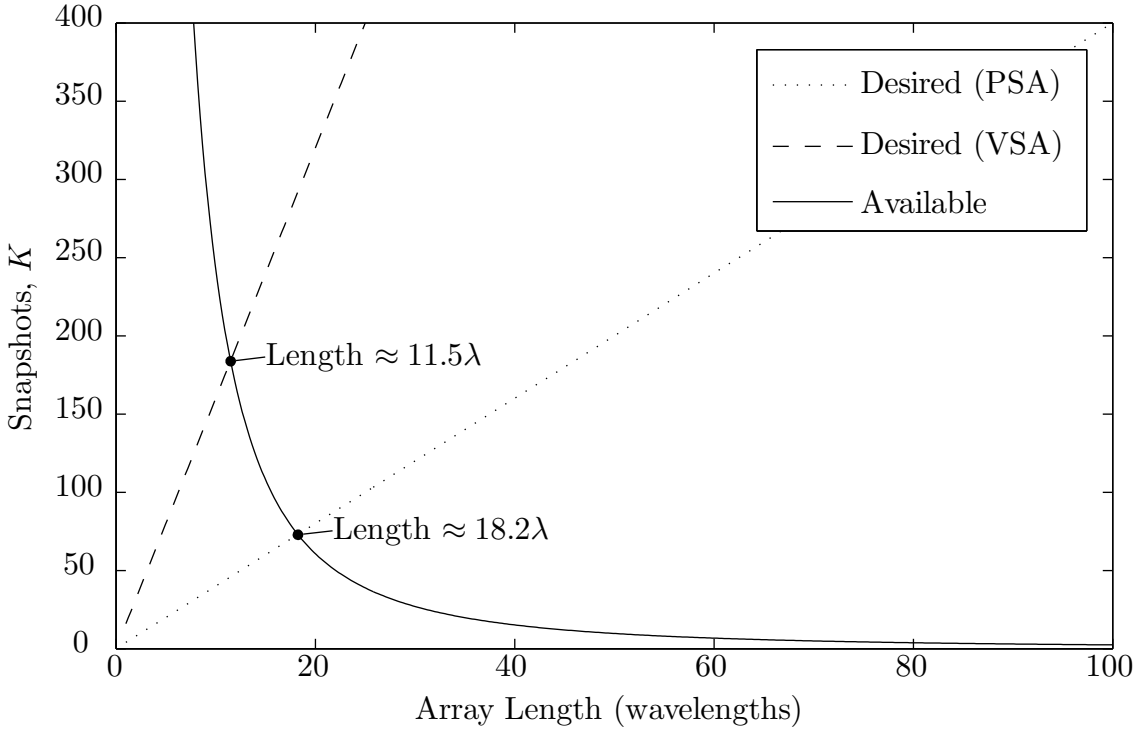

Figure 1.4.3: Snapshot deficiency: $R = 10$ km, $v = 20$ knots, $f = 200$ Hz

times the data dimension, the number of snapshots desired, K_{des}, for a VSA and PSA are

$$\begin{aligned} K_{\text{des,PSA}} &\approx 2 \cdot N \\ &= 4L/\lambda \qquad (1.4.6) \\ K_{\text{des,VSA}} &\approx 2 \cdot 4N \\ &= 16L/\lambda. \qquad (1.4.7) \end{aligned}$$

A typical scenario with $R = 10$ km, $v = 20$ knots, and $f = 200$ Hz yields the curves shown in Figure 1.4.3. These curves illustrate a fundamental adaptive processing problem: the number of available snapshots is usually far fewer than desired. The problem is worse for vector-sensor arrays because of the higher dimension. As indicated on the plot, covariance matrices are poorly estimated for vector-sensor arrays longer than about 11.5λ, or $N > 23$. The same problem exists for pressure-sensor ar-

rays, but at longer lengths (18.2λ, or $N > 36$). Adaptive VSA processing techniques must combat this snapshot deficiency to be effective in practice.

High dimensional vector-sensor data complicates adaptive processing a second way by increasing the computational requirements. A typical adaptive processing operation, the singular value decomposition of a covariance matrix, generally takes $\mathrm{O}(N^3)$ floating point operations. A vector-sensor array, then, increases the computational burden by a factor of roughly $4^3 = 64$.

Current approaches for high-dimensional adaptive array processing fall into three categories. The first category augments the covariance matrix to make it well-conditioned. Fixed and variable diagonal loading as covered in [10] and [11] take this approach. The second category performs adaptive beamforming in reduced-dimension linear subspaces. Many techniques fall in this category, including some suggested in [4]. The third category utilizes additional information to improve the covariance matrix estimate. One such technique is "Physically Constrained Maximum-Likelihood (PCML)" estimation [12]. The problem of high-dimension is more pronounced with vector-sensor arrays, so existing and new techniques must be closely examined.

1.5 Key Findings

This thesis includes several novel contributions to the field of array processing. It establishes the limits of VSA performance and describes practical techniques that approach these limits. Organized by chapter, the most significant contributions are:

Ch 2: A thorough exploration of vector-sensor array fundamentals. One key finding in this area is that many useful properties are *not* exhibited by vector-sensor arrays. Another key finding is a real expression for the VSA beampattern.

Ch 3: Two performance bounds on a critical VSA capability: resolving pressure ambiguities. These bounds relate ambiguity resolution to the common problems of detection and estimation. Key findings include showing that

1) the bounds, although fundamentally different, both depend on the same simple quantity, and 2) good performance is theoretically possible in most cases.

Ch 4: The design of robust, fixed weights with excellent performance characteristics. Key findings include the "Minimum Sensitivity" criterion, an algorithm for designing robust weights, and a demonstration of improved performance.

Ch 5: The derivation of optimum subspaces that enable or improve adaptive processing. Key findings include 1) the optimality criterion of "inner product preservation," 2) the derivation of eigenbeam subspaces as least-squares designs, and 3) a demonstration of significant performance improvement.

Several of the contributions listed above are summarized at the end of the thesis in Section 6.1, Figure 6.1.1.

1.6 Sensor and Environment Model

This entire document assumes the same sensor and environment model to simplify discussion.[3] Each section explicitly notes any departures from or extensions to this common model. The subsequent analysis assumes the following sensor model:

1. *Co-located sensor components.* The hydrophone and three geophones of each vector-sensor are located at the same point and observing the same state. In practice, this requires the component spacing to be small compared with the minimum wavelength.

2. *Point sensors.* Each vector-sensor is modeled as a single point. In practice, this requires the sensor dimensions to be small compared with the minimum wavelength.

[3]The same sensor, environment, and plane wave models are also covered in [1].

3. *Geophones with cosine response.* The signal response of each geophone is proportional to the cosine of the angle between the geophone axis and the source. Cosine geophone response results from measuring velocity along only one axis.

4. *Orthogonal geophones.* The axes of the three geophones are orthogonal. This is true in practice when each vector-sensor is a static unit.

The thesis also assumes the following environment model:

1. *Free-space environment.* Sound waves travel in a quiescent, homogeneous, isotropic fluid wholespace. This implies direct-path propagation only.

2. *Narrowband signals.* The signal is analyzed at a single frequency. This means the signal is sufficiently band-limited to allow narrowband processing in the frequency domain. Passive sonar systems typically operate over a wide, many-octave bandwidth; narrowband signals may be obtained by computing the discrete Fourier transform of the measurements and processing each frequency bin.

3. *Plane wave propagation.* The sound waves are planar at each sensor and across the array. This implies the unit vector from each sensor to the source is the same, regardless of the sensor location. Sound waves are approximately planar when the source is beyond the Fresnel range [8].

The underlying assumptions and notation are similar to those in [2, 6, 13] although this document has a different objective.

1.7 Plane Wave Measurement Model

Under the assumptions in Section 1.6, consider a plane wave parameterized by azimuth $\phi \in [0,\ 2\pi)$ and elevation $\psi \in [-\pi/2,\ \pi/2]$ impinging on an array of N vector sensors. The remainder of the thesis assumes a right-handed coordinate system with $\phi = 0$ as forward endfire, $\phi = \pi/2$ as port broadside, $\psi = 0$ as zero elevation, and $\psi = \pi/2$ as upward. The parameters ϕ and ψ are sometimes grouped into the vector $\boldsymbol{\Theta}$ for notational convenience. Without loss of generality, assume the geophone axes

are parallel to the axes of the coordinate system. If this is not the case, the data from each vector sensor can be rotated to match the coordinate axes. The unit vector,

$$\mathbf{u} = \left[\cos\phi\cos\psi,\ \ \sin\phi\cos\psi,\ \ \sin\psi\right]^T, \tag{1.7.1}$$

points from the origin to the source (or, opposite the direction of the wave propagation). The following derivations touch only briefly on direct-path acoustic propagation. For a much more detailed study of ocean acoustics, see [14].

Under common conditions, the components of velocity relate linearly to pressure. Assuming an inviscid homogeneous fluid, the Navier-Stokes equations become the Euler equations

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}^T \nabla \mathbf{v} = -\frac{\nabla p}{\rho} \tag{1.7.2}$$

where $\mathbf{v}$ is fluid velocity, ρ is density, and p is pressure. For acoustic propagation this equation is linearized, neglecting the convective acceleration term $\mathbf{v}^T\nabla\mathbf{v}$. With a plane wave, the pressure p relates across time t and position $\mathbf{x}$ through the sound speed c:

$$p(\mathbf{x}, t) = f\left(\frac{\mathbf{u}^T\mathbf{x}}{c} + t\right) \tag{1.7.3}$$

$$\therefore \quad \nabla p = \frac{\mathbf{u}}{c} \cdot \frac{\partial p}{\partial t}. \tag{1.7.4}$$

Substituting Equation 1.7.4 into the Euler equations in 1.7.2 shows that under weak initial conditions the pressure and fluid velocity obey the plane wave impedance relation

$$\mathbf{v} = -\frac{\mathbf{u}}{\rho c}\ p. \tag{1.7.5}$$

Because the geophones are aligned with the coordinate axes, they simply measure the components of the velocity vector $\mathbf{v}$. The resulting linear relationship between the pressure and each component of the fluid velocity greatly simplifies the analysis of vector-sensor array performance.

The linear relationship in Equation 1.7.5 enables expressing the velocity measure-

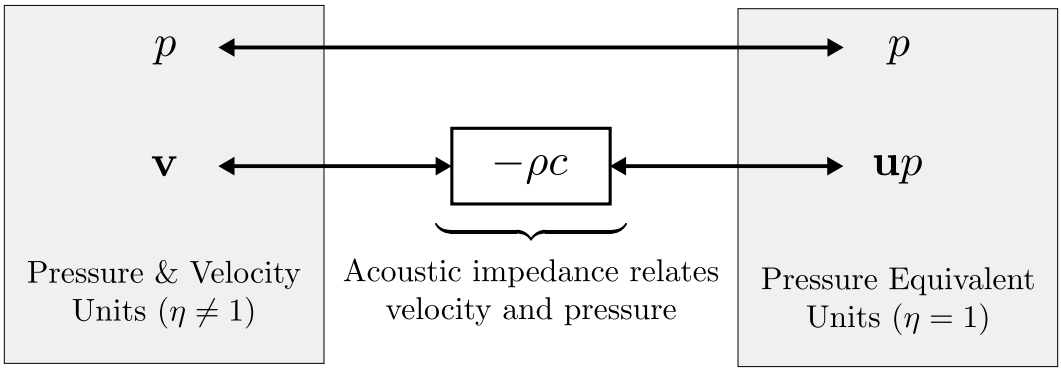

Figure 1.7.1: Vector-sensor measurements, scaled to common units

ments in terms of pressure and the source unit vector. Returning to the array of N vector-sensors, the plane wave measurement of the nth vector-sensor in phasor form is

$$e^{jk_0(\mathbf{r}_n^T\mathbf{u})}\begin{bmatrix}1\\-\mathbf{u}/\rho c\end{bmatrix} \tag{1.7.6}$$

where $\mathbf{r}_n$ is the position of the sensor and

$$k_0 \triangleq \frac{2\pi}{\lambda} \tag{1.7.7}$$

is the wavenumber for a wavelength λ. The term outside the vector is the wave phase delay, which factors out because of Equation 1.7.5. Only the gain difference between the pressure sensors and geophones is important. For convenience, this thesis chooses a normalization that absorbs that gain difference into the pressure term:

$$e^{jk_0(\mathbf{r}_n^T\mathbf{u})}\begin{bmatrix}\eta\\\mathbf{u}\end{bmatrix}. \tag{1.7.8}$$

Although this choice of normalization seems arbitrary, it results in simpler expressions later and is similar to the notation used in [2, 6, 13]. [4] Also note that this choice of normalization requires a factor of $(\rho c)^{-2}$ when comparing beam estimates in units of absolute power.

[4]The η defined here is not exactly the same as the one used in [2, 6, 13].

The remainder of this thesis uses the gain factor $\eta = 1$ in all derivations unless otherwise stated. In most cases, the results are easily extended to arbitrary η, sometimes by inspection. The choice of $\eta = 1$ for analysis has an important practical motivation involving the trade-off between left/right resolution and white noise array gain or sensitivity. Although the pressure-sensor often has higher gain ($\eta > 1$) for actual vector-sensor arrays, the array data is easily normalized to common units as shown in Figure 1.7.1. Normalizing the units produces two results: 1) a slight loss of array gain because of the increased geophone noise, and 2) an improved ability to resolve ambiguities. Vector-sensor arrays are generally chosen for their ambiguity resolution, so (2) takes precedent. Put another way, using the $\eta = 1$ data normalization strengthens ambiguity resolution at the possible expense of white noise gain.

Chapter 2

Elements of Vector-Sensor Array Processing

Any field, no matter how well organized, possesses a set of fundamentals that must be understood before moving into advanced topics. The field of acoustic vector-sensor array processing is built from elements common to all array processing and elements specific to vector-sensor arrays. The source [5] reviews the former; this chapter introduces some of the latter. Most of the concepts introduced in this chapter are new to the literature and, although simple, have profound consequences.

2.1 Vector-Sensor Array Beampattern

One of the most fundamental differences between vector-sensor arrays and pressure-sensor arrays is the structure of the beampattern. Building on the plane wave measurement model provided in Section 1.7, this section provides an expression for the beampattern of an arbitrary vector-sensor array with arbitrary element weighting. It then simplifies this expression for the case of a uniform linear vector-sensor array. The symmetry of the uniform linear array leads to the use of conjugate symmetric weights, a real beampattern, and a reflection symmetry relating beams in one quadrant to the other three quadrants.

2.1.1 General Complex Beampattern Expression

The most general case for a vector-sensor beampattern is an arbitrary array with an arbitrary, complex element weighting. Beginning with the measurement of a single vector-sensor in Equation 1.7.8, the beampattern of an N-element vector-sensor array is the weighted sum

$$y(\boldsymbol{\Theta}) = \sum_{n=1}^{N} \mathbf{w}_n^H \mathbf{v}_n(\boldsymbol{\Theta}) \tag{2.1.1}$$

where

$$\mathbf{v}_n(\boldsymbol{\Theta}) \triangleq e^{jk_0\,\mathbf{r}_n^T\mathbf{u}(\boldsymbol{\Theta})} \begin{bmatrix} 1 \\ \mathbf{u}(\boldsymbol{\Theta}) \end{bmatrix} \tag{2.1.2}$$

is the measurement of the n^{th} vector-sensor and $\mathbf{w}_n$ are the weights. Recall from Section 1.7 that $\mathbf{r}_n$ is the position of the n^{th} vector-sensor, k_0 is the wavenumber, and $\mathbf{u}(\boldsymbol{\Theta})$ is the unit vector directed toward $\boldsymbol{\Theta}$. Without knowledge of any sensor positions or constraints on the weights, Equation 2.1.1 cannot be simplified further. It is generally a complex valued expression that is difficult to analyze partly because the unit vector, $\mathbf{u}$, appears both inside and outside the complex exponential. The beampattern at any point is a linear combination of the weights, so defining the array measurement and weight vectors,

$$\mathbf{v}(\boldsymbol{\Theta}) \triangleq \begin{bmatrix} \mathbf{v}_1^T(\boldsymbol{\Theta}) & \mathbf{v}_2^T(\boldsymbol{\Theta}) & \cdots & \mathbf{v}_N^T(\boldsymbol{\Theta}) \end{bmatrix}^T \tag{2.1.3}$$

$$\mathbf{w} \triangleq \begin{bmatrix} \mathbf{w}_1^T & \mathbf{w}_2^T & \cdots & \mathbf{w}_N^T \end{bmatrix}^T, \tag{2.1.4}$$

enables writing Equation 2.1.1 as a compact inner product:

$$y(\boldsymbol{\Theta}) = \mathbf{w}^H \mathbf{v}(\boldsymbol{\Theta}). \tag{2.1.5}$$

Sampling the beampattern at a set of M points, $\{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_M\}$, corresponds to the linear transformation

$$\mathbf{y}^* = \mathbf{V}^H \mathbf{w}, \tag{2.1.6}$$

with

$$\mathbf{V} \triangleq \left[\begin{array}{cccc} \mathbf{v}(\mathbf{\Theta}_1) & \mathbf{v}(\mathbf{\Theta}_2) & \cdots & \mathbf{v}(\mathbf{\Theta}_M) \end{array} \right] \tag{2.1.7}$$

$$\mathbf{y} \triangleq \left[\begin{array}{cccc} y(\mathbf{\Theta}_1) & y(\mathbf{\Theta}_2) & \cdots & y(\mathbf{\Theta}_M) \end{array} \right]^T. \tag{2.1.8}$$

This linear transformation is valid for any arbitrary vector-sensor array; its real counterpart is derived in the next section and forms the foundation of beampattern design via convex optimization in Chapter 4.

2.1.2 Simplifications Exploiting Symmetry

For a linear vector-sensor array with elements symmetric about the origin, a series of simplifications to Equation 2.1.1 is possible. These simplifications allow 1) conjugate symmetry that reduces the number of variables from $8N$ to $3N$ and 2) quadrant symmetry that reduces the design burden by a factor of four. This thesis discusses signals in the x-y plane, but the results extend easily to the 3-D case. When dealing with signals in the horizontal plane, the vertical geophone contributes nothing and is ignored. Because the array is linear and the position and direction vectors are in the horizontal plane,

$$k_0 \, \mathbf{r}_n^T \mathbf{u}(\mathbf{\Theta}) = d_n k_0 \cos\phi \tag{2.1.9}$$

where d_n is the position of the element along the array. Ignoring the vertical geophone, the measurement vector of a single vector-sensor is

$$\mathbf{v}_n(\phi) = e^{j d_n k_0 \cos\phi} \begin{bmatrix} 1 \\ \cos\phi \\ \sin\phi \end{bmatrix}. \tag{2.1.10}$$

Writing each weight vector in terms of magnitude and phase gives

$$\mathbf{w}_n \triangleq \left[\begin{array}{ccc} a_n e^{j\alpha_n} & b_n e^{j\beta_n} & c_n e^{j\gamma_n} \end{array} \right]^T, \tag{2.1.11}$$

where a_n, α_n, *etc.* are real. Substituting Equations 2.1.10 and 2.1.11 into Equation 2.1.1 yields

$$\begin{aligned} y(\phi) &= \sum_{n=1}^{N} a_n e^{j(d_n k_0 \cos\phi - \alpha_n)} \\ &\quad + b_n \cos\phi \, e^{j(d_n k_0 \cos\phi - \beta_n)} \\ &\quad + c_n \sin\phi \, e^{j(d_n k_0 \cos\phi - \gamma_n)}. \end{aligned} \tag{2.1.12}$$

Because the element spacing is symmetric about the array center, 1) the vectors $\mathbf{v}(\Theta)$ are conjugate symmetric and 2) most problems involve only conjugate symmetric weights (see Appendix A.2). The full-length (conjugate symmetric) weight vector, $\mathbf{w}$, is fully characterized by a half-length weight vector, $\widetilde{\mathbf{w}}$. Assuming an even number of elements, the parameterization is

$$L \triangleq \frac{N}{2} \tag{2.1.13}$$

$$\widetilde{d}_l \triangleq d_{L+l} \tag{2.1.14}$$

$$\widetilde{\mathbf{w}}_l \triangleq \begin{bmatrix} \widetilde{a}_l e^{j\widetilde{\alpha}_l} & \widetilde{b}_l e^{j\widetilde{\beta}_l} & \widetilde{c}_l e^{j\widetilde{\gamma}_l} \end{bmatrix}^T \tag{2.1.15}$$

$$\mathbf{w}_n = \begin{cases} \widetilde{\mathbf{w}}_{n-L} & n > L \\ \widetilde{\mathbf{w}}^*_{L-n+1} & n \leq L \end{cases} \tag{2.1.16}$$

for real variables $\widetilde{a}_l$, $\widetilde{\alpha}_l$, *etc.*. The beampattern in Equation 2.1.12 becomes a real function when the weights are conjugate symmetric:

$$\begin{aligned} y(\phi) &= \sum_{l=1}^{L} 2\widetilde{a}_l \cos(\widetilde{d}_l k_0 \cos\phi - \widetilde{\alpha}_l) \\ &\quad + 2\widetilde{b}_l \cos(\widetilde{d}_l k_0 \cos\phi - \widetilde{\beta}_l) \cos\phi \\ &\quad + 2\widetilde{c}_l \cos(\widetilde{d}_l k_0 \cos\phi - \widetilde{\gamma}_l) \sin\phi. \end{aligned} \tag{2.1.17}$$

Note the similarity between the derivation above and the steps involved in FIR filter design; this aspect of the vector-sensor beampattern is explored more in Section 2.3.

Using trigonometric identities, Equation 2.1.17 simplifies further:

$$\begin{aligned} y(\phi) &= 2\sum_{l=1}^{L} \widetilde{a}_l \cos(\widetilde{d}_l k_0 \cos\phi)\cos\widetilde{\alpha}_l + \widetilde{a}_l \sin(\widetilde{d}_l k_0 \cos\phi)\sin\widetilde{\alpha}_l \\ &\quad + \widetilde{b}_l \cos(\widetilde{d}_l k_0 \cos\phi)\cos\widetilde{\beta}_l \cos\phi + \widetilde{b}_l \sin(\widetilde{d}_l k_0 \cos\phi)\sin\widetilde{\beta}_l \cos\phi \\ &\quad + \widetilde{c}_l \cos(\widetilde{d}_l k_0 \cos\phi)\cos\widetilde{\gamma}_l \sin\phi + \widetilde{c}_l \sin(\widetilde{d}_l k_0 \cos\phi)\sin\widetilde{\gamma}_l \sin\phi \\ &= 2\sum_{l=1}^{L} \cos(\widetilde{d}_l k_0 \cos\phi)\left[\widetilde{a}_l \cos\widetilde{\alpha}_l + \widetilde{b}_l \cos\widetilde{\beta}_l \cos\phi + \widetilde{c}_l \cos\widetilde{\gamma}_l \sin\phi\right] \\ &\quad + \sin(\widetilde{d}_l k_0 \cos\phi)\left[\widetilde{a}_l \sin\widetilde{\alpha}_l + \widetilde{b}_l \sin\widetilde{\beta}_l \cos\phi + \widetilde{c}_l \sin\widetilde{\gamma}_l \sin\phi\right] \\ &= 2\sum_{l=1}^{L} \cos(\widetilde{d}_l k_0 \cos\phi)\left[\widetilde{a}_l^{\mathcal{R}} + \widetilde{b}_l^{\mathcal{R}} \cos\phi + \widetilde{c}_l^{\mathcal{R}} \sin\phi\right] \\ &\quad + \sin(\widetilde{d}_l k_0 \cos\phi)\left[\widetilde{a}_l^{\mathcal{I}} + \widetilde{b}_l^{\mathcal{I}} \cos\phi + \widetilde{c}_l^{\mathcal{I}} \sin\phi\right]. \end{aligned} \tag{2.1.18}$$

The last step changes from a magnitude/phase parameterization to a real/imaginary parameterization, using the superscripts $^{\mathcal{R}}$ and $^{\mathcal{I}}$ indicate the real and imaginary parts of the weights. Four aspects of Equation 2.1.18 are worth noting. First, the conjugate symmetry reduces the number of (real) variables from $6N$ to $3N$. Second, for a linear vector-sensor array with elements spaced uniformly, d units apart, $\widetilde{d}_l = (l - \frac{1}{2})d$. Third, the mapping from the reduced variables in Equation 2.1.18 to the full, conjugate symmetric weight is a linear transformation. Fourth, the derivation of Equation 2.1.18 assumes even N but is easily modified for odd N.

The beampattern in Equation 2.1.18 is a simple inner product, the real counterpart to Equation 2.1.5. The single-sensor weight and measurement terms

$$\overline{\mathbf{v}}_l(\phi) \triangleq 2 \cdot \begin{bmatrix} \cos(\widetilde{d}_l k_0 \cos\phi) \\ \sin(\widetilde{d}_l k_0 \cos\phi) \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \cos\phi \\ \sin\phi \end{bmatrix} \tag{2.1.19}$$

$$\overline{\mathbf{w}}_l \triangleq \begin{bmatrix} \widetilde{a}_l^{\mathcal{R}} & \widetilde{b}_l^{\mathcal{R}} & \widetilde{c}_l^{\mathcal{R}} & \widetilde{a}_l^{\mathcal{I}} & \widetilde{b}_l^{\mathcal{I}} & \widetilde{c}_l^{\mathcal{I}} \end{bmatrix}^T \tag{2.1.20}$$

are the real counterparts to Equations 2.1.10 and 2.1.11. Concatenating these terms

yields the full array vectors

$$\overline{\mathbf{v}}(\phi) \triangleq \left[\begin{array}{cccc} \overline{\mathbf{v}}_1^T(\phi) & \overline{\mathbf{v}}_2^T(\phi) & \cdots & \overline{\mathbf{v}}_L^T(\phi) \end{array} \right]^T \tag{2.1.21}$$

$$\overline{\mathbf{w}} \triangleq \left[\begin{array}{cccc} \overline{\mathbf{w}}_1^T & \overline{\mathbf{w}}_2^T & \cdots & \overline{\mathbf{w}}_L^T \end{array} \right]^T, \tag{2.1.22}$$

which are the real counterparts to Equations 2.1.3 and 2.1.4. Writing Equation 2.1.18 as a real inner product gives

$$y(\phi) = \overline{\mathbf{w}}^T \overline{\mathbf{v}}(\phi) \tag{2.1.23}$$

$$\mathbf{y} = \overline{\mathbf{V}}^T \overline{\mathbf{w}}, \tag{2.1.24}$$

the real counterparts to Equation 2.1.5 and 2.1.6, respectively.

Although the beampattern in Equation 2.1.18 cannot be simplified further without restrictive assumptions, there is another way to exploit the symmetry of the array. The even/odd symmetry of the cosine/sine functions allows any beampattern to be "mirrored" easily to any of the four quadrants. A given beampattern is mirrored across the array axis by changing

$$\begin{array}{rcl} \widetilde{c}_l^{\mathcal{R}} & \mapsto & -\widetilde{c}_l^{\mathcal{R}} \\ \widetilde{c}_l^{\mathcal{I}} & \mapsto & -\widetilde{c}_l^{\mathcal{I}} \end{array} . \tag{2.1.25}$$

This transformation negates the cross-axial component and yields the same response as the original weight on the opposite side of the array. A similar transformation allows cross-axial mirroring, or mirroring from forward to aft:

$$\begin{array}{rcl} \widetilde{a}_l^{\mathcal{I}} & \mapsto & -\widetilde{a}_l^{\mathcal{I}} \\ \widetilde{b}_l^{\mathcal{R}} & \mapsto & -\widetilde{b}_l^{\mathcal{R}} \\ \widetilde{c}_l^{\mathcal{I}} & \mapsto & -\widetilde{c}_l^{\mathcal{I}} \end{array} . \tag{2.1.26}$$

In this case, the sign changes are a combination of conjugation and negation of the axial component. Performing both axial and cross-axial mirroring, one beam is mir-

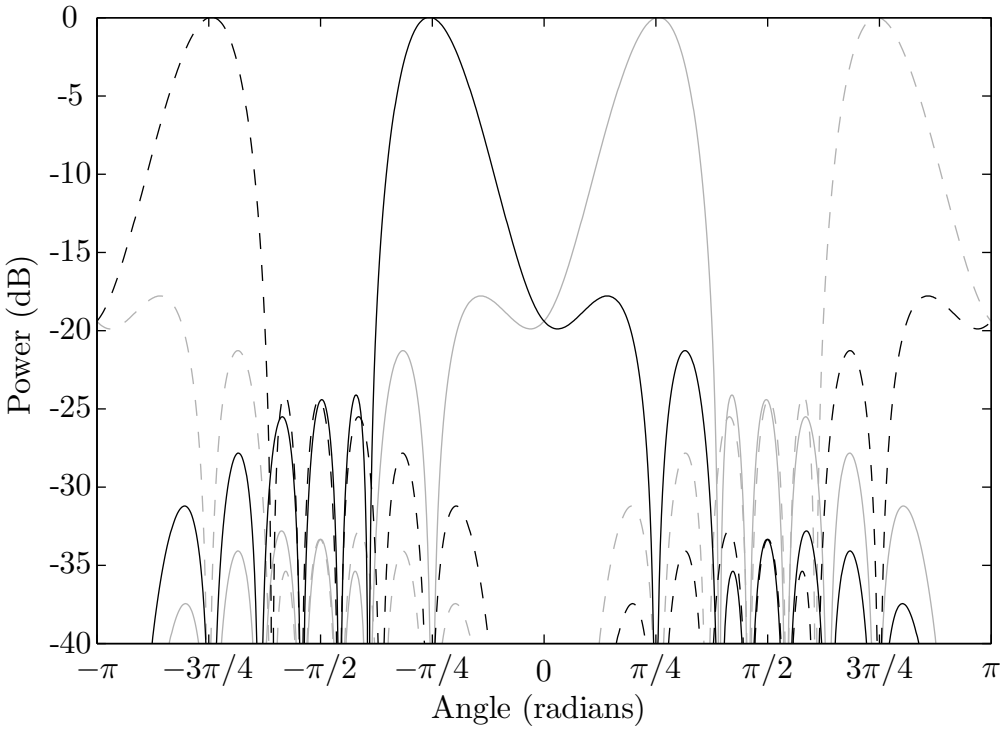

Figure 2.1.1: A VSA beampattern is easily "mirrored" to any quadrant

rored to any of the four quadrants. Figure 2.1.1 provides an example beampattern that is mirrored from one quadrant to the other three. The beampattern shown is for a uniform linear vector-sensor array with $N = 10$, $\phi_0 = -\pi/4$, and $f = 5/7 f_d$. In addition to being linear transformations, the mirroring operations only involve sign changes. Mirroring allows efficient conventional beamforming because *a single quadrant of partial sums from each sensor type forms a full set of beams spanning all quadrants with only sign changes.* Mirroring also reduces the effort required to design a set of beams by a factor of four.

2.2 Robustness and the Sensitivity Factor

The sensor and propagation models used in array processing often contain appreciable errors, or "mismatch." A significant source of mismatch is imperfection in the array itself: the exact gains, phases, positions, and orientations of the sensors are unknown. The "sensitivity factor" of a weight vector quantifies its robustness to these modeling

errors. The (normalized) sensitivity factor of a VSA weight vector, $\mathbf{w}$, is

$$\begin{aligned} \xi &\triangleq \mathbf{v}_0^H \mathbf{v}_0 \ \mathbf{w}^H \mathbf{w} \\ &= 2N \ \mathbf{w}^H \mathbf{w}. \end{aligned} \tag{2.2.1}$$

For weights subject to a unity gain constraint, the Cauchy-Schwarz inequality implies $\xi \geq 1$. The sensitivity factor fully characterizes the deviation of a pressure-sensor array beampattern under Gaussian errors (see [5, §2.6.3]). The relationship is more complex for vector-sensor arrays ([15]), but ξ remains an effective surrogate measure for the sensitivity of a weight vector to mismatch. Robustness generally decreases as ξ increases, so constraining ξ to be small provides robustness in adaptive beamforming [10]. Section 4.3.3 applies a similar technique to fixed weight design.

2.3 Properties of Linear Vector-Sensor Arrays

Because a pressure-sensor array is a subset of any vector-sensor array, it seems possible that many useful properties of linear pressure-sensor arrays extend to vector-sensor arrays. However, the additional complexity of vector-sensors makes it necessary to re-examine these properties because many require modification or do not apply.

2.3.1 Local Fourier Transform Property

One useful property of linear pressure-sensor arrays is that the beampattern is simply the discrete Fourier transform of the weights. This relationship enables using Fourier transform properties and the tools of FIR filtering in design and analysis. The first entry in Equation 2.1.10, corresponding to the pressure-sensor, is the complex exponential of a Fourier transform vector. For the other components, however, the Fourier transform relationship does not hold because of the $\sin\phi$ and $\cos\phi$ terms outside the exponential.

Although the exact Fourier transform relationship is not valid for vector-sensor arrays, a useful "local" Fourier transform property exists. A small region around any

nominal point in a linear VSA beampattern behaves like a Fourier transform. The gain of the directional sensors is approximately constant around the nominal angle. Treating the directional sensors as constant introduces some error, $\Delta\mathbf{v}(\phi)$, into the array manifold vectors. The error in the resulting beampattern is

$$\begin{aligned} |\Delta y(\phi)| &= \left|\mathbf{w}^H \left[\Delta\mathbf{v}(\phi)\right]\right| \\ &\leq |\mathbf{w}|\ |\Delta\mathbf{v}(\phi)|. \end{aligned} \tag{2.3.1}$$

The bound in Equation 2.3.1 arises from the Cauchy-Schwarz inequality and is not necessarily tight. If the sensitivity factor is bounded, $\xi \leq \alpha^2$, the magnitude of the weight is bounded, $|\mathbf{w}| \leq \alpha/\sqrt{2N}$. The error in the manifold vector comes only from errors in the directional terms in Equation 2.1.18:

$$\begin{aligned} |\Delta\mathbf{v}(\phi)|^2 &= 4 \sum_{l=1}^{L} \cos^2(\widetilde{d}_l k_0 \cos\phi)(\Delta\cos\phi)^2 + \cos^2(\widetilde{d}_l k_0 \cos\phi)(\Delta\sin\phi)^2 \\ &\quad + \sin^2(\widetilde{d}_l k_0 \cos\phi)(\Delta\cos\phi)^2 + \sin^2(\widetilde{d}_l k_0 \cos\phi)(\Delta\sin\phi)^2 \\ &= 2N \cdot [(\Delta\cos\phi)^2 + (\Delta\sin\phi)^2]. \end{aligned} \tag{2.3.2}$$

Substituting the bound on $|\mathbf{w}|$ and the expression for $|\Delta\mathbf{v}(\phi)|$ into Equation 2.3.1 gives

$$\begin{aligned} |\Delta y(\phi)| &\leq \frac{\alpha}{\sqrt{2N}} \sqrt{2N \cdot [(\Delta\cos\phi)^2 + (\Delta\sin\phi)^2]} \\ &= 2\alpha \sin\left(\frac{\Delta\phi}{2}\right) \qquad (2.3.3) \\ &\leq \alpha \cdot (\Delta\phi). \qquad (2.3.4) \end{aligned}$$

The last inequality is tight near the nominal angle. Equations 2.3.3 and 2.3.4 are useful for two reasons. First, they prove that in a small region ($\Delta\phi$ much less than a beamwidth) around any point, the beampattern approximately equals a weighted

Fourier transform. Because the pressure-sensor beampattern is a Fourier transform, the vector-sensor beampattern, around a given angle, behaves like a pressure-sensor beampattern. Second, because the deviation of the pressure-sensor beampattern between two sample points is bounded, Equations 2.3.3 and 2.3.4 prove that the deviation of the vector-sensor beampattern between two sample points is also bounded. In Chapter 4, this bounded deviation allows a vector-sensor beampattern to be approximated by a finite set of points with negligible error.

2.3.2 No Modulation or "Steering" Property

The Fourier transform relationship between the weights and the beampattern for a linear pressure-sensor array has many useful implications. One such implication is that modulating the phase of the weights "steers" an arbitrary beampattern (viewed in cosine-space) to any angle. The steered beampattern and the original beampattern have the same shape in cosine-space. In practice, the steering property means that only one real weight – a taper – designed at array broadside is sufficient to form identical beams anywhere.

As useful as this property is for linear pressure-sensor arrays, it does not apply to linear vector-sensor arrays. Like the Fourier transform relationship, the modulation property takes a modified, weakened form with vector-sensor arrays. Separating the vector-sensor measurements into phase and gain components reveals that 1) the phase component exhibits a modulation property in *cosine-space* like a linear pressure-sensor array, and 2) the gain component is rotated in *angle-space* by Euler rotations. Although each rotation is a linear transformation of the weight (or equivalently, the data), the gain and phase components cannot be separated by a linear system. Thus, no linear transformation steers a vector-sensor array beampattern to an arbitrary direction. Although the lack of a steering property means each beam must be designed separately, the "mirroring" techniques illustrated in Figure 2.1.1 provide a useful way to reduce the design burden.

2.3.3 Non-Polynomial Beampattern

Another useful uniform, linear, PSA result associated with the Fourier transform property is that the beampattern is a polynomial function of the variable $z = \cos(dk_0 \cos\phi)$. This well-known result is easily seen from Equation 2.1.18 by removing the directional elements, assuming real weights, and applying a Chebyshev polynomial relation to each nested cosine term. The polynomial form of the uniform, linear, PSA beampattern forms the foundation of many tools including Chebyshev, Taylor, and Villeneuve tapers and the Parks-McClellan algorithm [5, §3.4 and 3.6]. Such tools apply polynomial factorizations or approximations to the beampattern in z-space.

Unfortunately, the vector-sensor array beampattern does not have a similar polynomial form. For a polynomial beampattern representation to be useful, it must be an unconstrained polynomial in some real function $z(\phi)$. The following discussion suppresses the dependence of z on azimuth angle, ϕ, when convenient. The single-sensor case illustrates why such a representation does not exist for vector-sensor arrays. Two beampatterns possible with a single vector-sensor are those given by the axial and cross-axial directional sensors:

$$y_0(\phi) = \cos\phi \tag{2.3.5}$$

$$y_1(\phi) = \sin\phi. \tag{2.3.6}$$

Assume that an unconstrained polynomial representation *does* exist for some function z. Because a 2-D vector-sensor beampattern involves three weights, both beampatterns must correspond to unconstrained, quadratic polynomials in z, that is,

$$\cos\phi = a_0 z^2 + b_0 z + c_0 \tag{2.3.7}$$

$$\sin\phi = a_1 z^2 + b_1 z + c_1 \tag{2.3.8}$$

for some real coefficients a_0, a_1, b_0, *etc.* Breaking the first equation into the $a_0 \neq 0$ and $a_0 = 0$ (quadratic and linear) cases and solving, Equation 2.3.7 constrains the

function z to lie in one of the two sets of functions

$$\mathcal{Q}_0 = \left\{ \frac{-b_0 + s(\phi)\sqrt{b_0^2 - 4a_0(c_0 - \cos\phi)}}{2a_0} \;\middle|\; \begin{array}{c} a_0 \neq 0,\ s^2(\phi) = 1, \\ b_0^2 - 4a_0c_0 \geq |4a_0| \end{array} \right\} \tag{2.3.9}$$

$$\mathcal{L}_0 = \left\{ \frac{\cos\phi - c_0}{b_0} \;\middle|\; a_0 = 0 \right\}. \tag{2.3.10}$$

The sign function $s(\phi)$ takes only values of ± 1. To reconstruct both beampatterns, any function is further constrained by Equation 2.3.8. The functions in $\mathcal{L}_0$ are even and cannot construct the odd sine function via the composition in Equation 2.3.8. For any function in $\mathcal{Q}_0$ to convey sign information about ϕ, the sign function $s(\phi)$ must be odd. The form of z is thus restricted to a constant (even) part plus the odd part involving $s(\phi)$. The even part of the z function must become identically zero when substituted into Equation 2.3.8, leaving the requirement that

$$\sin\phi \propto s(\phi)\sqrt{1 + \alpha\cos\phi} \tag{2.3.11}$$

for some real coefficient α. For this to be true and continuous at the origin, $\alpha = -1$ and

$$|\sin\phi| \propto \sqrt{1 - \cos\phi}. \tag{2.3.12}$$

Because this is clearly not true, no function in $\mathcal{Q}_0$ satisfies Equations 2.3.8, *i.e.* no z function satisfies both Equations 2.3.7 and 2.3.8. Thus, no unconstrained polynomial form exists for the single-sensor beampatterns in Equations 2.3.5 and 2.3.6. Furthermore, because these beampatterns are possible with any vector-sensor array, VSA beampatterns generally do not have an unconstrained polynomial representation.

Although having a non-polynomial beampattern nullifies the polynomial-based techniques listed above, it does not mean equivalent results are impossible with acoustic vector-sensor arrays. As Chapter 4 shows, equivalent beampatterns are achievable with techniques not based on polynomial functions.

2.3.4 Weights With Nonlinear Element-Wise Phase

Weights exhibiting linear element-wise phase are another useful implication of the Fourier transform property of linear pressure-sensor arrays. The modulation property makes real weights designed at array broadside sufficient for use at any cosine angle. Real weights, when modulated to another cosine angle, become complex weights with a linear element-wise phase progression.

Because every replica vector on the linear vector-sensor array manifold exhibits a linear element-wise phase progression, it seems that the weights should necessarily exhibit this property as well. To the contrary, Appendix A.2 suggests that vector-sensor weights need not have linear element-wise phase. Chapter 4 proves the existence of such weights by example: many of the custom-designed weights have nonlinear element-wise phase progressions. Although weights with nonlinear element-wise phase stray from the concept of "spatial tapering," they often perform well with vector-sensor arrays. Depending on the design problem, forcing VSA weights to have linear element-wise phase may sacrifice significant performance.

2.3.5 Nonlinear Physical Constraints

A final property of vector-sensor arrays that deserves clarification is the nonlinearity of the physical constraints. The four measurements of a single vector-sensor are somewhat redundant. The omnidirectional sensor measures the pressure field; the directional sensors measure the gradient of the pressure field.

Although it seems that this redundancy should be easy to exploit, its nonlinear nature leads to complications. Even in the simplest case of the single plane-wave source, the measurements are related quadratically by power. For a single plane-wave source, the sum of the power measured by the directional sensors equals the total power measured by the omnidirectional sensor. With multiple sources, the relationship becomes even more complex and nonlinear. Full exploitation of such physical constraints requires nonlinear techniques such as [12].

2.4 Spatially Spread Sources with Linear VSAs

It is common scientific practice to simplify problems by discretizing distributions: point masses in physics, impulses and pure sinusoids in signal processing, and point sources in acoustics. Although these approximations are often extremely accurate, they are sometimes misleading. Modeling spatially distributed sounds as point sources occasionally predicts performance that is unachievable in practice. To avoid such a pitfall, this section derives integrals and approximations for 2-D spatially spread sources as observed by a linear vector-sensor array. The 2-D vector-sensor array is modeled for simplicity, but extending these results to 3-D is discussed where applicable. Assuming all spatial processes are zero-mean Gaussian, the quantity of interest is most generally the covariance between two sensors. Because a 2-D vector-sensor measures three quantities, this covariance is size 3×3 for a single pair and $3N \times 3N$ for an array of N sensors.

Figure 2.4.1 provides a notional comparison of point and spread sources. The point source corresponds to the impulsive spatial distribution denoted by the gray arrow in the top plot. The response of a vector-sensor array to this point source gives the familiar beampattern shown in the bottom plot. The spatially spread source, by contrast, corresponds to a uniform spatial distribution in cosine-space on the starboard side of the array. The array response to both distributions exhibits sidelobe structure because of the finite aperture and "backlobe" structure because of the pressure ambiguity. The spread source integrates power over a range of angles, "filling-in" nulls and widening the array response. The spatial spreading in Figure 2.4.1 is exaggerated to illustrate its effects on the array response; spatial distributions are often more concentrated than the figure suggests.

2.4.1 General Integral Form

Although spatially spread sources are unexplored with linear vector-sensor arrays, they are common with linear pressure-sensor arrays. Because a pressure-sensor array is a subset of a vector-sensor array, this thesis carefully chooses a spatial spread-

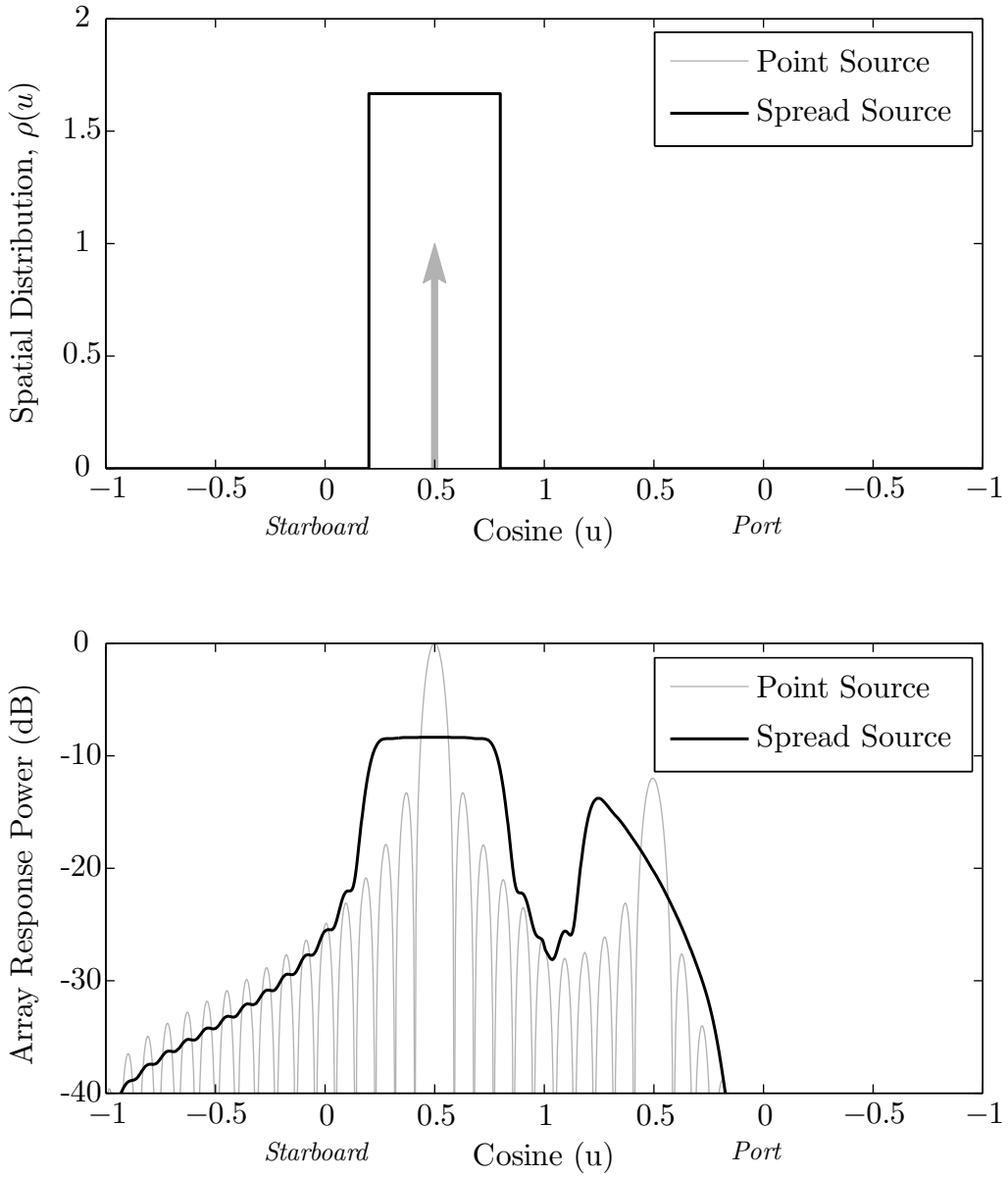

Figure 2.4.1: Notional response of a VSA to point and spread sources

ing model consistent with the decades of vetted pressure-sensor work. This section extends the model presented in [5, §8.9], analyzing an azimuthal distribution of uncorrelated, zero-mean, Gaussian sources. The distribution is specified in terms of azimuthal cosine (rather than angle), keeping with convention, encouraging closed-form expressions, and restricting the distribution to one side of the array. The results are easily extended to two-sided distributions by expressing any two-sided distribution as the sum of two, one-sided distributions. Because the integrated sources are uncorrelated, the covariance between two sensors is given by the single integral

$$r_{01} = \int_{-1}^{+1} \rho(u) v_0(u) v_1^*(u)\, du \tag{2.4.1}$$

where $u = \cos\phi$ is the azimuthal cosine, $\rho(u)$ is the spatial distribution of power, and $v_i(u)$ are the responses of each sensor to a signal at u. When the two sensors are part of a linear vector-sensor array, each response contains a gain term depending only on direction and a phase term depending on both position and direction. If the sensor position along the array axis is x and the gain of each sensor is $g_i(u)$, the response is

$$v_i(u, x) \triangleq g_i(u) e^{jk_0 xu}. \tag{2.4.2}$$

The gain terms for the geophone elements are simply the azimuthal sine and cosine expressed in terms of u. Using subscripts o, x, y for the omnidirectional, inline, and cross-axial sensors, these gain terms are

$$g_o(u) = 1 \tag{2.4.3}$$

$$g_x(u) = u \tag{2.4.4}$$

$$g_y(u) = \pm\sqrt{1 - u^2}. \tag{2.4.5}$$

The sign of $g_y(u)$ changes depending on the side of the array. The remainder of this section assumes $g_y(u)$ is positive, corresponding to a distribution on the port side of

the array. Substituting Equation 2.4.2 into Equation 2.4.1 gives

$$\begin{aligned} r(x_0, x_1) &= \int_{-1}^{+1} \rho(u) g_0(u) g_1^*(u) e^{jk_0 x_0 u} e^{-jk_0 x_1 u}\, du \\ &= \int_{-1}^{+1} \rho(u) g_0(u) g_1^*(u) e^{jk_0 (x_0 - x_1) u}\, du \end{aligned} \tag{2.4.6}$$

Equation 2.4.6 is easily written in terms of the distance between the sensors, $\delta \triangleq x_0 - x_1$, and the composite gain function of the sensor pair, $G_{01}(u) \triangleq g_0(u) g_1^*(u)$:

$$r(\delta) \triangleq \int_{-1}^{+1} \rho(u) G_{01}(u) e^{jk_0 \delta u}\, du. \tag{2.4.7}$$

Extending the covariance function, $r(\delta)$, to 3-D vector-sensor arrays requires no additional work as the elevation terms fall outside the integral. The integral in Equation 2.4.7 is the windowed Fourier transform of $\rho(u) G_{01}(u)$, so a closed form seems possible. Unfortunately, the number and variety of gain functions make obtaining closed forms for *all* integrals very difficult with a given spatial distribution. The exact integral form in Equation 2.4.7 does, however, admit several useful and insightful approximations.

2.4.2 Constant Gain Approximation

The simplest and most useful approximation to Equation 2.4.7 arises from the smooth nature of the gain functions and the small width of typical spatial spreading. The standard deviation of the distribution is usually small (less than 5% of cosine-space) when modeling spatially spread sources. Over such a small range of u, the gain functions are well-approximated as constant. This constant gain approximation yields a simple but powerful model for vector-sensor spatial spreading using covariance matrix tapers.

When the sensor gains are approximated as constant, incorporating spatial spreading simply modulates the existing covariance function. Without loss of generality,

assume the spatial distribution has mean u_0 and is a shifted version of the zero-mean distribution $\rho_0(u)$. Applying the constant gain approximation to Equation 2.4.7 at u_0 allows the gain terms to be taken outside the integral:

$$r(\delta) \approx g_0(u_0)g_1(u_0) \int_{-1}^{+1} \rho_0(u-u_0)e^{jk_0\delta u}\, du. \tag{2.4.8}$$

Equation 2.4.8 is simplified in two steps. The first step is extending the range of the distribution to include the entire real line. The extended region $u \notin [-1,+1]$ is referred to as "virtual" space because it does not correspond to real azimuthal angles. It does, however, provide a natural extension for spatially spread sources at array endfire, where the distribution extends into virtual space. The second step is utilizing a Fourier transform property to simplify the integral. Translation in the u domain corresponds to a phase shift in the δ domain. Applying both steps to Equation 2.4.8 gives

$$\begin{aligned} r(\delta) &\approx g_0(u_0)g_1(u_0) \int_{-\infty}^{+\infty} \rho_0(u-u_0)e^{jk_0\delta u}\, du \\ &= g_0(u_0)g_1(u_0)e^{jk_0\delta u_0} \cdot P_0(k_0\delta), \end{aligned} \tag{2.4.9}$$

where

$$P_0(k_0\delta) \triangleq \int_{-\infty}^{+\infty} \rho_0(u)e^{jk_0\delta u}\, du \tag{2.4.10}$$

is the Fourier transform of the distribution $\rho_0(u)$. Equation 2.4.9 is divided into two terms. The first term is the original covariance function without spatial spreading. The effects of spatial spreading appear as a modulation by the second term, $P_0(k_0\delta)$. This modulating term, or "spread function," is independent of the source location given by the mean, u_0. Because it is a Fourier integral, the spread function often has a closed form. Two common and tractable choices for $\rho_0(u)$ are the uniform and Gaussian distributions. These distributions and their associated spread functions are summarized in the table

	$\rho_0(u)$	$P_0(k_0\delta)$
Uniform	$\begin{cases} 1/(2\sigma_u\sqrt{3}) & \lvert u\rvert \le \sigma_u\sqrt{3} \\ 0 & \text{otherwise} \end{cases}$	$\operatorname{sinc}(k_0\delta\sigma_u\sqrt{3})$
Gaussian	$\dfrac{1}{\sqrt{2\pi\sigma_u^2}} \exp\left\{-\dfrac{u^2}{2\sigma_u^2}\right\}$	$\exp\{-(k_0\delta\sigma_u)^2/2\}$

,

where $\operatorname{sinc}(\cdot)$ is the unnormalized sinc function and σ_u is the standard deviation of the distribution in cosine-space. Equation 2.4.9 reveals that 1) the effects of spatial spreading are well-approximated by modulation and 2) the modulating spread function does not depend on source location.

For an array of sensors, the constant gain approximation enables modeling spatial spreading with covariance matrix tapers. Each entry of the covariance matrix (for a single source) is Equation 2.4.9 evaluated at the correct inter-element distance. Separating the terms of Equation 2.4.9 reveals that modulating the point source covariance matrix, $\mathbf{R}$, approximates the spatially spread covariance matrix, $\mathbf{R}_s$:

$$\mathbf{R}_s \approx \mathbf{R} \odot \mathbf{P}. \tag{2.4.11}$$

The modulation matrix, $\mathbf{P}$, is given by the spread function and does not depend on the contents of $\mathbf{R}$. By linearity, any covariance matrix that is the sum of a (possibly infinite) number of point sources is approximately "spread" by applying the same modulation matrix. The matrix $\mathbf{P}$ is often referred to as a "covariance matrix taper" because of its similarity to temporal or spatial tapering. The three components of each vector-sensor are co-located, so the covariance matrix taper for a vector-sensor is simply an extension of the taper for omnidirectional sensor,

$$\mathbf{P}_{\text{2D-VSA}} = \mathbf{P}_{\text{PSA}} \otimes \mathbf{1}_{3\times 3}. \tag{2.4.12}$$

For a 3-D VSA, the 3×3 matrix of ones is replaced by a 4×4 matrix of ones.

Figure 2.4.2 reveals how accurately the constant gain approximation models uniform spatial spreading. The figure illustrates response patterns for an $N = 10$ element, 2-D vector-sensor array at its design frequency. As shown in the figure, both the constant gain approximation and the exact integral expand the point source and "fill-in" the nulls in the response pattern. For a typical case with reasonably small spreading away from array endfire, the approximation is almost indistinguishable from the exact integral. In a more extreme case with large spreading near array endfire, the errors introduced by the approximation are minor but visible. The approximation is less accurate at endfire where the sensor gains may change rapidly with u. If the "extreme" case were moved any closer to endfire, spreading would extend into virtual space; the approximation would be useful but the exact integral would be undefined.

2.4.3 Second-Order Gain Approximation

The previous section applies a zeroth-order Taylor series approximation to the sensor gains, *i.e.* the gains are approximated as constant. This section explores higher-order approximations and develops a second-order approximation for uniform spatial spreading. Higher-order approximations become increasingly accurate at the expense of analytical simplicity. Any approximation greater than zero-order loses the simplicity and power of the covariance matrix taper interpretation.

A closed form expression is first derived for any polynomial gain function. Consider the n^{th}-order gain function

$$G_n(u) \triangleq (u - u_0)^n, \tag{2.4.13}$$

which is a simple monomial in $\tilde{u} \triangleq u - u_0$. Extending the integral into virtual space transforms the covariance function (Equation 2.4.7) into

$$\begin{aligned} r_n(\delta) &\triangleq \int_{-\infty}^{+\infty} G_n(u)\rho(u)e^{jk_0\delta u}\,du \\ &= \int_{-\infty}^{+\infty} (u - u_0)^n\,\rho_0(u - u_0)e^{jk_0\delta u}\,du. \end{aligned} \tag{2.4.14}$$

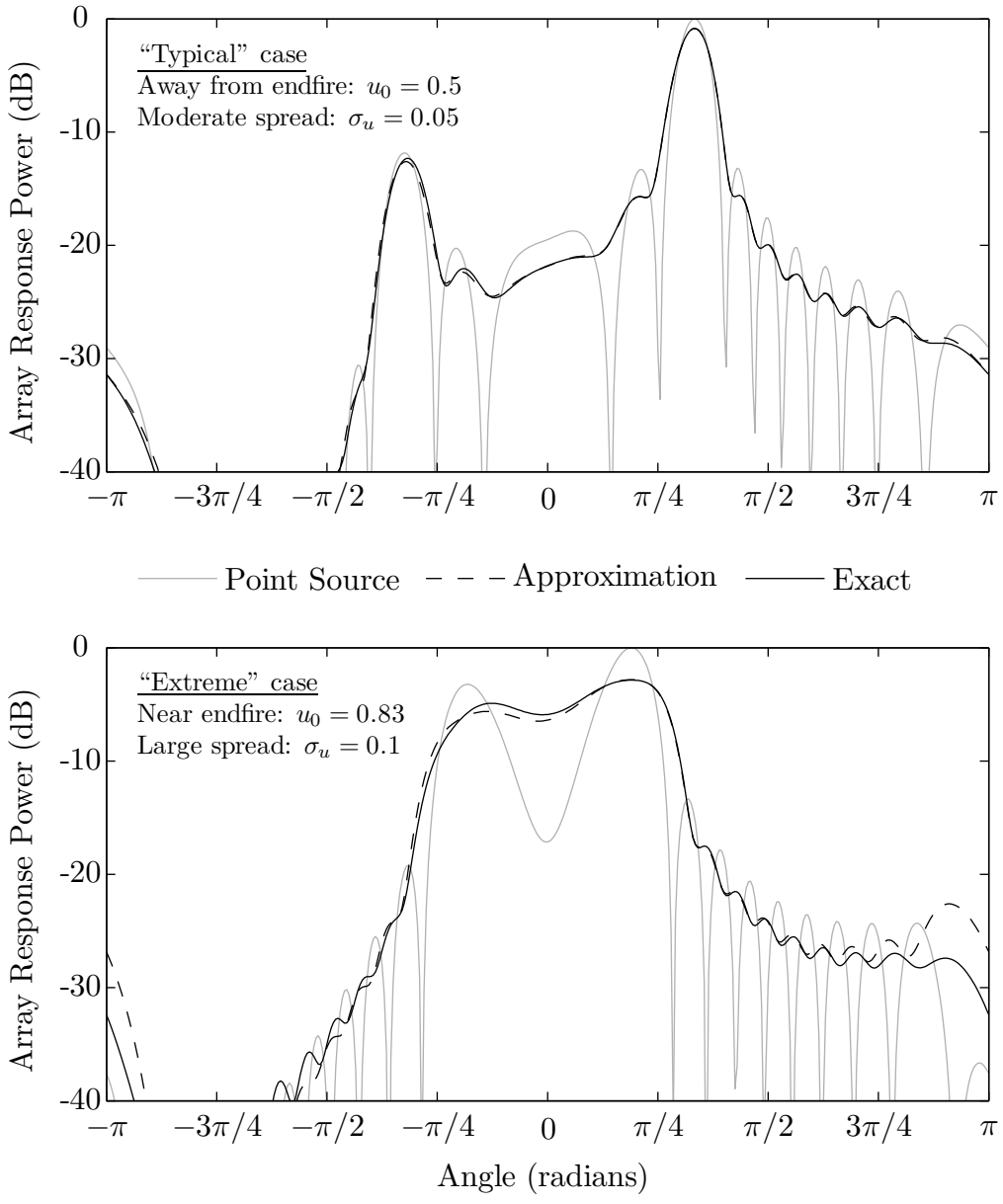

Figure 2.4.2: Constant gain approximation to uniform spatial spreading

Equation 2.4.14 is an inverse Fourier transform, so 1) translation in one domain corresponds to a phase shift in the other and 2) modulation by a monomial in one domain corresponds to differentiation in the other. Applied in sequence to Equation 2.4.14, these properties yield the closed form

$$\begin{aligned} r_n(\delta) &= e^{jk_0\delta u_0}\int_{-\infty}^{+\infty}\tilde{u}^n\rho_0(\tilde{u})e^{jk_0\delta\tilde{u}}\,d\tilde{u} \\ &= e^{jk_0\delta u_0}j^{-n}P_0^{(n)}(k_0\delta), \end{aligned} \tag{2.4.15}$$

where $P_0^{(n)}$ is the n^{th} derivative of P_0 with respect to its argument. Any gain function that is an n^{th}-order polynomial is expressible as a linear combination of $G_0(u),\ G_1(u),\ \ldots,\ G_n(u)$. It therefore has a closed form covariance function as a linear combination of $r_0(\delta),\ r_1(\delta),\ \ldots,\ r_n(\delta)$.

For acoustic vector-sensors, a second-order Taylor series is a convenient approximation to the gain functions because most terms are exact. Six composite gain functions must be integrated to fill each 3×3 block in the covariance matrix:

$$G_{oo}(u) = 1 \tag{2.4.16}$$

$$G_{ox}(u) = u \tag{2.4.17}$$

$$G_{xx}(u) = u^2 \tag{2.4.18}$$

$$G_{yy}(u) = 1-u^2 \tag{2.4.19}$$

$$G_{oy}(u) = \sqrt{1-u^2} \tag{2.4.20}$$

$$G_{xy}(u) = u\sqrt{1-u^2}. \tag{2.4.21}$$

A second-order Taylor series is exact for the first four terms. Over a small region of u, the last two terms are well-approximated by second-order expansions about u_0,

$$G_{oy}(u) \approx \sqrt{1-u_0^2} - \frac{u_0}{\sqrt{1-u_0^2}}(u-u_0) - \frac{1/2}{(1-u_0^2)^{3/2}}(u-u_0)^2 \tag{2.4.22}$$

$$G_{xy}(u) \approx u_0\sqrt{1-u_0^2} + \frac{1-2u_0^2}{\sqrt{1-u_0^2}}(u-u_0) + \frac{u_0(2u_0^2-3)}{2(1-u_0^2)^{3/2}}(u-u_0)^2. \tag{2.4.23}$$

Obtaining closed form covariance functions for this second-order approximation requires the derivatives $P_0^{(n)}(z)$ for $n = 1, 2$. These derivatives are easily computed for many distributions, but only uniform spreading is included here. For uniform spreading of width $\Delta \triangleq \sigma_u\sqrt{3}$, Section 2.4.2 gives

$$P_0(z) = j_0(z\Delta). \tag{2.4.24}$$

The zeroth-order spherical Bessel function, $j_0(\cdot)$, is equivalent to the unnormalized sinc function. Derivatives of the spherical Bessel functions are related through the following recursion [16, 10.1.20]:

$$\frac{d}{dz}j_n(z) = \frac{n}{2n+1}j_{n-1}(z) - \frac{n+1}{2n+1}j_{n+1}(z). \tag{2.4.25}$$

This recursion enables writing each $P_0^{(n)}(z)$ as a linear combination of the first n spherical Bessel functions. Applying Equation 2.4.25 gives the first two derivatives,

$$\begin{aligned} j^{-1}P_0^{(1)}(z) &= j^{-1}\frac{d}{dz}j_0(z\Delta) \\ &= j\Delta j_1(z\Delta) \end{aligned} \tag{2.4.26}$$

$$\begin{aligned} j^{-2}P_0^{(2)}(z) &= j^{-2}\frac{d^2}{dz^2}j_0(z\Delta) \\ &= j^{-1}\frac{d}{dz}\left[j\Delta j_1(z\Delta)\right] \\ &= \Delta^2\left[\frac{1}{3}j_0(z\Delta) - \frac{2}{3}j_2(z\Delta)\right]. \end{aligned} \tag{2.4.27}$$

Substituting these derivatives into Equation 2.4.15 yields

$$r_0(\delta) = e^{jk_0\delta u_0}\cdot j_0(k_0\delta\Delta) \tag{2.4.28}$$

$$r_1(\delta) = e^{jk_0\delta u_0}\cdot j\Delta j_1(k_0\delta\Delta) \tag{2.4.29}$$

$$r_2(\delta) = e^{jk_0\delta u_0}\cdot \Delta^2\left[\frac{1}{3}j_0(k_0\delta\Delta) - \frac{2}{3}j_2(k_0\delta\Delta)\right]. \tag{2.4.30}$$

The gain functions in Equations 2.4.16-2.4.19, 2.4.22, and 2.4.23 are second-order polynomials, so their covariance functions require only the first three $r_n(\delta)$:

$$r_{oo}(\delta) = r_0(\delta) \tag{2.4.31}$$

$$r_{ox}(\delta) = u_0 r_0(\delta) + r_1(\delta) \tag{2.4.32}$$

$$r_{xx}(\delta) = u_0^2 r_0(\delta) + 2u_0 r_1(\delta) + r_2(\delta) \tag{2.4.33}$$

$$r_{yy}(\delta) = (1-u_0^2) r_0(\delta) - 2u_0 r_1(\delta) - r_2(\delta) \tag{2.4.34}$$

$$r_{oy}(\delta) \approx \sqrt{1-u_0^2}\, r_0(\delta) - \frac{u_0}{\sqrt{1-u_0^2}} r_1(\delta) - \frac{1/2}{(1-u_0^2)^{3/2}} r_2(\delta) \tag{2.4.35}$$

$$r_{xy}(\delta) \approx u_0\sqrt{1-u_0^2}\, r_0(\delta) + \frac{1-2u_0^2}{\sqrt{1-u_0^2}} r_1(\delta) + \frac{u_0(2u_0^2-3)}{2(1-u_0^2)^{3/2}} r_2(\delta). \tag{2.4.36}$$

Although it is not done here, substituting $r_n(\delta)$ into these equations yields closed forms in terms of the first three spherical Bessel functions.

The covariance functions obtained with the second-order approximation are more accurate but more complex than those obtained with the constant gain approximation. The complexity of the second-order approximation is evident in the lengthy covariance functions and the coupling of spreading width, Δ, and mean, u_0. This coupling makes a covariance matrix taper interpretation impossible and impedes analysis. The accuracy of the second-order approximation is shown in Figure 2.4.3 for the same cases as Figure 2.4.2. Comparing the two figures reveals that the second-order approximation introduces little error, even with large spreading at endfire where the Taylor series is less accurate (see Section 2.4.2). For the purposes of this document, the constant gain approximation is chosen hereafter for its simplicity, intuition, and reasonable accuracy.

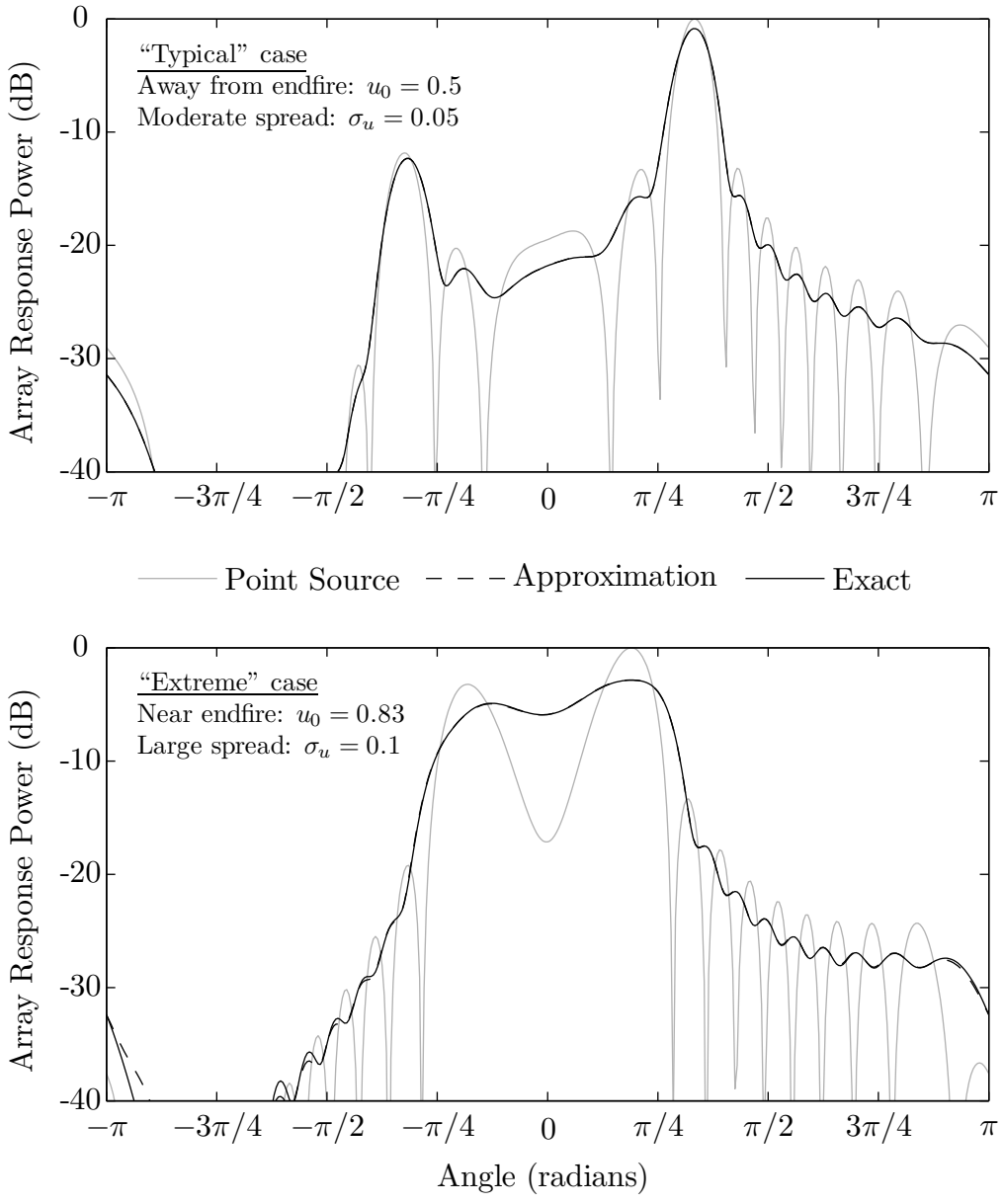

Figure 2.4.3: Second-order approximation to uniform spatial spreading

Chapter 3

Performance Limits of Vector-Sensor Arrays

Section 1.4.1 demonstrates that the bounds commonly employed in pressure-sensor array processing do not necessarily reflect the improvements offered by vector-sensors. To understand this discrepancy and motivate alternative bounds, the following paragraphs formalize the process by which existing bounds arose.

The first step in establishing a bound is identifying the relevant performance dimension. This performance dimension is often revealed by the engineering problem itself. Pressure-sensor arrays developed to localize sound, so a natural performance dimension is angular resolution. Viewed another way, arrays of sensors are employed to amplify sound from a specific direction, so another popular performance dimension is array gain.

The second step toward a useful bound is choosing a scenario or model that accurately represents the given performance dimension. Like a good scientific experiment, the scenario should isolate the performance dimension as a dependent variable and clearly define the independent variables. Useful scenarios often fall into two categories: classification problems and estimation problems. For localization, the standard scenario is estimating the direction-of-arrival of signals in noise; for gain, it is detecting the presence of signals in noise.

The third and final step is deciding the type of bound and applying it to the chosen scenario. This decision depends on the goals of the researcher, the complexity of the problem, the required strength or "tightness" of the bound, and the scenario itself. Each bound has an analytical complexity associated with the derivation and a computational complexity associated with its numerical evaluation.

The three-step process outlined above also helps establish useful performance bounds for vector-sensor array processing. For the first step, Section 1.3 illustrates that the relevant performance dimension for vector-sensor arrays is *resolution of pressure ambiguities*. Sections 3.1 and 3.2 proceed through the second and third steps to establish both a classification-based bound and an estimation-based bound. The objective of this chapter is not an exhaustive treatise on vector-sensor performance bounds; the objective is motivating the study of non-traditional problems that better illustrate vector-sensor capabilities.

The key contribution of this chapter is a new theoretical foundation for vector-sensor performance. The two performance bounds developed in this chapter are distinct, but the conclusion is the same: linear vector-sensor arrays are theoretically able to resolve left/right ambiguities very well from almost any direction, at any frequency, and with any number of elements. The performance also seems to be robust, not relying on point nulls or "supergain."

3.1 Novel Classification-Based Bound

One useful scenario that highlights the vector-sensor array's ability to resolve pressure ambiguities is illustrated in Figure 3.1.1. In this scenario, a narrowband sound source is positioned at one of two possible locations relative to a vector-sensor array. Both locations are chosen to yield identical pressure measurements, *e.g.* the left and right sides of a uniform linear array. Because these locations are ambiguous to a pressure-sensor array, this model isolates the vector-sensor performance. Under this setup, the array measures K independent observations of the source corrupted by additive noise

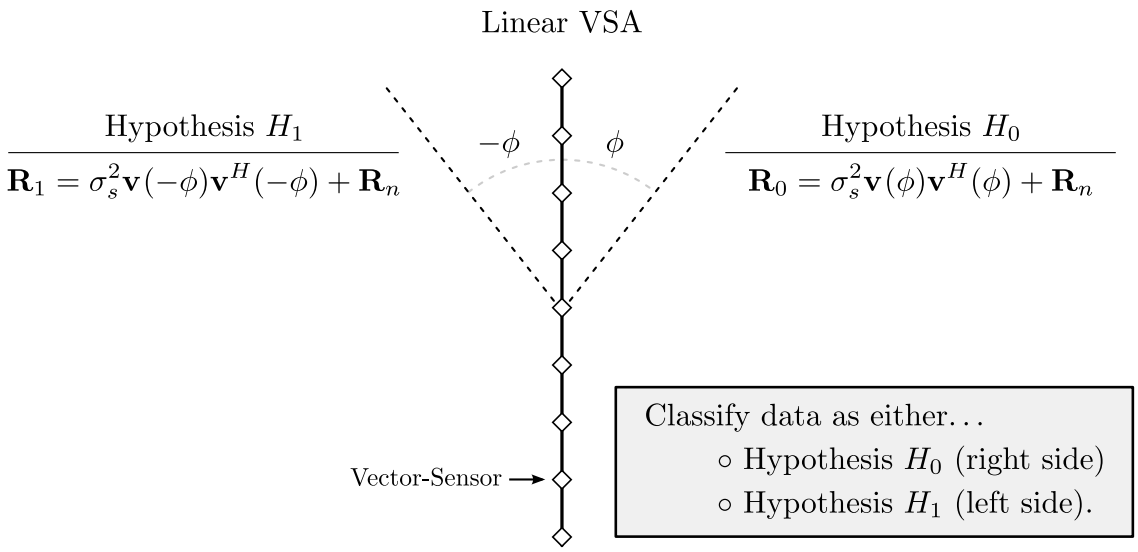

Figure 3.1.1: Example right/left (binary) classification scenario

and/or interference. Each observation, $\mathbf{x}$, is given by

$$\mathbf{x} = \mathbf{x}_s + \mathbf{x}_n \tag{3.1.1}$$

where $\mathbf{x}_s$ and $\mathbf{x}_n$ are the signal and noise measurements, respectively. To simplify expressions, the observation column vectors are horizontally concatenated into the $4N \times K$ matrix $\mathbf{X}$, which is also used to form the sample covariance matrix $\widehat{\mathbf{R}}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_K \end{bmatrix} \tag{3.1.2}$$

$$\widehat{\mathbf{R}} = \frac{1}{K} \cdot \mathbf{X}\mathbf{X}^H. \tag{3.1.3}$$

The source is complex zero-mean Gaussian with *known* power σ_s^2; the noise is complex zero-mean Gaussian with *known* covariance $E\{\mathbf{x}_n \mathbf{x}_n^H\} = \mathbf{R}_n$. The two hypotheses, H_0 and H_1, have respective prior probabilities π_0 and π_1. The source replica vector is either $\mathbf{v}_0$ or $\mathbf{v}_1$, respectively.

The above scenario is a binary hypothesis test because there are only two classes.

For binary hypotheses, the log-likelihood ratio test

$$\ln \Lambda(\mathbf{X}) = \ln p_{\mathbf{X}|H}(\mathbf{X}|H_1) - \ln p_{\mathbf{X}|H}(\mathbf{X}|H_0) \underset{\text{`}H_0\text{'}}{\overset{\text{`}H_1\text{'}}{\gtrless}} \ln \frac{\pi_0}{\pi_1} \tag{3.1.4}$$

minimizes, and thus bounds, the probability of error. Although this is a binary hypothesis test, it is *not* the standard detection problem because there is no null hypothesis in which the source is absent.

3.1.1 Derivation

Having formally stated the scenario and chosen the type of bound, the task is now to derive an expression for the minimum probability of error. The procedure follows close to [17], but modified here for the complex distributions and matrix measurements. The derivation is in four steps: 1) forming the log-likelihood ratio test, 2) establishing the characteristic functions of the test statistic under both hypotheses, 3) evaluating the cumulative distribution functions from the characteristic functions, and 4) expressing the minimum probability of error in terms of the cumulative distribution functions.

The first step is deriving an expression for the log-likelihood ratio test. Under hypothesis H_i, $i \in \{0, 1\}$, the data matrix, $\mathbf{X}$, is zero-mean complex Gaussian with *known* covariance matrix

$$\mathbf{R}_i = \sigma_s^2 \mathbf{v}_i \mathbf{v}_i^H + \mathbf{R}_n. \tag{3.1.5}$$

The probability density for $\mathbf{X}$ under H_i is a function of the sample covariance matrix only,

$$p_{\widehat{\mathbf{R}}|H}(\widehat{\mathbf{R}}|H_i) = |\pi \mathbf{R}_i|^{-K} \cdot \exp\left[-K \mathrm{tr}(\mathbf{R}_i^{-1}\widehat{\mathbf{R}})\right] \tag{3.1.6}$$

where $|\cdot|$ denotes the matrix determinant. The log-likelihood function under H_i is then

$$\ln p_{\widehat{\mathbf{R}}|H}(\widehat{\mathbf{R}}|H_i) = -K \ln |\pi \mathbf{R}_i| - K \mathrm{tr}(\mathbf{R}_i^{-1}\widehat{\mathbf{R}}), \tag{3.1.7}$$

resulting in the log-likelihood ratio

$$\begin{aligned}\ln \Lambda(\widehat{\mathbf{R}}) &= K\left[-\ln|\pi\mathbf{R}_1| - \mathrm{tr}(\mathbf{R}_1^{-1}\widehat{\mathbf{R}}) + \ln|\pi\mathbf{R}_0| + \mathrm{tr}(\mathbf{R}_0^{-1}\widehat{\mathbf{R}})\right] \\ &= K\cdot\left\{\ln\frac{|\mathbf{R}_0|}{|\mathbf{R}_1|} + \mathrm{tr}\left[(\mathbf{R}_0^{-1}-\mathbf{R}_1^{-1})\widehat{\mathbf{R}}\right]\right\}. \qquad (3.1.8)\end{aligned}$$

Substituting this expression into Equation 3.1.4 yields the log-likelihood ratio test

$$\begin{aligned}K\cdot\left\{\ln\frac{|\mathbf{R}_0|}{|\mathbf{R}_1|} + \mathrm{tr}\left[(\mathbf{R}_0^{-1}-\mathbf{R}_1^{-1})\widehat{\mathbf{R}}\right]\right\} &\underset{\text{`}H_0\text{'}}{\overset{\text{`}H_1\text{'}}{\gtrless}} \ln\frac{\pi_0}{\pi_1} \\ K\cdot\mathrm{tr}\left[(\mathbf{R}_0^{-1}-\mathbf{R}_1^{-1})\widehat{\mathbf{R}}\right] &\underset{\text{`}H_0\text{'}}{\overset{\text{`}H_1\text{'}}{\gtrless}} \ln\frac{\pi_0}{\pi_1} - K\cdot\ln\frac{|\mathbf{R}_0|}{|\mathbf{R}_1|}. \qquad (3.1.9)\end{aligned}$$

The test is written more compactly after defining a few terms:

$$\begin{aligned}\eta &\triangleq \ln\frac{\pi_0}{\pi_1} - K\cdot\ln\frac{|\mathbf{R}_0|}{|\mathbf{R}_1|} && (3.1.10)\\ \mathbf{Q} &\triangleq \mathbf{R}_0^{-1}-\mathbf{R}_1^{-1} && (3.1.11)\\ f(\widehat{\mathbf{R}}) &\triangleq K\cdot\mathrm{tr}\left[\mathbf{Q}\widehat{\mathbf{R}}\right] && (3.1.12)\\ f(\widehat{\mathbf{R}}) &\underset{\text{`}H_0\text{'}}{\overset{\text{`}H_1\text{'}}{\gtrless}} \eta. && (3.1.13)\end{aligned}$$

In this form, the scalar function $f(\widehat{\mathbf{R}})$ is the test statistic and η is the threshold.

The second step in the derivation finds the characteristic function of the test statistic under each hypothesis. The characteristic function, $\phi_i(\cdot)$, of the test statistic under hypothesis H_i is given by the Fourier transform,

$$\begin{aligned}\phi_i(\omega) &= \mathrm{E}\left\{e^{-j\omega f(\widehat{\mathbf{R}})} \mid H_i\right\} \\ &= \int |\pi\mathbf{R}_i|^{-K}\cdot\exp\left\{-K\mathrm{tr}\left[\mathbf{R}_i^{-1}\widehat{\mathbf{R}}\right]\right\}\cdot\exp\left\{-j\omega\, K\mathrm{tr}\left[\mathbf{Q}\widehat{\mathbf{R}}\right]\right\} d\mathbf{X} \\ &= \int |\pi\mathbf{R}_i|^{-K}\cdot\exp\left\{-K\mathrm{tr}\left[\left(\mathbf{R}_i^{-1}+j\omega\mathbf{Q}\right)\widehat{\mathbf{R}}\right]\right\} d\mathbf{X}. \qquad (3.1.14)\end{aligned}$$

To find a closed form for this integral, the exponential involving $\widehat{\mathbf{R}}$ is converted into the form of a complex Gaussian density (Equation 3.1.6). Defining the covariance matrix

$$\boldsymbol{\Sigma} \triangleq \left(\mathbf{R}_i^{-1} + j\omega\mathbf{Q}\right)^{-1}, \tag{3.1.15}$$

the determinant becomes

$$\begin{aligned} |\pi\mathbf{R}_i| &= \left|\pi\left(\mathbf{I} + j\omega\mathbf{R}_i\mathbf{Q}\right)\left(\mathbf{I} + j\omega\mathbf{R}_i\mathbf{Q}\right)^{-1}\mathbf{R}_i\right| \\ &= \left|\pi\left(\mathbf{I} + j\omega\mathbf{R}_i\mathbf{Q}\right)\boldsymbol{\Sigma}\right| \\ &= \left|\mathbf{I} + j\omega\mathbf{R}_i\mathbf{Q}\right| \cdot |\pi\boldsymbol{\Sigma}|. \end{aligned} \tag{3.1.16}$$

Incorporating this result into the integral, Equation 3.1.14, obtains a closed form for the characteristic function:

$$\begin{aligned} \phi_i(\omega) &= \int \left|\mathbf{I} + j\omega\mathbf{R}_i\mathbf{Q}\right|^{-K} \cdot |\pi\boldsymbol{\Sigma}|^{-K} \exp\left\{-K\mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}\widehat{\mathbf{R}}\right]\right\} d\mathbf{X} \\ &= \left|\mathbf{I} + j\omega\mathbf{R}_i\mathbf{Q}\right|^{-K} \cdot \int |\pi\boldsymbol{\Sigma}|^{-K} \exp\left\{-K\mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}\widehat{\mathbf{R}}\right]\right\} d\mathbf{X} \\ &= \left|\mathbf{I} + j\omega\mathbf{R}_i\mathbf{Q}\right|^{-K}. \end{aligned} \tag{3.1.17}$$

The characteristic function is now in closed form, but its use is still limited because each evaluation involves the determinant of a potentially large matrix. Thankfully, evaluating the determinant under both hypotheses is simplified through the use of a generalized eigenvalue decomposition. Writing out the characteristic functions under both hypotheses gives

$$\begin{aligned} \phi_1(\omega) &= \left|\mathbf{I} + j\omega\mathbf{R}_1(\mathbf{R}_0^{-1} - \mathbf{R}_1^{-1})\right|^{-K} \\ &= \left|\mathbf{I} + j\omega(\mathbf{R}_1\mathbf{R}_0^{-1} - \mathbf{I})\right|^{-K} \end{aligned} \tag{3.1.18}$$

$$\begin{aligned} \phi_0(\omega) &= \left|\mathbf{I} + j\omega\mathbf{R}_0(\mathbf{R}_0^{-1} - \mathbf{R}_1^{-1})\right|^{-K} \\ &= \left|\mathbf{I} + j\omega(\mathbf{I} - \mathbf{R}_0\mathbf{R}_1^{-1})\right|^{-K} \\ &= \left|\mathbf{I} + j\omega[\mathbf{I} - (\mathbf{R}_1\mathbf{R}_0^{-1})^{-1}]\right|^{-K}. \end{aligned} \tag{3.1.19}$$

Recalling that the determinant of a matrix is the product of its eigenvalues, both characteristic functions are expressible in terms of λ_n, the eigenvalues of $\mathbf{R}_1\mathbf{R}_0^{-1}$:

$$\phi_1(\omega) = \left[\prod_n 1 + j\omega(\lambda_n - 1)\right]^{-K} \tag{3.1.20}$$

$$\phi_0(\omega) = \left[\prod_n 1 + j\omega(1 - 1/\lambda_n)\right]^{-K}. \tag{3.1.21}$$

These λ_n are also known as the "generalized eigenvalues" of the pair $(\mathbf{R}_1, \mathbf{R}_0)$. Equations 3.1.20 and 3.1.21 allow both characteristic functions to be evaluated easily from a shared eigenvalue decomposition.

The third step in deriving a bound on the probability of error is evaluating the cumulative distribution functions from their associated characteristic functions. The cumulative distribution function of the test statistic, $f(\widehat{\mathbf{R}})$, is related to the characteristic function through the integral

$$\mathrm{P}\left\{f(\widehat{\mathbf{R}}) < \eta | H_i\right\} = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty \frac{1}{\omega}\mathrm{Im}\left\{\phi_i(\omega)e^{j\omega\eta}\right\}\, d\omega \tag{3.1.22}$$

as given in [18]. Numerical evaluation of the integral in Equation 3.1.22 is complicated by the infinite upper limit. Equations 3.1.20 and 3.1.21, however, decrease rapidly with ω. With L non-unity eigenvalues, the integrand decreases asymptotically like $w^{-(LK+1)}$. Such a fast decay means that evaluating the integral in Equation 3.1.22 to high precision only requires a finite and reasonably small upper limit. Alternative approaches such as partial fraction expansion and saddle-point methods yield higher precision, especially in the tails of the distribution, but are unnecessary for this problem.[1]

The trivial last step in the derivation is expressing the probability of error in terms of the cumulative distribution functions of the test statistic. For this binary hypothesis test, the probability of error is determined by the weighted probability of error under each hypothesis. The total error is easily written in terms of the

[1] All results shown are from numerical integration verified by partial fraction decomposition.

cumulative distribution functions,

$$\begin{aligned} P_e &= \pi_0 \mathrm{P}\left\{\text{`}H_1\text{'}|H_0\right\} + \pi_1 \mathrm{P}\left\{\text{`}H_0\text{'}|H_1\right\} \\ &= \pi_0 \mathrm{P}\left\{f(\widehat{\mathbf{R}}) > \eta | H_0\right\} + \pi_1 \mathrm{P}\left\{f(\widehat{\mathbf{R}}) < \eta | H_1\right\} \\ &= \pi_0 \left[1 - \mathrm{P}\left\{f(\widehat{\mathbf{R}}) < \eta | H_0\right\}\right] + \pi_1 \mathrm{P}\left\{f(\widehat{\mathbf{R}}) < \eta | H_1\right\} \end{aligned} \tag{3.1.23}$$

where P_e is the probability of error. Summarizing the derivation, the minimum probability of error for the classification problem illustrated in Figure 3.1.1 is given by Equation 3.1.23. This equation is evaluated with the help of Equation 3.1.22 applied to Equations 3.1.20 and 3.1.21.

3.1.2 Analysis

The probability of error bound derived above involves many independent variables: arrival angle, signal-to-noise ratio (SNR), frequency, number of sensors, noise distribution, and the prior probabilities of each hypothesis. This subsection analyzes the bound under one insightful scenario with white noise and equal prior probabilities. In this subset of the parameter space, a simple formula illustrates the behavior of the bound and reveals important properties of linear vector-sensor arrays.

With white noise and equal prior probabilities, the remaining independent variables are arrival angle, SNR, frequency, and number of sensors. Holding the latter two variables constant at moderate values of $f = 5/7 f_d$ and $N = 10$ sensors, the bound is evaluated as a function of arrival angle, $\cos\phi$, and array SNR, $2N\sigma_s^2$. Contours of the resulting image, shown in Figure 3.1.2, reveal the ability of a linear vector-sensor array to resolve left/right ambiguities. The results are symmetric about broadside and endfire, so only one quadrant is shown. The most apparent feature of Figure 3.1.2 is that the performance of the vector-sensor array is effectively uniform over a large fraction of cosine-space. The SNR required to achieve a given probability of error changes by less than 3 dB for most of the space ($-0.9 \leq \cos\phi \leq 0$) but diverges at endfire ($-1 \leq \cos\phi \leq -0.9$). Another feature of Figure 3.1.2 is that a low probability

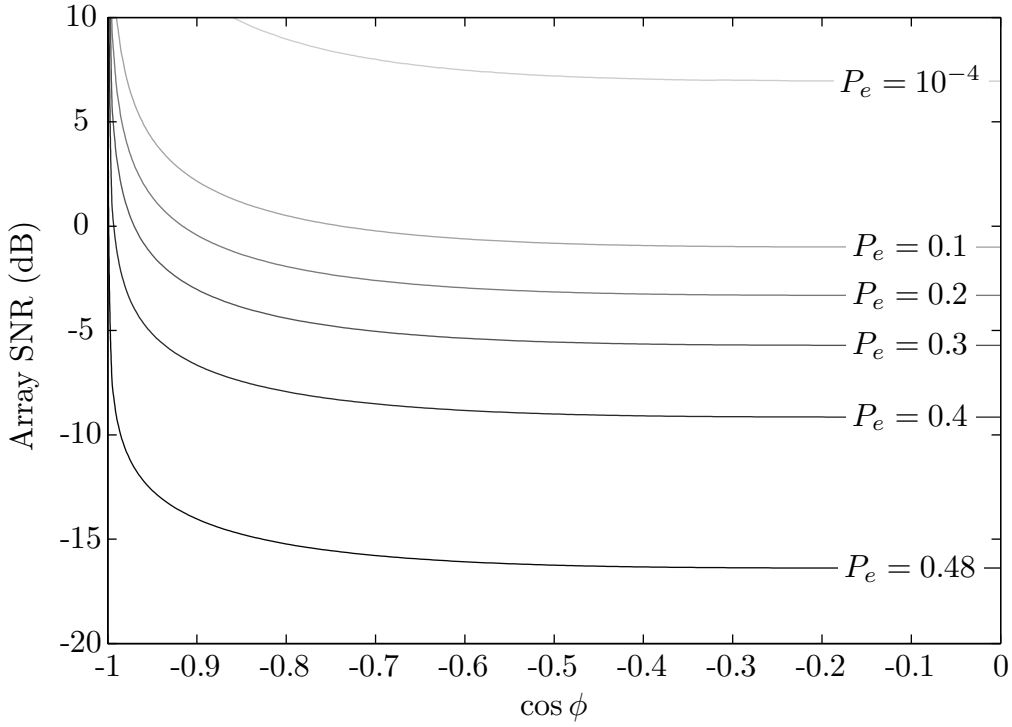

Figure 3.1.2: Probability of error contours: $f = \frac{5}{7} f_d$, $N = 10$

of error is achieved almost everywhere even with a moderate SNR. Specifically, less than 10% probability of error is achieved over 90% of cosine-space for a very weak, 3 dB source.

The first question raised by the good left/right resolution in Figure 3.1.2 is how the behavior scales with frequency or the number of sensors. Figure 3.1.3 displays the same probability of error contours for a different scenario at $f = 1/7 f_d$ and with only $N = 3$ sensors. One might expect the performance to suffer because of the low spatial resolution of such an array, but it appears unchanged from the original Figure 3.1.2. Intuitively, the ability of vector-sensors to resolve pressure ambiguities depends only on the directional sensors. The performance measured in the figures does not change because the response of the directional sensors does not change with frequency or number.

The concept of statistical divergence provides much more rigorous insight into the classification bound. Loosely speaking, divergences measure the dissimilarity between

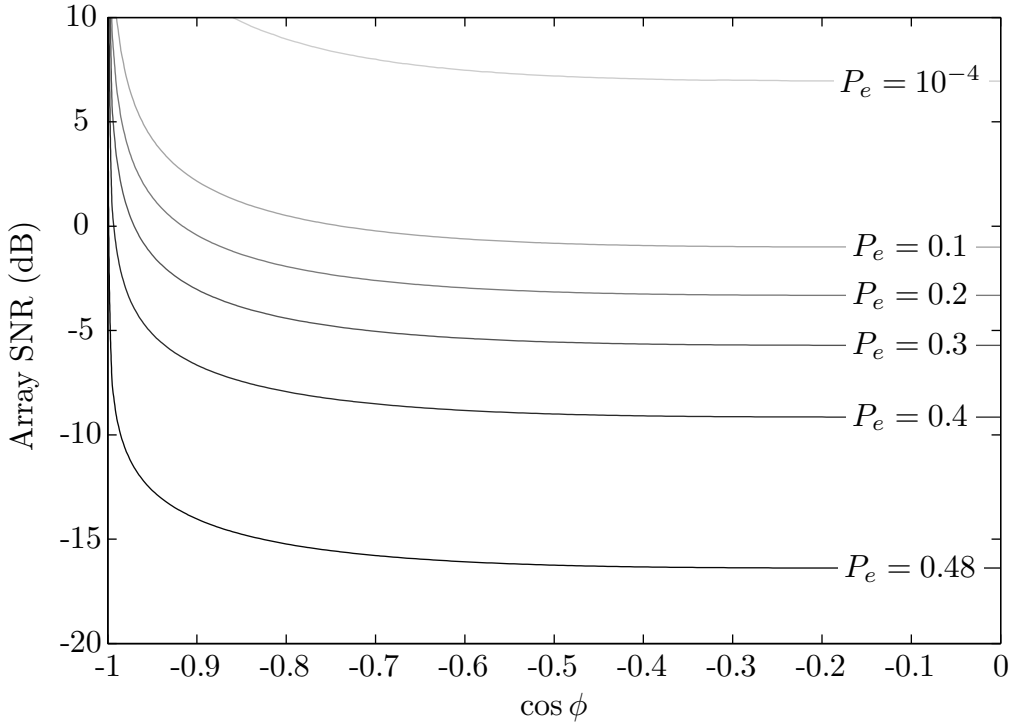

Figure 3.1.3: Probability of error contours: $f = \frac{1}{7} f_d$, $N = 3$

two distributions. In this case, divergences quantify the left/right information measured by the vector-sensor array. Intuitively, the more information provided by the array, the lower the probability of error. One divergence that often arises in information theory and binary hypothesis testing is the Kullback-Leibler (K-L) divergence [19, §12.7 - 12.9]. The K-L divergence between two probability distributions, $p_0(x)$ and $p_1(x)$, is

$$\mathrm{D}(p_0||p_1) \triangleq \int_{-\infty}^{+\infty} p_0(x) \ln \frac{p_0(x)}{p_1(x)}\, dx. \tag{3.1.24}$$

The K-L divergence takes a simple form for the zero-mean, complex Gaussian distributions considered here:

$$\mathrm{D}(p_0||p_1) = \frac{K}{2} \left[\ln \frac{|\mathbf{R}_1|}{|\mathbf{R}_0|} + \mathrm{tr}\left(\mathbf{R}_1^{-1}\mathbf{R}_0\right) - 4N \right]. \tag{3.1.25}$$

Recall that K is the number of independent observations and N is the number of vector-sensors. Under the weak condition that the noise is left/right symmetric (see

Appendix A.1), the K-L divergence simplifies to

$$\mathrm{D}(p_0||p_1) = \frac{K}{2}\left[\mathrm{tr}\left(\mathbf{R}_1^{-1}\mathbf{R}_0\right) - 4N\right] \tag{3.1.26}$$

because $|\mathbf{R}_1| = |\mathbf{R}_0|$. In this case, the K-L divergence is symmetric with respect to its arguments, $\mathrm{D}(p_0||p_1) = \mathrm{D}(p_1||p_0)$, and proportional to the J-divergence used in [17]. For the case of white noise, $\mathbf{R}_i = \mathbf{I} + \sigma^2\mathbf{v}_i\mathbf{v}_i^H$. Applying the matrix inversion lemma to $\mathbf{R}_1^{-1}$ yields

$$\begin{aligned}
\mathrm{D}(p_0||p_1) &= \frac{K}{2}\left\{\mathrm{tr}\left[\left(\mathbf{I} - \frac{\mathbf{v}_1\mathbf{v}_1^H}{\frac{1}{\sigma^2} + \mathbf{v}_1^H\mathbf{v}_1}\right)\left(\mathbf{I} + \sigma^2\mathbf{v}_0\mathbf{v}_0^H\right)\right] - 4N\right\} \\
&= \frac{K}{2}\left\{\mathrm{tr}\left[\mathbf{I} + \sigma^2\mathbf{v}_0\mathbf{v}_0^H - \frac{\mathbf{v}_1\mathbf{v}_1^H}{\frac{1}{\sigma^2} + \mathbf{v}_1^H\mathbf{v}_1} - \frac{\sigma^2\mathbf{v}_1\mathbf{v}_1^H\mathbf{v}_0\mathbf{v}_0^H}{\frac{1}{\sigma^2} + \mathbf{v}_1^H\mathbf{v}_1}\right] - 4N\right\} \\
&= \frac{K}{2}\left\{4N + 2N\sigma^2 - \frac{2N}{\frac{1}{\sigma^2} + 2N} - \frac{\sigma^2}{\frac{1}{\sigma^2} + 2N}|\mathbf{v}_1^H\mathbf{v}_0|^2 - 4N\right\}.
\end{aligned} \tag{3.1.27}$$

Recalling the work in [1, §2.2.2],

$$\mathbf{v}_1^H\mathbf{v}_0 = 2N\cos^2\phi. \tag{3.1.28}$$

Substituting into Equation 3.1.27 and collecting terms yields the simple expression

$$\mathrm{D}(p_0||p_1) = \frac{K}{2}\frac{\gamma^2}{1+\gamma}\left(1 - \cos^4\phi\right) \tag{3.1.29}$$

where

$$\gamma \triangleq 2N\sigma^2 \tag{3.1.30}$$

is the array SNR. The K-L divergence is a function of only cosine angle, array SNR, and number of observations; it does not depend on frequency or number of sensors. Just as each choice of independent variables has an associated probability of error, each also has an associated divergence. Contours of the divergence, analogous to the probability of error contours plotted in Figures 3.1.2 and 3.1.3, are given by curves

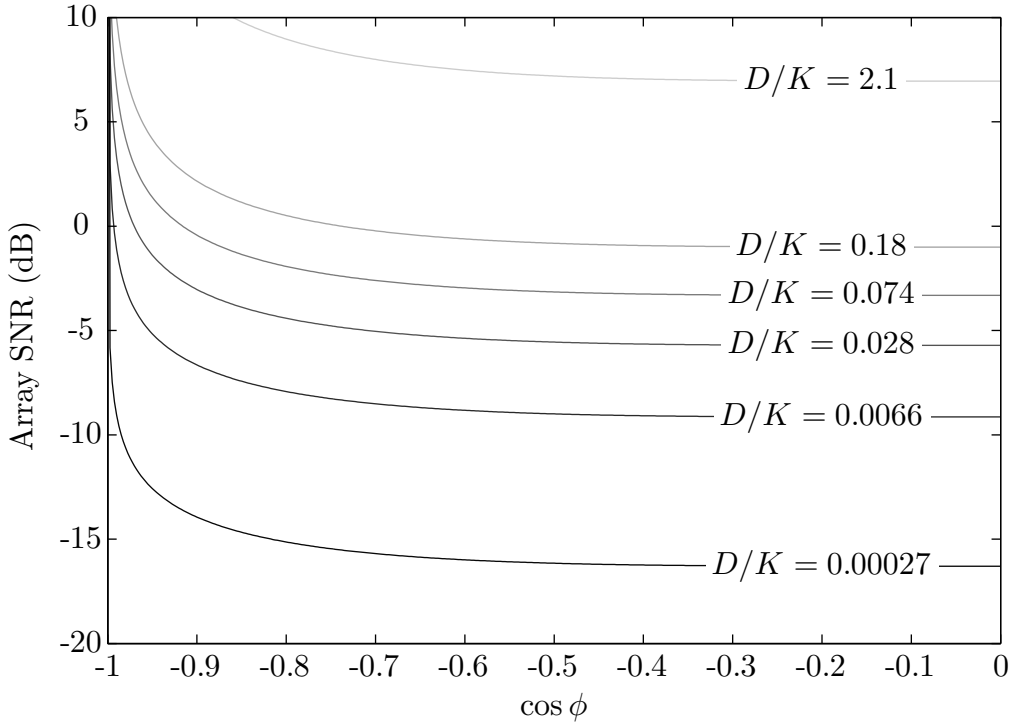

Figure 3.1.4: Divergence contours: $f = \frac{5}{7} f_d$, $N = 10$

of constant value, D:

$$D = \frac{K}{2} \frac{\gamma^2}{1+\gamma} \left(1 - \cos^4 \phi\right). \tag{3.1.31}$$

Parameterizing these curves as functions of $\cos\phi$ involves solving the quadratic Equation 3.1.31 for a positive γ to get

$$\gamma = \frac{1}{1 - \cos^4 \phi} \left[\frac{D}{K} + \sqrt{\left(\frac{D}{K}\right)^2 + 2\frac{D}{K}\left(1 - \cos^4 \phi\right)} \right]. \tag{3.1.32}$$

A carefully chosen set of these divergence contours are shown in Figure 3.1.4. When the number of observations is held constant, *contours of equal divergence correspond exactly to contours of equal error probability* in Figures 3.1.2 and 3.1.3. A proof of this conjecture is given in Appendix A.1. Although Equation 3.1.32 has a much simpler form than the probability of error, it still captures the important aspects of left/right resolution with vector-sensor arrays. Namely, it formally proves that the left/right resolution is independent of both frequency and number of sensors and is uniform

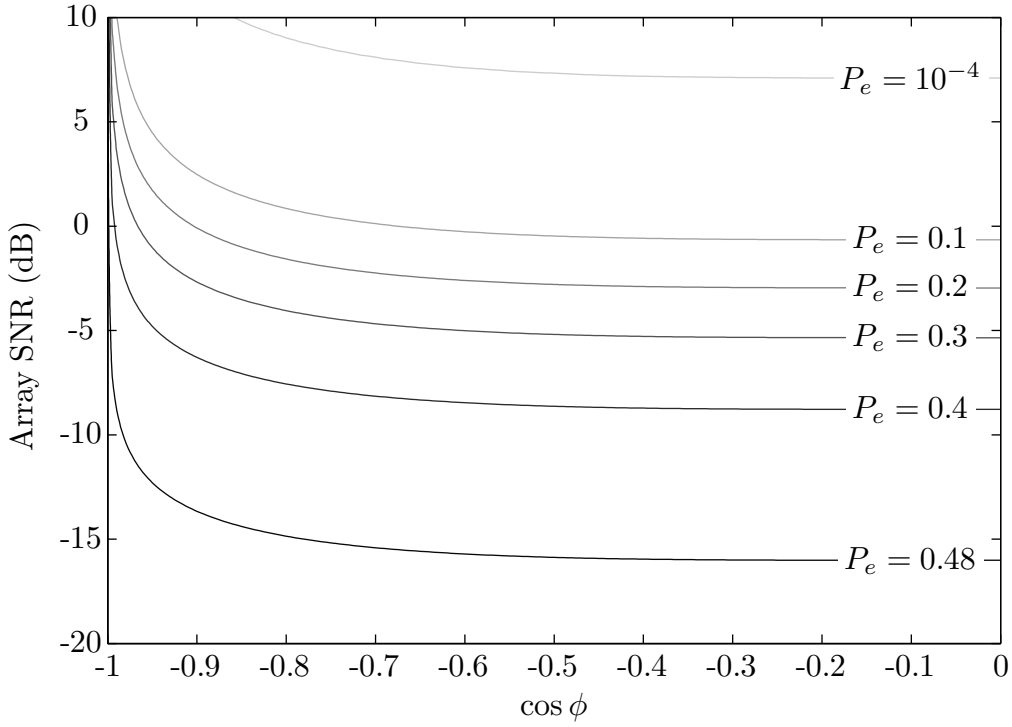

Figure 3.1.5: Probability of error contours: $f = \frac{5}{7} f_d$, $N = 10$, $\Delta \approx 0.05$

over most of cosine space.

The results shown above indicate that linear vector-sensor arrays are theoretically effective at determining left/right. However, to demonstrate that vector-sensor performance is robust, *i.e.* not reliant on point nulls or "supergain", the analysis is repeated for spatially spread sources. Recalling the discussion of spatial spreading in Section 2.4, a covariance matrix taper is used to approximate a uniformly distributed source. Returning to the $N = 10$ element array at $f = 5/7 f_d$ from Figure 3.1.2, the same source is spread over 1/6 of a beamwidth (≈ 0.05 in cosine-space) to obtain Figure 3.1.5. Comparing the two figures reveals that distributing the source has a minor effect on the left/right performance. Intuitively, the null that allows for left/right performance is determined by the directional sensors and is already very wide (see [1, Ch. 2]); therefore, it is relatively robust. The results in Figure 3.1.5 suggest that vector-sensor left/right performance is robust in theory.

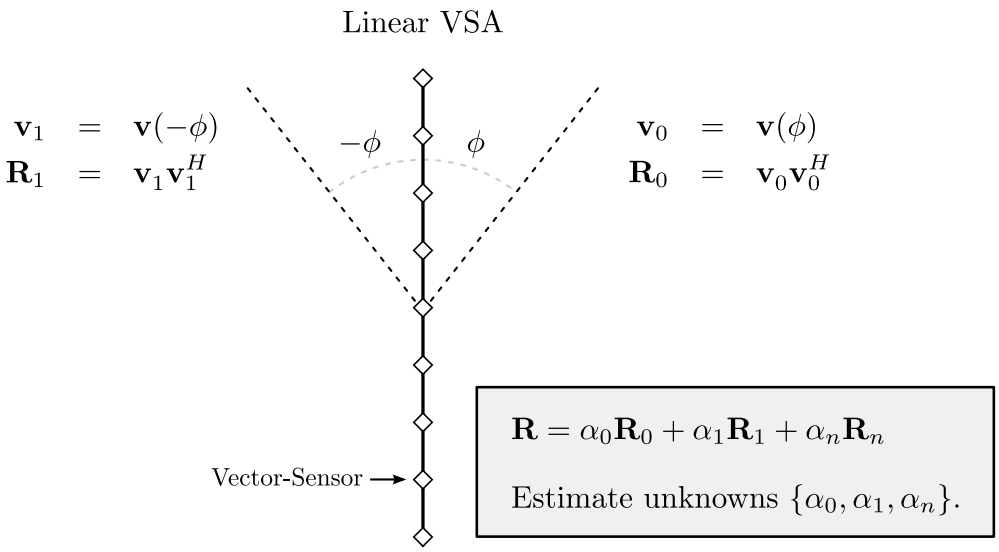

Figure 3.2.1: Example right/left power estimation scenario

3.2 Novel Estimation-Based Bound

Although the classification bound derived in the Section 3.1 provides insight into the left/right information measured by a linear vector-sensor array, it relies on the unrealistic assumption that the source power is known. A more realistic passive sonar scenario involves power estimation as illustrated in Figure 3.2.1. In this scenario, the objective is estimating the powers of sources on either side of the array and the power of the background noise. Both sources yield identical pressure measurements, so any ability to resolve differences in power arises from the directional, vector-sensor elements. Assuming zero-mean complex Gaussian distributions, the unknown, deterministic parameters are the three powers $\{\alpha_0, \alpha_1, \alpha_n\}$; the known parameters are the azimuth angles $\pm\phi$ and noise covariance $\mathbf{R}_n$. The derivation does not restrict the form of the noise covariance, but the analysis focuses on white noise. As with the previous problem, the K array measurements are summarized by the sample covariance matrix $\widehat{\mathbf{R}}$.

The estimation problem in Figure 3.2.1 is closely related to typical array processing. Passive sonar systems often estimate power at many hypothesized directions and frequencies, displaying these power estimates to trained technicians. The power

distribution across angle and frequency is essentially a two-dimensional spectrogram, so the results here are directly related to spectrogram estimation and periodograms (see [5, 20]). Many aspects of this problem also relate to the rejection of interference or jamming.

For the estimation problem in Figure 3.2.1, a number of performance bounds exist [21]. They include the Chapman-Robbins or Barankin bound, Bhattacharyya bound, and Cramér-Rao bound. For random, unknown parameters, others bounds exist which include the Weiss-Weinstein, Ziv-Zakai, and Bayesian Cramér-Rao bounds. The Cramér-Rao bound is chosen here because the goal is providing insight into a novel problem, not obtaining the tightest and most complex bound. The Cramér-Rao bound in this section shares only its name with those in [1, 2, 4, 6]; the underlying problems, derivations, and results are fundamentally different. A good introduction to the Cramér-Rao bound and its application to standard measures of array performance is [5].

The Cramér-Rao bound for a parameter vector $\boldsymbol{\theta}$ states that the error covariance of any unbiased estimate $\widehat{\boldsymbol{\theta}}$ obeys

$$\mathrm{E}\{(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\} \succeq \mathbf{J}(\boldsymbol{\theta})^{-1}, \tag{3.2.1}$$

where $\mathbf{J}(\boldsymbol{\theta})$ is the "Fisher information matrix." Each entry of the Fisher information matrix is given by

$$[\mathbf{J}(\boldsymbol{\theta})]_{ij} = -\mathrm{E}\left\{ \frac{\partial^2}{\partial\theta_i \partial\theta_j} \ln \Lambda(\widehat{\mathbf{R}}) \right\}. \tag{3.2.2}$$

The matrix inequality above has several equivalent and intuitive interpretations

$$\mathbf{A} \succeq \mathbf{B} \quad \Leftrightarrow \quad \mathbf{A} - \mathbf{B} \text{ is positive semidefinite} \tag{3.2.3}$$

$$\Leftrightarrow \quad \mathbf{x}^T\mathbf{A}\mathbf{x} \geq \mathbf{x}^T\mathbf{B}\mathbf{x} \quad \forall \mathbf{x}. \tag{3.2.4}$$

A consequence of Equation 3.2.1 is that the error variance of a scalar estimate $\widehat{\theta}_i$ is

bounded by a diagonal element of the same inverse,

$$\mathrm{E}\{(\widehat{\theta_i} - \theta_i)^2\} \geq \left[\mathbf{J}(\boldsymbol{\theta})^{-1}\right]_{ii}. \tag{3.2.5}$$

For the estimation problem in this section, the vector of unknown parameters consists of source and noise powers,

$$\boldsymbol{\theta} \triangleq [\alpha_0 \quad \alpha_1 \quad \alpha_n]^T. \tag{3.2.6}$$

Using the Cramér-Rao bound, this section determines how much a signal on one side of a vector-sensor array interferes with power estimation on the opposing side.

3.2.1 Derivation

This section derives a Cramér-Rao lower bound for the power estimation problem outlined above. The derivation is carried out for an arbitrary noise covariance and is much more compact than that of the classification bound in Section 3.1. This brevity is partly because the resulting expression is not easily simplified even in special cases.

The Fisher information matrix is first derived element-by-element. The log-likelihood function for zero-mean, complex Gaussian measurements is given in Equation 3.1.7. Entries of the associated Fisher information matrix, $\mathbf{J}(\boldsymbol{\theta})$, have a very simple form,

$$[\mathbf{J}(\boldsymbol{\theta})]_{ij} = K \mathrm{tr}\left(\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_i}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_j}\right). \tag{3.2.7}$$

Because each term is scaled by the number of observations, K, this derivation focuses on the single observation case without loss of generality. The covariance matrix, $\mathbf{R}$, is a function of the three unknown powers,

$$\begin{aligned} \mathbf{R} &= \alpha_0\mathbf{R}_0 + \alpha_1\mathbf{R}_1 + \alpha_n\mathbf{R}_n \\ &= \alpha_0\mathbf{v}_0\mathbf{v}_0^H + \alpha_1\mathbf{v}_1\mathbf{v}_1^H + \alpha_n\mathbf{R}_n. \end{aligned} \tag{3.2.8}$$

From this linear combination, the necessary partial derivatives for the three unknown

parameters are easy to compute

$$\frac{\partial \mathbf{R}}{\partial \alpha_0} = \mathbf{v}_0 \mathbf{v}_0^H \tag{3.2.9}$$

$$\frac{\partial \mathbf{R}}{\partial \alpha_1} = \mathbf{v}_1 \mathbf{v}_1^H \tag{3.2.10}$$

$$\frac{\partial \mathbf{R}}{\partial \alpha_n} = \mathbf{R}_n. \tag{3.2.11}$$

The $(1,2)$ term in the Fisher information matrix involves the two sources and is a convenient place to begin:

$$\begin{aligned} [\mathbf{J}(\boldsymbol{\theta})]_{12} &= \mathrm{tr}\left(\mathbf{R}^{-1}\mathbf{v}_0\mathbf{v}_0^H\mathbf{R}^{-1}\mathbf{v}_1\mathbf{v}_1^H\right) \\ &= \mathrm{tr}\left(\mathbf{v}_1^H\mathbf{R}^{-1}\mathbf{v}_0\mathbf{v}_0^H\mathbf{R}^{-1}\mathbf{v}_1\right) \\ &= \left|\mathbf{v}_0^H\mathbf{R}^{-1}\mathbf{v}_1\right|^2. \end{aligned} \tag{3.2.12}$$

The first step above (and many steps to come) uses the identity $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$. Equation 3.2.12 extends to any term involving only the two sources. Moving to the $(1,3)$ term involving one source and the noise,

$$\begin{aligned} [\mathbf{J}(\boldsymbol{\theta})]_{13} &= \mathrm{tr}\left(\mathbf{R}^{-1}\mathbf{v}_0\mathbf{v}_0^H\mathbf{R}^{-1}\mathbf{R}_n\right) \\ &= \mathbf{v}_0^H\mathbf{R}^{-1}\mathbf{R}_n\mathbf{R}^{-1}\mathbf{v}_0. \end{aligned} \tag{3.2.13}$$

Equation 3.2.13 also extends to the other source. The diagonal, $(3,3)$ term involves only the noise power:

$$[\mathbf{J}(\boldsymbol{\theta})]_{33} = \mathrm{tr}\left(\mathbf{R}^{-1}\mathbf{R}_n\mathbf{R}^{-1}\mathbf{R}_n\right). \tag{3.2.14}$$

Unlike the other terms, Equation 3.2.14 is not easily simplified by eliminating the matrix trace function.

Enough work exists now to write the full Fisher information matrix. The Fisher

information matrix for this three-parameter case is best viewed in block form,

$$\mathbf{J}(\boldsymbol{\theta}) = \left[\begin{array}{c|c} \mathbf{J}_{ss} & \mathbf{J}_{sn} \\ \hline \mathbf{J}_{sn}^T & J_{nn} \end{array}\right], \tag{3.2.15}$$

where $\mathbf{J}_{ss}$ is the 2×2 matrix involving only the two sources, $\mathbf{J}_{sn}$ is the 2×1 vector involving the sources and noise, and J_{nn} is a scalar involving only the noise. Condensing the results derived in the previous paragraph,

$$\mathbf{J}_{ss} = (\mathbf{V}^H \mathbf{R}^{-1} \mathbf{V}) \odot (\mathbf{V}^H \mathbf{R}^{-1} \mathbf{V})^* \tag{3.2.16}$$

$$\mathbf{J}_{sn} = \operatorname{diag}\left(\mathbf{V}^H \mathbf{R}^{-1} \mathbf{R}_n \mathbf{R}^{-1} \mathbf{V}\right) \tag{3.2.17}$$

$$J_{nn} = \operatorname{tr}\left(\mathbf{R}^{-1} \mathbf{R}_n \mathbf{R}^{-1} \mathbf{R}_n\right), \tag{3.2.18}$$

where the diag$(\cdot)$ function extracts the main diagonal of a matrix as a column vector, $\odot$ is the Hadamard or element-wise product, * denotes conjugation, and

$$\mathbf{V} \triangleq [\mathbf{v}_0 \quad \mathbf{v}_1]. \tag{3.2.19}$$

Each evaluation of the Fisher information matrix involves several matrix products and a matrix inverse. Although the inverse is computed efficiently via the matrix inversion lemma, the expressions are already in their most simplified analytical form.

3.2.2 Analysis

The derivation in the previous section is complete and compact, but it is not fully satisfying for two reasons. First, its interpretation is not immediately clear. Each choice of independent variables yields a matrix inequality that is not intuitive. Second, its dependencies are not obvious. It is difficult to discern from the expressions how the bound changes with frequency, number of sensors, *etc.* This section addresses both points, providing a useful visualization of the bound and an approximation that clarifies the dependencies.

The first analysis problem is interpreting the bound. A clear interpretation requires 1) reducing the large parameter space to a small region of interest, and 2) obtaining an intuitive metric from the covariance matrix inequality. The following paragraphs describe one interpretation of the Cramér-Rao bound and illustrate the bound with several examples.

The large parameter space makes it difficult to interpret the Cramér-Rao bound. However, examining a reduced parameter space delineates the a clear region of interest. Assuming other parameters are fixed, the space of all three unknown parameters $\{\alpha_0, \alpha_1, \alpha_n\}$ provides a useful context for interpreting the bound. Symmetry in the problem and practical considerations suggest there are only three distinct regions as determined by the strength of the sources relative to the noise. In the first region, both sources dominate the noise. This high-SNR region is uninteresting because 1) sources of interest are often weak for passive sonar and 2) simple algorithms exist that perform well in the absence of noise. In the second region, the noise dominates one or more source. This region is equally uninteresting; reliably estimating the power of such weak sources is extremely difficult. The third region includes most cases of interest and is characterized by noise power on the same order as at least one source. This region of interest is fully explored for the white noise case when $\alpha_n \approx 2N\alpha_0$ and α_1 is swept from $\alpha_1 \ll \alpha_0$ to $\alpha_1 \gg \alpha_0$. The factor of $2N$ accounts for the array gain. Under this scenario, the source α_0 is treated as a "target" and α_1 as a "jammer." The goal is resolving the true power of the target regardless of the jammer power. The entire region of interest consists of only three regimes: the no jammer regime ($\alpha_1 \ll \alpha_0$), the weak jammer regime ($\alpha_1 \approx \alpha_0$), and the strong jammer regime ($\alpha_1 \gg \alpha_0$).

Having reduced the parameter space to a reasonable size, the remaining difficulty is interpreting the covariance matrix inequality. Keeping with the "target and jammer" interpretation, the wanted parameter is the target power α_0; the other powers, $\{\alpha_1, \alpha_n\}$, are nuisance parameters. The error variance of the wanted parameter, α_0, is bounded below by the $(1,1)$ term of $\mathbf{J}(\boldsymbol{\theta})^{-1}$ (see Equation 3.2.5). A useful quantity

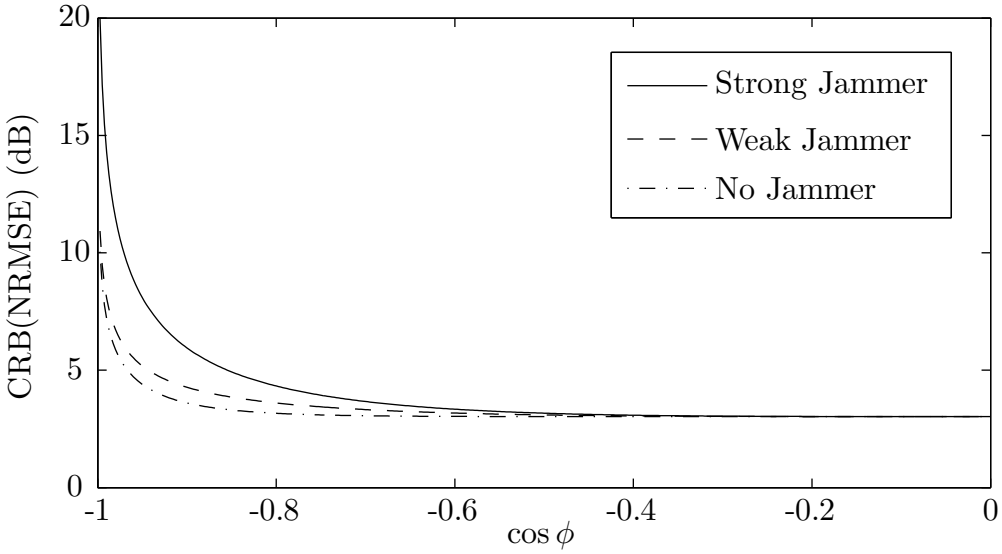

Figure 3.2.2: Left/right power estimation error: $f = \frac{5}{7} f_d$, $N = 10$

that summarizes this variance is the normalized root mean square error (NRMSE),

$$\text{NRMSE} \triangleq \frac{\sqrt{\text{var}(\widehat{\alpha}_0)}}{\alpha_0} \tag{3.2.20}$$

$$\text{CRB(NRMSE)} = \frac{1}{\alpha_0}\sqrt{[\mathbf{J}(\boldsymbol{\theta})^{-1}]_{11}} \tag{3.2.21}$$

$$\text{NRMSE} \geq \text{CRB(NRMSE)}, \tag{3.2.22}$$

where $\widehat{\alpha}_0$ is any unbiased estimate of α_0 and CRB$(\cdot)$ denotes the Cramér-Rao bound. Figure 3.2.2 plots curves of CRB(NRMSE) versus azimuth cosine for the three regimes. The standard $N = 10$ element vector-sensor array at $f = 5/7 f_d$ is used with $K = 1$, $\mathbf{R}_n = \mathbf{I}$, $\alpha_0 = 1/(2N)$, and $\alpha_n = 1$. A high CRB indicates that the jammer irrevocably corrupts estimates of the target power. The most interesting aspect of Figure 3.2.2 is that, as with the classification bound derived in Section 3.1, the predicted VSA performance is uniformly good over most of cosine space. Specifically, the CRB changes by less than 3 dB over 90% of space ($-0.9 \leq \cos\phi \leq 0$) but diverges at endfire ($-1 \leq \cos\phi \leq -0.9$). As with the classification bound in Section 3.1, good performance is predicted almost everywhere with even a weak source.

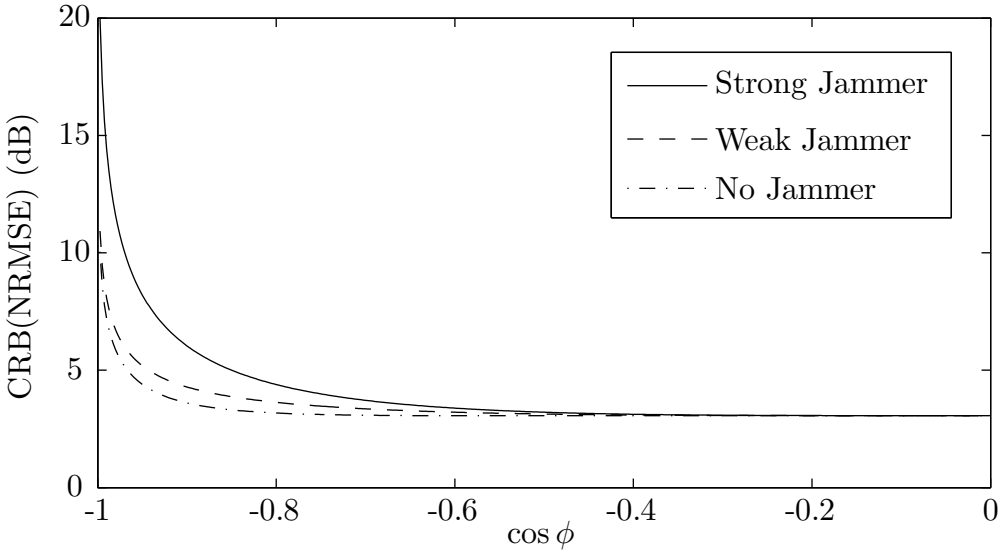

Figure 3.2.3: Left/right power estimation error: $f = \frac{1}{7} f_d$, $N = 3$

As with the classification bound, the results in Figure 3.2.2 do not change significantly with frequency or number of sensors. Figure 3.2.3 displays the same curves for an array with much lower resolution: $N = 3$ and $f = 1/7 f_d$. Comparing Figure 3.2.2 to Figure 3.2.3 reveals negligible differences, suggesting again that the left/right resolution is an inherent capability of the directional sensors and is almost unaffected by their number or spacing. Furthermore, the Cramér-Rao bound is proportional to $\sqrt{K}$, so changing the number of observations only shifts the curves in log-space and does not affect the conclusions.

The vector-sensor performance predicted by the bound also seems to be robust. As is done for the classification bound in Figure 3.1.5, uniform spatial spreading is introduced via a covariance matrix taper in Figure 3.2.4. Distributing the sources increases the bound only slightly and does not change the conclusion that vector-sensor performance is uniformly good over most of space. As with the classification bound, the wide null placed by the directional sensors appears to provide robust left/right discrimination without relying on "supergain." Introducing a covariance matrix taper requires modifying the derivation in Section 3.2.1. The bound is not re-derived here for brevity and because the terms do not simplify beyond Equation 3.2.7.

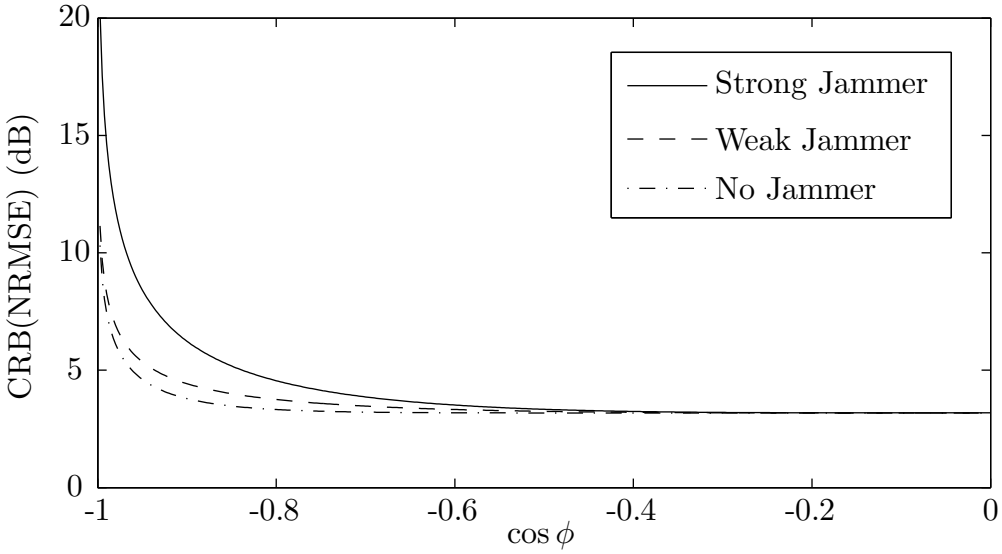

Figure 3.2.4: Left/right power estimation error: $f = \frac{5}{7} f_d$, $N = 10$, $\Delta \approx 0.05$

The second analysis problem is formalizing the dependencies of the Cramér-Rao bound. Figures 3.2.2 and 3.2.3 suggest that VSA performance does not depend on the number of sensors or the frequency. The following analysis proves this independence in the strong jammer regime, $\alpha_1 \to \infty$. The strong jammer regime is intuitively the most difficult, reflecting the "worst-case" VSA performance. In the above limit,

$$\mathbf{R}^{-1} \frac{\partial \mathbf{R}}{\partial \alpha_1} \to \mathbf{0}, \tag{3.2.23}$$

so the terms in the Fisher information matrix dealing with the jammer go to zero. Thus, the jammer can be treated as deterministic and removed from the matrix (see [5, Example 8.11.2]). The CRB(NRMSE) depends only on the $(1,1)$ term of $\mathbf{J}(\boldsymbol{\theta})^{-1}$,

$$\begin{aligned}
\left[\mathbf{J}(\boldsymbol{\theta})^{-1}\right]_{11} &= \left[\begin{bmatrix} \left(\mathbf{v}_0^H \mathbf{R}^{-1} \mathbf{v}_0\right)^2 & \mathbf{v}_0^H \mathbf{R}^{-2} \mathbf{v}_0 \\ \mathbf{v}_0^H \mathbf{R}^{-2} \mathbf{v}_0 & \operatorname{tr}\left(\mathbf{R}^{-2}\right) \end{bmatrix}^{-1} \right]_{11} \\
&= \frac{\operatorname{tr}\left(\mathbf{R}^{-2}\right)}{\left(\mathbf{v}_0^H \mathbf{R}^{-1} \mathbf{v}_0\right)^2 \operatorname{tr}\left(\mathbf{R}^{-2}\right) - \left(\mathbf{v}_0^H \mathbf{R}^{-2} \mathbf{v}_0\right)^2} \\
&= \left[\left(\mathbf{v}_0^H \mathbf{R}^{-1} \mathbf{v}_0\right)^2 - \frac{\left(\mathbf{v}_0^H \mathbf{R}^{-2} \mathbf{v}_0\right)^2}{\operatorname{tr}\left(\mathbf{R}^{-2}\right)} \right]^{-1},
\end{aligned} \tag{3.2.24}$$

where $K = 1$ is assumed. Applying the matrix inversion lemma to $\mathbf{R}^{-1}$ simplifies Equation 3.2.24. The white noise scenario with $\mathbf{R}_n = \mathbf{I}$, $\alpha_0 = 1/(2N)$, and $\alpha_n = 1$ yields an intuitive and simple result. Writing the covariance matrix,

$$\mathbf{R} = \mathbf{I} + [\mathbf{v}_0 \quad \mathbf{v}_1] \begin{bmatrix} \alpha_0 & 0 \\ 0 & \alpha_1 \end{bmatrix} [\mathbf{v}_0 \quad \mathbf{v}_1]^H, \tag{3.2.25}$$

the inverse is

$$\mathbf{R}^{-1} = \mathbf{I} - [\mathbf{v}_0 \quad \mathbf{v}_1] \left(\begin{bmatrix} 1/\alpha_0 & 0 \\ 0 & 1/\alpha_1 \end{bmatrix} + [\mathbf{v}_0 \quad \mathbf{v}_1]^H [\mathbf{v}_0 \quad \mathbf{v}_1] \right)^{-1} [\mathbf{v}_0 \quad \mathbf{v}_1]^H. \tag{3.2.26}$$

Recall that the inner product $\mathbf{v}_1^H \mathbf{v}_0$ is given in Equation 3.1.28. Taking the limit $\alpha_1 \to \infty$, substituting for α_0, and using some algebra gives

$$\mathbf{R}^{-1} = \mathbf{I} - [\mathbf{v}_0 \quad \mathbf{v}_1] \left(2N \begin{bmatrix} 2 & \cos^2\phi \\ \cos^2\phi & 1 \end{bmatrix} \right)^{-1} [\mathbf{v}_0 \quad \mathbf{v}_1]^H. \tag{3.2.27}$$

Applying this inverse to the target replica vector yields a simple expression for $\mathbf{R}^{-1}\mathbf{v}_0$:

$$\begin{aligned}
\mathbf{R}^{-1}\mathbf{v}_0 &= \mathbf{v}_0 - [\mathbf{v}_0 \quad \mathbf{v}_1] \left(2N \begin{bmatrix} 2 & \cos^2\phi \\ \cos^2\phi & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2N \\ 2N\cos^2\phi \end{bmatrix} \\
&= \mathbf{v}_0 - [\mathbf{v}_0 \quad \mathbf{v}_1] \begin{bmatrix} 2 & \cos^2\phi \\ \cos^2\phi & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \cos^2\phi \end{bmatrix} \\
&= \mathbf{v}_0 - \frac{[\mathbf{v}_0 \quad \mathbf{v}_1]}{2 - \cos^4\phi} \begin{bmatrix} 1 & -\cos^2\phi \\ -\cos^2\phi & 2 \end{bmatrix} \begin{bmatrix} 1 \\ \cos^2\phi \end{bmatrix} \\
&= \mathbf{v}_0 - \frac{[\mathbf{v}_0 \quad \mathbf{v}_1]}{2 - \cos^4\phi} \begin{bmatrix} 1 - \cos^4\phi \\ \cos^2\phi \end{bmatrix} \\
&= \frac{\mathbf{v}_0 - \mathbf{v}_1 \cos^2\phi}{2 - \cos^4\phi}.
\end{aligned} \tag{3.2.28}$$

This expression yields two of the three terms required in Equation 3.2.24:

$$\begin{aligned}
\mathbf{v}_0^H \mathbf{R}^{-1} \mathbf{v}_0 &= \frac{2N - 2N\cos^4\phi}{2 - \cos^4\phi} \\
&= 2N \frac{1 - \cos^4\phi}{2 - \cos^4\phi} \qquad (3.2.29) \\
\mathbf{v}_0^H \mathbf{R}^{-2} \mathbf{v}_0 &= \frac{2N - 4N\cos^4\phi + 2N\cos^4\phi}{(2 - \cos^4\phi)^2} \\
&= 2N \frac{1 - \cos^4\phi}{(2 - \cos^4\phi)^2}. \qquad (3.2.30)
\end{aligned}$$

The remaining term, the trace of $\mathbf{R}^{-2}$, comes by inspection of the eigenvalues of $\mathbf{R}^{-1}$. In the above limit, $\mathbf{R}^{-1}$ has one zero eigenvalue and one non-unity eigenvalue; the remaining $4N - 2$ eigenvalues are unity. The non-unity eigenvalue arises from the noise combined with the component of the target orthogonal to $\mathbf{v}_1$ and is thus $[1 + (1 - \cos^4\phi)]^{-1}$. The trace is therefore

$$\begin{aligned}
\mathrm{tr}\left(\mathbf{R}^{-2}\right) &= (4N - 2) + (2 - \cos^4\phi)^{-2} \\
&= \frac{(4N - 2)(2 - \cos^4\phi)^2 + 1}{(2 - \cos^4\phi)^2}. \qquad (3.2.31)
\end{aligned}$$

Substituting terms into Equation 3.2.24 gives

$$\begin{aligned}
\left[\mathbf{J}(\boldsymbol{\theta})^{-1}\right]_{11} &= \left[(2N)^2 \frac{(1 - \cos^4\phi)^2}{(2 - \cos^4\phi)^2} - \frac{(2N)^2 \frac{(1-\cos^4\phi)^2}{(2-\cos^4\phi)^4}}{\frac{(4N-2)(2-\cos^4\phi)^2+1}{(2-\cos^4\phi)^2}}\right]^{-1} \\
&= \frac{(2 - \cos^4\phi)^2}{(2N)^2(1 - \cos^4\phi)^2 \left[1 - \frac{1}{(4N-2)(2-\cos^4\phi)^2+1}\right]}. \qquad (3.2.32)
\end{aligned}$$

Noticing that

$$1 - \frac{1}{(4N - 2)(2 - \cos^4\phi)^2 + 1} \approx 1 \qquad (3.2.33)$$

for even a small number of sensors, Equation 3.2.21 is well-approximated by

$$\begin{aligned} \text{CRB(NRMSE)} &\approx \frac{2-\cos^4\phi}{1-\cos^4\phi} \\ &= 1+\frac{1}{1-\cos^4\phi}. \end{aligned} \tag{3.2.34}$$

Furthermore, the above approximation remains a valid lower bound because it is always less than the CRB. Equation 3.2.34 is a key result, as it verifies analytically that in the strong jammer regime, the normalized Cramér-Rao bound does not depend on frequency and only very weakly on the number of sensors. It predicts with a high degree of accuracy the "Strong Jammer" curves in Figures 3.2.2 and 3.2.3.

3.3 Conclusions and Intuition

This chapter isolates, measures, and bounds the VSA ability to discriminate signals that are ambiguous with a PSA. Section 3.1 explores the problem of binary classification when the signal power is known; Section 3.2 explores the problem of joint estimation when the powers are unknown. Although the two scenarios are distinct, they lead to similar conclusions and provide intuition about VSA performance.

The results prove that theoretical VSA ambiguity resolution is 1) good everywhere except array endfire and 2) independent of the number of sensors or the analysis frequency. Generally, bounds only prove that good performance *may be* achievable. However, the bounds developed in this chapter are tight. The classification-based bound in Section 3.1 is constructive, *i.e.* its derivation involves a likelihood ratio test which achieves the minimum probability of error. The estimation-based bound in Section 3.2 is asymptotically tight. Maximum likelihood estimators for this type of problem are asymptotically efficient ([5, §8.5]), approaching the Cramér-Rao bound as the number of observations approaches infinity ($K \to \infty$). Because both bounds are tight, they prove that good performance *is* achievable under the scenarios described in Sections 3.1 and 3.2.

Shared elements of the analysis also provide intuition about linear VSA capabilities. Equations 3.1.29 and 3.2.34 reveal that both performance bounds are a function of the same quantity, the left/right rejection of a single vector-sensor:

$$\zeta(\phi) \triangleq \cos^4 \phi. \tag{3.3.1}$$

Ambiguity resolution stems from the directional ability of each vector-sensor, so *the behavior of one sensor provides intuition about the behavior of an array.* Just as with a single vector-sensor, the ability of a VSA to resolve pressure ambiguities

- Does not change with the analysis frequency
- Is good except near endfire (where $\zeta(\phi) \approx 1$)
- Is robust (because left/right nulls are wide).

According to the same principle, the number of vector-sensors only affects left/right performance through the array gain.

Chapter 4

Fixed Weight Design for Uniform Linear Vector-Sensor Arrays

Chapters 2 and 3 explore the properties and performance limits of vector-sensor arrays. Chapter 3 predicts that vector-sensor ambiguity resolution is very good almost everywhere in theory. The question remains, however: is this performance achievable in practice?

Building with the tools outlined in Chapter 2, this chapter designs fixed weights for linear vector-sensor arrays. Designing "good" fixed weights for such arrays means balancing the competing objectives of low sensitivity to modeling errors, a narrow beamwidth, and low sidelobe levels. After surveying and categorizing existing techniques, this chapter proposes the use of offline convex optimization for beampattern design. The techniques described in Section 4.3.3 design robust, fixed weights for efficient non-adaptive processing. In many scenarios, these weights achieve good performance like that predicted in Chapter 3. Each modified beampattern in the chapter is compared to the "original," or conventional, beampattern.

4.1 Designs Using the Array Modulation Theorem (AMT)

Good linear PSA beampatterns are easily achieved for every "look direction" by applying a well-designed spatial taper. Section 2.3.2 shows that this is not possible with linear vector-sensor arrays, but a related technique illustrates the use of spatial tapers in VSA processing. Imposing structure on the weights enables VSA beampattern design using the array modulation theorem. Structuring the weights yields an intuitive but less flexible technique.

4.1.1 The Array Modulation Theorem and VSAs

The array modulation theorem provides a useful visualization of spatial tapering applied to vector-sensor arrays. It introduces "pattern multiplication," which simplifies certain designs by factoring the beampattern into two intuitive terms. This subsection introduces the array modulation theorem and illustrates its application to linear vector-sensor arrays.

The array modulation theorem states that the beampattern for an array of identical, directional sensors is the beampattern assuming omnidirectional sensors modulated by the response of the directional sensor. Because the sensor response is the same for each element, it factors out of the beampattern summation. Proof of this factorization is provided in [5, §2.8] for the general case and in [1, §2.1] for vector-sensor arrays. A key restriction of the array modulation theorem is that the responses of each sensor element must be identical.

Applying the array modulation theorem to vector-sensor arrays is not obvious because the VSA contains four types of sensors with different responses. To apply the theorem with "look direction" ϕ_0, consider weighting the n^{th} vector-sensor by

$$\mathbf{w}_n = \mathbf{p}\ t_n\ e^{jk_0 d_n \cos\phi_0}, \tag{4.1.1}$$

where $\mathbf{p}$ forms the same linear combination from every vector-sensor and t_n applies

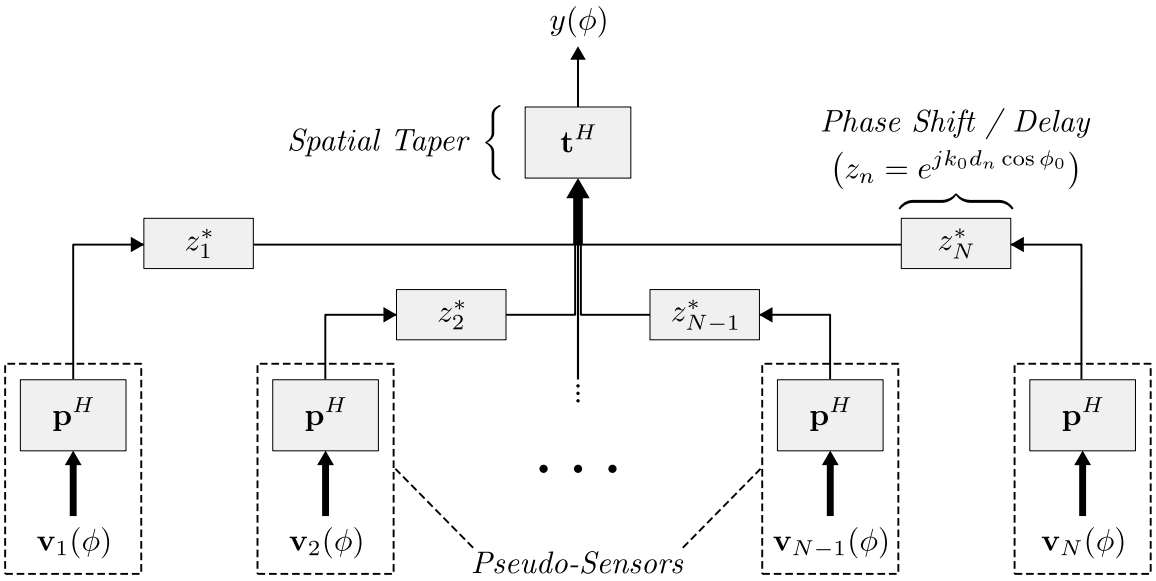

Figure 4.1.1: Weighting scheme for which the array modulation theorem applies

a unique scale factor to each. Figure 4.1.1 illustrates such a structured weighting scheme. The 4×1 vector $\mathbf{p}$ applies the same weighting to each vector-sensor, forming N identical "pseudo-sensors." The $N \times 1$ spatial taper $\mathbf{t} = [t_1 \; t_2 \; \cdots \; t_N]^T$ forms a beampattern from these pseudo-sensors. Because the pseudo-sensors are identical, the array modulation theorem implies that *the VSA beampattern is the beampattern of the taper modulated by the response of the pseudo-sensor.*

4.1.2 AMT Beampattern Design

The array modulation theorem enables the design of interesting beampatterns which combine spatial tapering with pseudo-sensor nulling. Some of these beampatterns are explored in [3, 4]. The remainder of this section provides example beampatterns which use the pseudo-sensor response to null the pressure-sensor ambiguity and the spatial taper to control sidelobes. For comparison, all examples are with $\phi_0 = -\pi/4$, $N = 10$ elements, and $f = 5/7 f_d$.

A naive initial approach places the minimum of the pseudo-sensor response at zero in the direction of the pressure-sensor ambiguity, effectively fixing the pseudo-sensor beampattern and its derivative at a point. Assuming a three-dimensional vector-

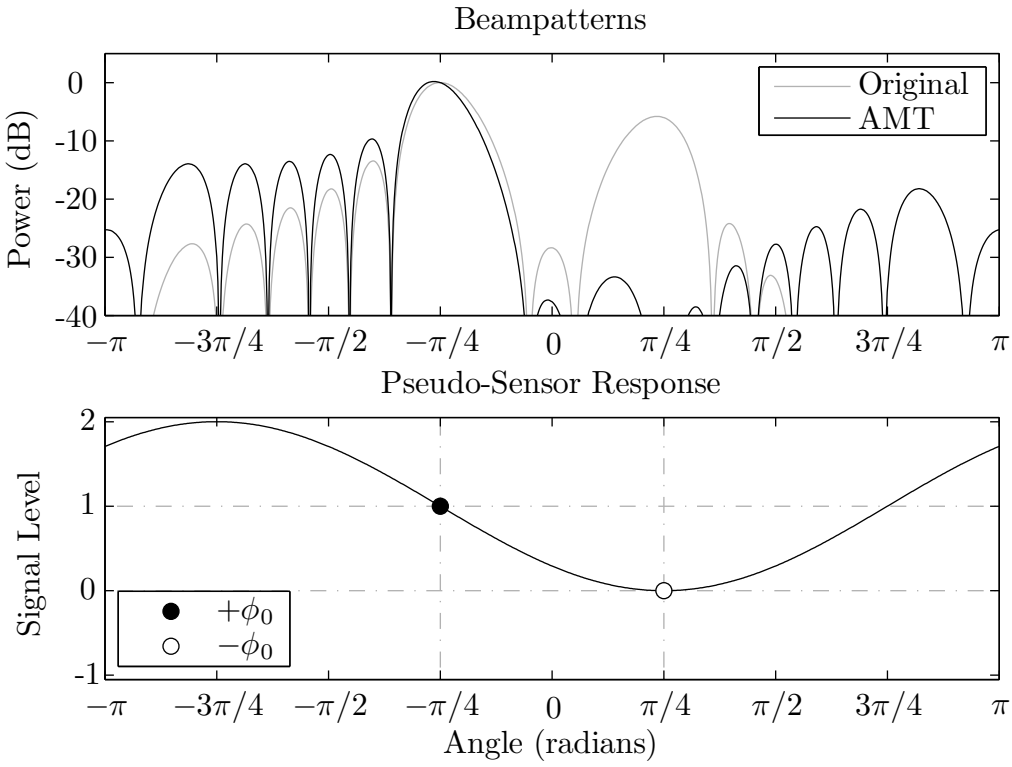

Figure 4.1.2: AMT beampattern: second-order constraints, uniform taper

sensor gives the pseudo-sensor weighting

$$\mathbf{p} \propto \begin{bmatrix} 1 \\ -\cos\phi_0 \\ \sin\phi_0 \end{bmatrix}. \tag{4.1.2}$$

With a uniform spatial taper, this weight yields modified beampatterns like the example shown in Figure 4.1.2. The stringent, second-order constraints elevate the pseudo-sensor response away from the null. This elevated pseudo-sensor response raises the sidelobes of the VSA beampattern to significant levels and increases the sensitivity factor of the weight by 6.02 dB, suggesting the weight is not robust. Replacing the uniform spatial taper with a 25 dB Taylor taper improves the sidelobe structure as shown in Figure 4.1.3 but increases the sensitivity factor to 6.45 dB. Furthermore, the spatial taper does not reduce sidelobes to the desired level because they are modulated higher by the pseudo-sensor response.

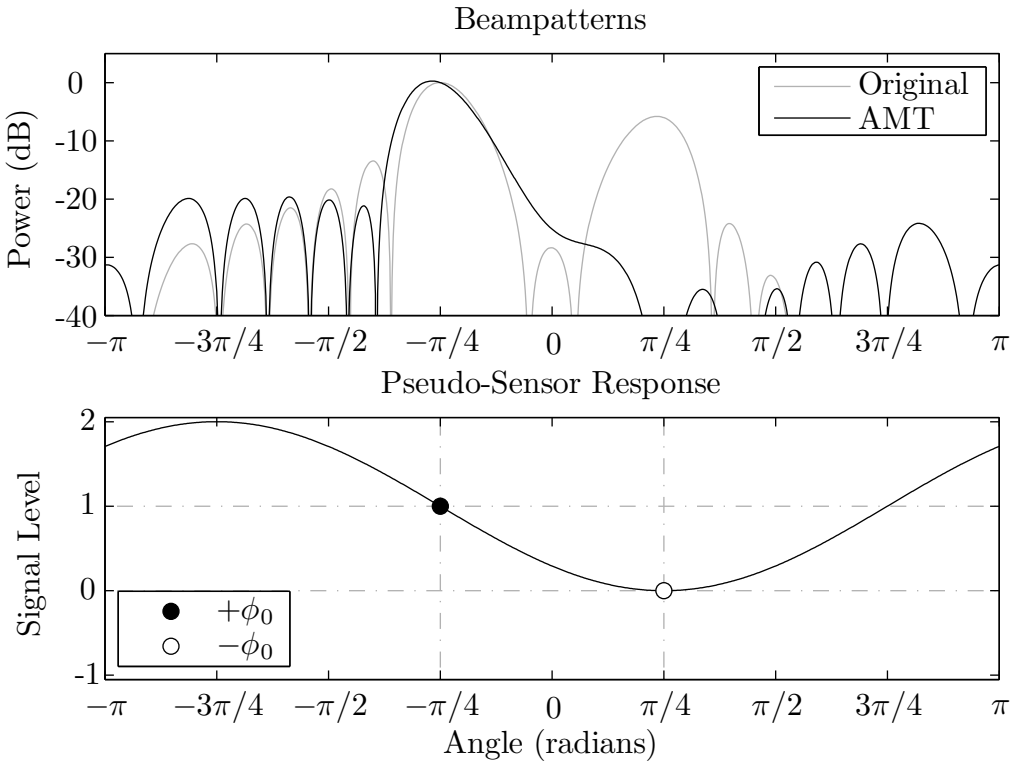

Figure 4.1.3: AMT beampattern: second-order constraints, 25 dB Taylor taper

An alternative approach defines

$$\mathbf{h}(\phi) \triangleq \begin{bmatrix} 1 \\ \cos\phi \\ \sin\phi \end{bmatrix} \tag{4.1.3}$$

and chooses the "optimum" pseudo-sensor weighting

$$\mathbf{p} \;\propto\; \left(\mathbf{I} - \frac{\mathbf{h}(-\phi_0)\mathbf{h}^T(-\phi_0)}{\mathbf{h}^T(-\phi_0)\mathbf{h}(-\phi_0)}\right)\mathbf{h}(\phi_0) \tag{4.1.4}$$

$$= \;\mathbf{h}(\phi_0) - \mathbf{h}(-\phi_0)\cos^2\phi_0. \tag{4.1.5}$$

This weighting minimizes the sensitivity factor subject to the point null constraint at $-\phi_0$ and the unity gain constraint at ϕ_0. The form of Equation 4.1.4 highlights its interpretation as a projection of the unmodified weighting, $\mathbf{h}(\phi_0)$, into the space orthogonal to $\mathbf{h}(-\phi_0)$. The optimum weighting produces beampatterns like the ex-

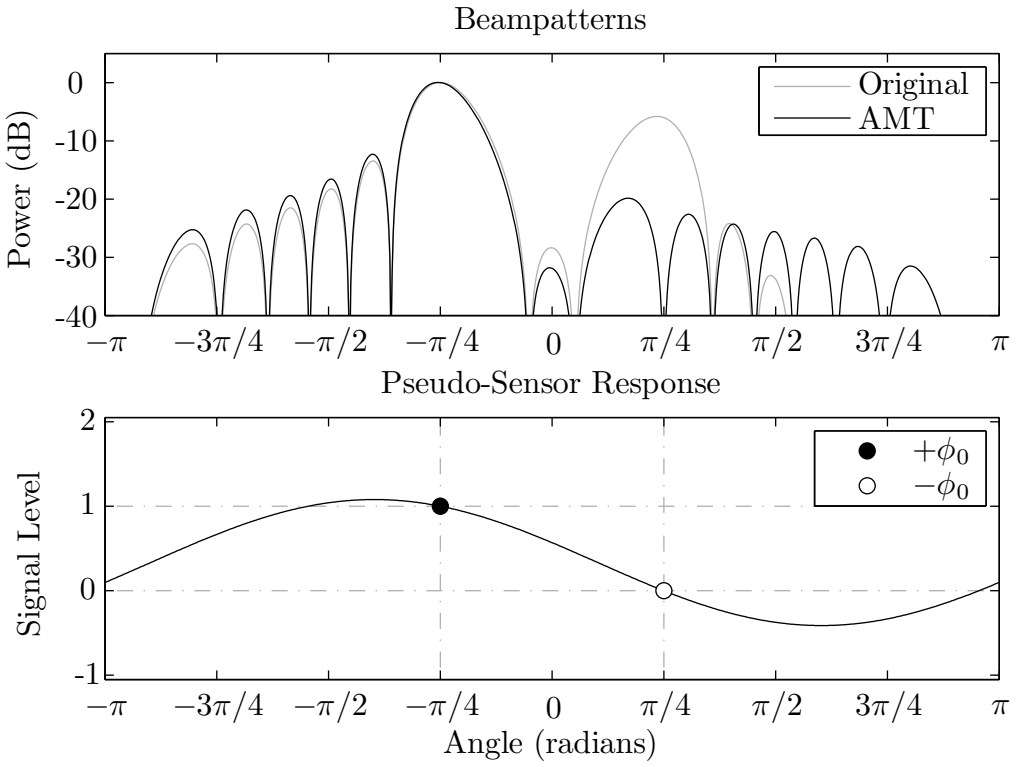

Figure 4.1.4: AMT beampattern: "optimum" null, uniform taper

ample in Figure 4.1.4. The sensitivity factor of the example weight is elevated by only 1.25 dB and the pressure-sensor ambiguity is reduced to a reasonably low level, but the sidelobes are still higher than desired. Applying a 25 dB Taylor taper results in the beampattern shown in Figure 4.1.5. Although the spatial taper lowers the sidelobes to an acceptable level, the pressure-sensor ambiguity again becomes the dominant feature. The addition of a spatial taper increases the sensitivity factor by 1.68 dB compared to the original, or conventional, weights.

Starting from the result in Figure 4.1.5, two final modifications are worth mentioning. First, rather than constraining the null and minimizing the sensitivity factor, one could constrain the sensitivity factor and minimize the power at $-\phi_0$. Applying such a constraint makes the weights more robust at the expense of nulling the pressure-sensor ambiguity. Because the pseudo-sensor weighting is a small vector, the sensitivity constraint is computationally efficient. Second, one could offset the null placement slightly to avoid an uneven splitting of the ambiguity. The beampattern

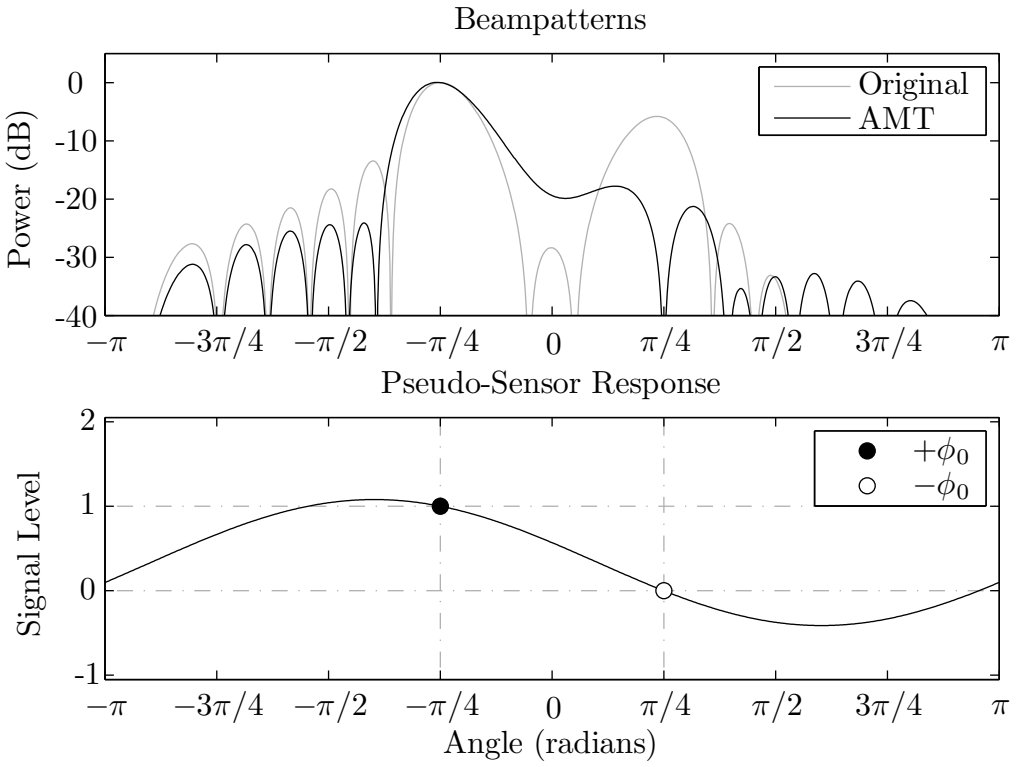

Figure 4.1.5: AMT beampattern: "optimum" null, 25 dB Taylor taper

resulting from an "offset" null is illustrated in Figure 4.1.6. Offsetting the null moves it closer to ϕ_0 and thus raises the sensitivity factor slightly to 1.85 dB.

4.1.3 Limitations of the AMT Approach

Pattern multiplication is simple and intuitive, but it does not fully exploit the capabilities of a vector-sensor array. Requiring that all vector-sensors form identical pseudo-sensors is overly restrictive. Although some weights factor this way, many useful weights do not.

The weights designed using pattern multiplication are restricted by the shape and sensitivity of the pseudo-sensor response. Without losing generality, the vector-sensor weight is parameterized as

$$\mathbf{p} = \begin{bmatrix} \alpha \\ -\beta \cos \phi_m \\ -\beta \sin \phi_m \end{bmatrix} \tag{4.1.6}$$

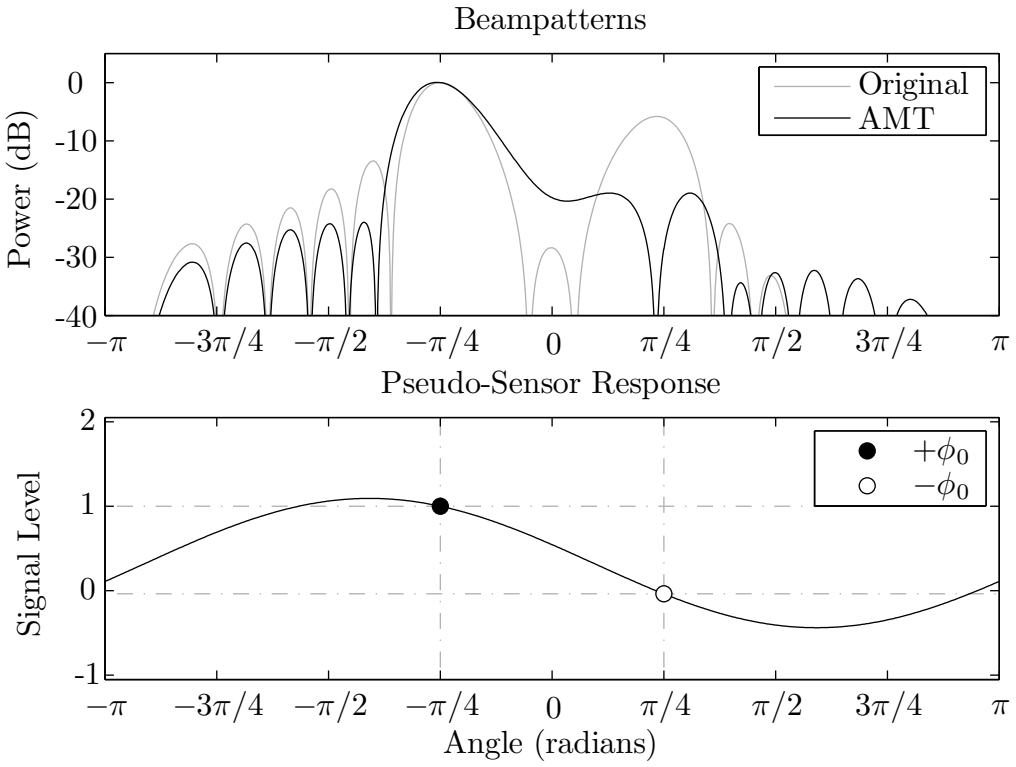

Figure 4.1.6: AMT beampattern: "offset" null, 25 dB Taylor taper

with $\beta \geq 0$. With this parameterization, the shape of the pseudo-sensor response is restricted to the linearly transformed cosine function

$$y_p(\phi) \triangleq \mathbf{p}^T \mathbf{h}(\phi) \tag{4.1.7}$$

$$= \alpha - \beta \cos(\phi - \phi_m), \tag{4.1.8}$$

which has a minimum at $y_p(\phi_m) = \alpha - \beta$ and a maximum at $y_p(\pi + \phi_m) = \alpha + \beta$. Thus, the pseudo-sensor response has at most two nulls and only one minimum. Requiring unity gain at ϕ_0 leaves only two degrees-of-freedom for nulling the pressure ambiguity, *regardless of the number of sensors in the array.* The sensitivity factor, $\alpha^2 + \beta^2$, is constrained for robust weights, leaving very little freedom in the design.

A final critique of pattern multiplication is that it does not directly address the design objectives stated in this work: low sensitivity, a narrow beamwidth, and low sidelobe levels. Although many of the *ad hoc* extensions to pattern multiplication address these objectives, the resulting weights are not optimal with respect to any

specific design criterion.

4.2 Designs Using A Priori Noise Distributions

Another large class of algorithms for beampattern design is based on a priori knowledge of the noise environment. This section mentions several such algorithms and make the assumed noise distribution explicit for each. A novel result for VSA spatially spread sources is included. The section concludes with a critique of the technique. As before, all examples in this section are with an $N = 10$ element vector-sensor array at $f = 5/7 f_d$, steered to $\phi_0 = -\pi/4$.

If the noise distribution is known, the "optimum" choice of beamforming weights is the solution to the minimum variance distortionless response (MVDR) problem:

$$\begin{aligned} \text{minimize} \quad & \mathbf{w}^H \widetilde{\mathbf{R}} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^H \mathbf{v}_0 = 1 \end{aligned} \tag{4.2.1}$$

for the noise covariance matrix, $\widetilde{\mathbf{R}}$, and signal replica vector, $\mathbf{v}_0$. The linear constraint avoids distortion of signals in the replica direction; the quadratic objective minimizes the leakage of interference into the beamformer output. For invertible $\widetilde{\mathbf{R}}$ the weight vector has a closed form,

$$\mathbf{w} = \frac{\widetilde{\mathbf{R}}^{-1} \mathbf{v}_0}{\mathbf{v}_0^H \widetilde{\mathbf{R}}^{-1} \mathbf{v}_0}. \tag{4.2.2}$$

For singular $\widetilde{\mathbf{R}}$, it is convenient to define the MVDR weights in the limit

$$\mathbf{w} = \lim_{\epsilon \to 0} \frac{(\widetilde{\mathbf{R}} + \epsilon^2 \mathbf{I})^{-1} \mathbf{v}_0}{\mathbf{v}_0^H (\widetilde{\mathbf{R}} + \epsilon^2 \mathbf{I})^{-1} \mathbf{v}_0}. \tag{4.2.3}$$

The above limit exists for any covariance matrix, but care must sometimes be taken to ensure numerical stability. Equation 4.2.3 has many natural interpretations: 1) it gives the minimum norm solution to the under-determined problem, 2) the resulting weight vector has the smallest sensitivity factor of any solution, and 3) the case of a degenerate Gaussian distribution is treated properly as a limit.

4.2.1 Spatially Discrete Sources and Point Nulls

A common and practical approach to VSA beampattern design places a null on the pressure-sensor ambiguity. The general formulation of this problem includes a sensitivity constraint or diagonal loading. The equivalent a priori noise distribution is

$$\widetilde{\mathbf{R}} = \mathbf{v}_b \mathbf{v}_b^H + \epsilon^2 \mathbf{I}, \tag{4.2.4}$$

where $\mathbf{v}_b$ is the signal replica of the pressure-sensor ambiguity, or "backlobe." The diagonal loading factor, ϵ^2, arises in many contexts, often as a regularization term or as the Lagrange multiplier for a quadratic/sensitivity constraint.

Although this "point null" approach seems more formal than the *ad hoc* designs using pattern multiplication, the two techniques are equivalent. The equivalence is shown by applying the matrix inversion lemma to Equation 4.2.2 with the noise distribution from Equation 4.2.4:

$$\begin{aligned} \mathbf{w} &\propto \left[\mathbf{v}_b \mathbf{v}_b^H + \epsilon^2 \mathbf{I}\right]^{-1} \mathbf{v}_0 \\ &\propto \left[\mathbf{I} - \frac{\mathbf{v}_b \mathbf{v}_b^H}{\epsilon^2 + 2N}\right] \mathbf{v}_0 \\ &= \mathbf{v}_0 - \mathbf{v}_b \left(\frac{2N}{\epsilon^2 + 2N} \cos^2 \phi_0\right). \end{aligned} \tag{4.2.5}$$

Because the backlobe is a pressure-sensor/phase ambiguity, both the signal replica and the backlobe replica have a common phase vector, $\mathbf{v}_p$:

$$\mathbf{v}_0 = \mathbf{v}_p \otimes \mathbf{h}(+\phi_0) \tag{4.2.6}$$

$$\mathbf{v}_b = \mathbf{v}_p \otimes \mathbf{h}(-\phi_0). \tag{4.2.7}$$

For more on the Kronecker product representation of the vector-sensor array beampattern, see [1, Ch. 2]. Substituting these expressions and using the properties of the

Kronecker product gives

$$\begin{aligned} \mathbf{w} &\propto [\mathbf{v}_p \otimes \mathbf{h}(+\phi_0)] - [\mathbf{v}_p \otimes \mathbf{h}(-\phi_0)] \left(\frac{2N}{\epsilon^2 + 2N} \cos^2 \phi_0 \right) \\ &= \mathbf{v}_p \otimes \overline{\mathbf{p}}, \end{aligned} \tag{4.2.8}$$

where

$$\overline{\mathbf{p}} \triangleq \mathbf{h}(+\phi_0) - \mathbf{h}(-\phi_0) \left(\frac{2N}{\epsilon^2 + 2N} \cos^2 \phi_0 \right). \tag{4.2.9}$$

Equation 4.2.8 reveals that the optimum weight resulting from this a priori distribution naturally takes the form of pattern multiplication (Section 4.1.1). Furthermore, the weight is equivalent to Equation 4.1.5 in the limit $\epsilon^2 \to 0$.

Designing VSA beampatterns using spatially discrete sources, or point nulls, extends to multiple sources at arbitrary locations. Such approaches may yield acceptable beampatterns, but no suitable method for determining the placement of sources/nulls is known. With pressure-sensor arrays, polynomial and Fourier transform properties analytically provide zero/null placement for Chebyshev, Taylor, Villeneuve, and other beampatterns. With vector-sensor arrays, however, no such analysis exists.

4.2.2 Spatially Spread Sources and Sector Nulls

As shown in Figure 4.1.6, the point null technique (or the equivalent array modulation technique) is unable to reduce the entire backlobe to a sufficiently low level. Choosing an effective set of constraint points is difficult in the vector-sensor array case, so extending the technique to multiple linear constraints is problematic.

An alternative approach uses the novel results in Section 2.4 to place a "sector null" in the direction of the pressure ambiguity. The a priori noise covariance for this approach is generally

$$\widetilde{\mathbf{R}} = \mathbf{R}_b + \epsilon^2 \mathbf{I}, \tag{4.2.10}$$

where $\mathbf{R}_b$ is the covariance matrix of a spatially spread source located in the backlobe direction. One of the three techniques in Section 2.4 is easily used to obtain $\mathbf{R}_b$. Because the covariance matrix of the spatially spread source is not full rank, some

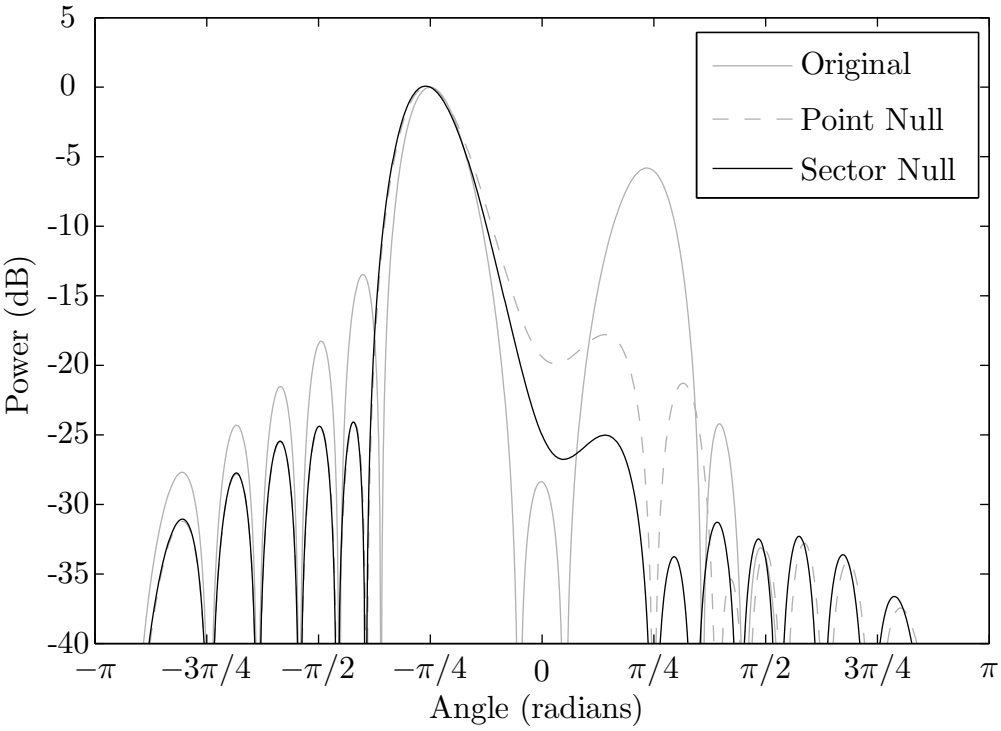

Figure 4.2.1: Comparison of tapered point and sector null beampatterns

amount of diagonal loading is required or the limiting form in Equation 4.2.3 must be used. Figure 4.2.1 demonstrates that this approach produces acceptable beampatterns when tuned properly. In this case, even a small amount of uniform spatial spreading ($\sigma_u = 4 \times 10^{-3}$, about 2.8% of the beamwidth) and diagonal loading ($\epsilon^2 = 2N \times 10^{-4}$) substantially improves the backlobe null. A 25 dB Taylor taper is also applied to reduce the sidelobe levels. The point and sector null beampatterns have sensitivity factors of 1.68 dB and 1.71 dB, respectively.

Unfortunately, obtaining good beampatterns like the one shown in Figure 4.2.1 is difficult in practice. Unlike the point null technique, no closed form expression exists for either the covariance matrix or the weight vector. Widening the backlobe null relies heavily on the sub-dominant eigenvectors of $\mathbf{R}_b$, requiring burdensome, high-precision, numerical integration for some terms (see Equation 2.4.7). Furthermore, even a careful implementation of the sector null technique requires time-consuming parameter tuning and necessitates the use a spatial taper.

4.2.3 Limitations of the A Priori Approach

Section 4.2 presents a unified picture of many fixed weight design techniques for vector-sensor arrays. Viewing these techniques in the context of "a priori noise distributions" highlights the primary limitation common to all: no technique explicitly addresses the key objectives of low sidelobes, a narrow mainlobe, and low sensitivity. As such, each approach requires a patchwork of tools including diagonal loading and spatial tapers to meet the objectives of this chapter. In every case, the parameter tuning is *ad hoc* and not separable. Like VSA pattern multiplication, the a priori approach yields weights that are suboptimal with respect to the stated design objectives.

4.3 Designs Using Convex Optimization

Sections 4.1 and 4.2 summarize many typical techniques for VSA fixed weight design. The techniques discussed share one powerful criticism: they do not directly address the stated objectives of VSA beampattern design. Judging these techniques by beamwidth, sidelobe level, and sensitivity is, in a sense, asking for one thing but wanting another. The distinction between the implicit and explicit objectives is more than philosophical.

This section shows that explicitly optimizing for the design objectives yields three benefits. First, the resulting beampatterns are optimal in terms of one or more objective. The "optimality gap" of existing techniques is sometimes revealed to be substantial. Second, trade-offs between the multiple objectives become straightforward. Third, no *ad hoc* parameter tuning is required. The only parameters are the intuitive constraints applied to each design objective.

The techniques presented in this section connect the fields of array processing and convex optimization. Such a connection has been noted before in the optimization literature, the filter design literature, and the array processing literature. Similar techniques are applied to the related problem of FIR filter design in [23]. In [24],

the convex optimization material from [25] is applied to beampattern design. An iterative algorithm for beampattern design is also presented in [5, §3.9.3]. Finally, [26] illustrates the broad scope of applied convex optimization. Four contributions of this section are applying optimization techniques to vector-sensor arrays, studying the design problem in the appropriate multi-objective context, relating design criterion to popular results in pressure-sensor array processing, and describing an efficient algorithm for designing weights.

4.3.1 Spatial Quantization

Fixed weight design is often complicated by the uncountable number of points in the beampattern. Analytical methods, such as polynomial approximation and sampling of the Fourier transform, reduce the complexity of pressure-sensor array processing [5, §3.2-3.3]. Chapter 2 reveals that such methods generally do not apply to vector-sensor arrays. For vector-sensor arrays, spatial quantization converts the design problem into a manageable form.

The smooth nature of the vector-sensor array beampattern means it is approximated arbitrarily well by a finite number of sample points. Consider the important problem of constraining a vector-sensor array beampattern in some region. Section 2.3 shows that the deviation of the VSA beampattern between two sample points is bounded, so a finite sampling exists that constrains the beampattern to any arbitrary accuracy. The local Fourier transform property suggests even more: because the VSA beampattern behaves like a modified PSA beampattern on a a small scale, the same quantization scheme should work well with both array types. Existing work on FIR filter design (or equivalently, PSA beampattern design) utilizes a uniform grid in discrete-time frequency [27]. Relating this work to the vector-sensor array problem suggests a dense grid of $\approx 20N$ points in cosine-space ($\approx 10N$ per side) for a vector-sensor array beampattern. In some cases, an equally dense grid in angular space yields better results near array endfire. The tolerances in filter design are often tighter than in beampattern design, so this sampling appears more than adequate for

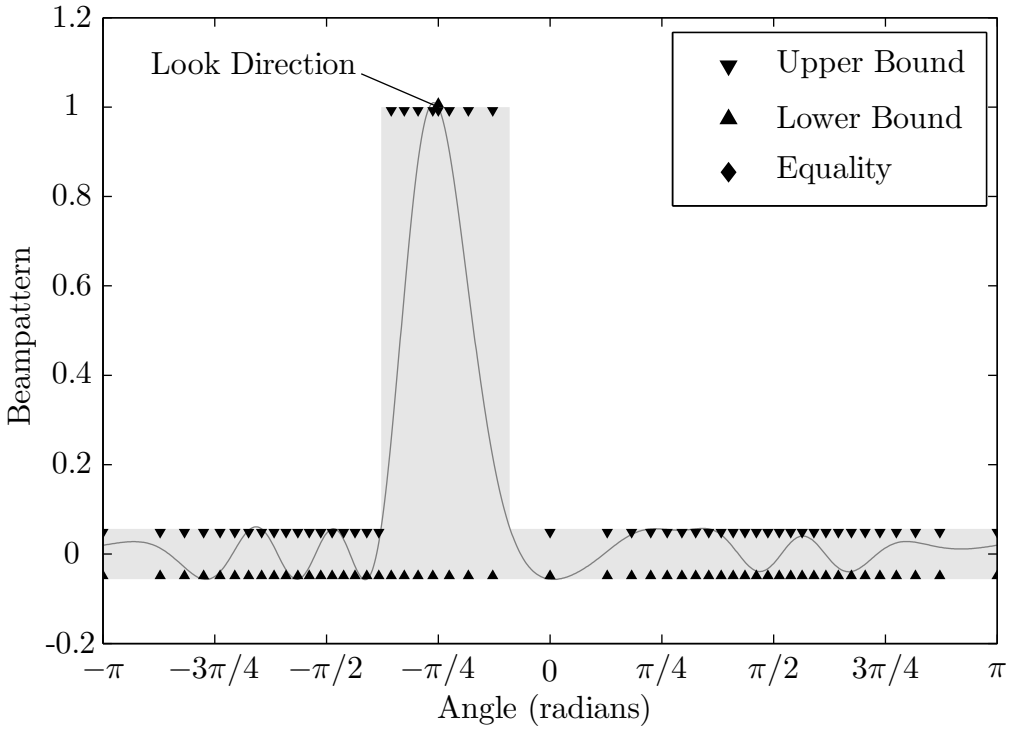

Figure 4.3.1: Coarse spatial quantization in cosine-space

most beampattern design problems. For extremely tight tolerances, exchange algorithms similar to [23] obtain an exact solution by iteratively updating the quantization grid. The results in this thesis use an extremely fine spatial quantization to ensure that any deviations are negligible. The results also focus primarily on quantization in the azimuthal dimension; Section 4.3.5 demonstrates that the resulting beampatterns behave well in both dimensions (azimuth and elevation).

Figure 4.3.1 illustrates a coarse spatial quantization in cosine-space for a vector-sensor beampattern steered to $\phi_0 = -\pi/4$. The array contains $N = 10$ vector-sensors with $f = 5/7 f_d$. The lone equality constraint forces unity gain in the look direction, *i.e.* a distortionless response. Upper and lower bounds on the beampattern, denoted by triangles in the figure, are enforced at each angular sample point. The quantization scheme partitions angular space into a mainlobe region and a sidelobe region.[1] In the mainlobe region, the beampattern is constrained to be no greater than unity to

[1]The mainlobe region in beampattern design relates to the passband *and* transition regions in FIR filter design.

avoid "squinting" of the beampattern. In the sidelobe region, the beampattern is constrained to be no greater than the desired sidelobe level. In both regions, the beampattern is constrained to be no less than the negative of the desired sidelobe level. The beampattern only becomes negative after the first null, so applying this lower bound in the mainlobe region avoids bifurcation of the beam. Together, these constraints approximate the shaded region in the figure. The coarse quantization in Figure 4.3.1 uses only $\approx 5N$ points, far fewer than the suggested $\approx 20N$. Even with such a coarse quantization, the beampattern deviates very little from the desired (shaded) region. Note that the cosine spacing is most dense at broadside where the beampattern changes rapidly with angle; it is least dense at endfire where the beampattern changes slowly with angle.

Spatial quantization has a rich and formal mathematical background. Although a full discussion is outside the scope of this document, the concepts are worth mentioning. The fundamentals of spatial quantization are deeply rooted in differential geometry. Many problems in signal and array processing involve inner products between a fixed vector and the set of points on a manifold in some high dimensional vector space. Examples include filter design (the fixed vector is the impulse response and the manifold is the Fourier basis) and general beampattern design (the vector is an aperture weighting and the manifold is the array manifold of replica vectors). Because the manifold is smooth, the inner product is smooth and is approximated to arbitrary accuracy by sampling. In this context, 1) the vector-sensor array manifold is a curve and 2) the cosine-spaced sampling selects points on the curve with approximately equal geodesic spacing. The constraints also have a geometric picture. Each linear inequality specifies a half-space; the collection specifies a high-dimensional polygon, or a polytope. An equality constraint specifies a plane. Quadratic constraints (introduced later) involve ellipsoidal regions. In this context, spatial quantization is equivalent to approximating a desired region, or "feasible set," with a polytope. Further treatment of spatial quantization also exists in the field of semi-infinite programming.

4.3.2 The Minimax Criterion

Because beampattern design is closely related to filter design, applying the same design criteria to both problems seems logical. The powerful "minimax" criterion is widely used in both filter design ([20, §7.4]) and PSA beampattern design ([5, §3.6]). Although the analytical techniques used in filter design do not apply with vector-sensor arrays, Section 2.3 lays the foundation for numerical minimax designs using linear programming. This section motivates the minimax criterion in array processing, applies it to VSA beampattern design, and discusses the results.

The minimax criterion arises in many contexts within array processing. Most generally, the design criterion is to *mini*mize the *max*imum error. Applying this concept to beampattern design translates into *mini*mizing the *max*imum sidelobe level. The minimax criterion is a natural choice when dealing with discrete interference or jamming, both common problems in array processing. The "worst-case" scenario for a given beampattern places all interference at the exact location of the maximum sidelobe, resulting in the lowest output signal-to-noise ratio. A minimax beampattern is therefore the best worst-case design. When weights must be designed without a priori information about interference location or power, the minimax criterion provides the lowest upper-bound on interference leakage. The minimax criterion also arises as a common objective in approximation theory. In this context, the minimax beampattern is an "optimal" approximation to the desired response. The maximum error is formally defined as the L^∞ norm, also known as the Chebyshev or uniform norm.

Designing a minimax VSA beampattern is equivalent to solving a real linear program (LP). Proving this equivalence requires only three steps thanks to the results in Section 2.1 and 4.3.1. First, recall from Section 4.3.1 that constraining the beampattern in the sidelobe region is equivalent to constraining it at finite number of points. A fixed spatial quantization can approximate the continuous constraint to an arbitrary accuracy. By iteratively modifying the constraint points, exchange algorithms find a quantization grid that yields the same result as the continuous constraint. Second, recall from Equation 2.1.18 that the VSA beampattern at any point is a real,

linear function of the real and imaginary parts of the weight vector. If the weight is represented as the real vector, $\overline{\mathbf{w}} \in \mathbb{R}^{3N}$, in the transformed space, the beampattern at azimuth angle ϕ is expressed easily as $y(\phi) = \overline{\mathbf{v}}^T(\phi)\overline{\mathbf{w}}$. Note that $\overline{\mathbf{v}}(\phi)$ is the *transformed* replica vector given by the real coefficients (see Section 2.1.2). The third and final step is writing the minimax criterion as a standard linear program. From the second step, an upper bound on the beampattern at angle ϕ is expressed as $\overline{\mathbf{v}}^T(\phi)\overline{\mathbf{w}} \leq \beta_{\mathrm{U}}$. A lower bound is similarly expressed as $-\overline{\mathbf{v}}^T(\phi)\overline{\mathbf{w}} \leq -\beta_{\mathrm{L}}$. Utilizing the spatial quantization illustrated in Figure 4.3.1 results in the minimax problem

$$\begin{array}{rrcll} \text{minimize} & \delta & & & \\ \text{subject to} & \overline{\mathbf{v}}^T(\phi_0)\overline{\mathbf{w}} & = & 1 & \\ & \overline{\mathbf{v}}^T(\phi_m)\overline{\mathbf{w}} & \leq & 1 & m \in \mathcal{M} \\ & \overline{\mathbf{v}}^T(\phi_m)\overline{\mathbf{w}} & \leq & \delta & m \in \mathcal{S} \\ & -\overline{\mathbf{v}}^T(\phi_m)\overline{\mathbf{w}} & \leq & \delta & m \in \{\mathcal{M} \cup \mathcal{S}\} \end{array} \quad . \tag{4.3.1}$$

The index sets $\mathcal{M}$ and $\mathcal{S}$ correspond to the mainlobe and sidelobe regions, respectively. In order, the constraints in Equation 4.3.1 enforce distortionless response, avoid mainlobe "squinting," upper-bound the sidelobe region, and lower-bound the beampattern. The minimization in Problem 4.3.1 is carried out over the variables δ and $\overline{\mathbf{w}}$. The objective and constraints in Problem 4.3.1 are linear functions of these variables, so the problem constitutes a linear program. Because linear programs are convex (see [25]), the minimax weights are a unique, global optimum. Refined numerical algorithms exist that solve linear programs quickly (worst-case polynomial time) to a very high precision. Two of the most common algorithms are interior point and active set (*i.e.* simplex) methods [26, 28].

Figure 4.3.2 provides an example VSA minimax beampattern. The beampatterns shown use the same parameters as the other sections: an $N = 10$ element vector-sensor array at frequency $f = 5/7 f_d$, steered to $\phi_0 = -\pi/4$. The mainlobe region corresponds to the ≈ 23 dB beamwidth of the conventional beampattern. Although the sensitivity factor of the minimax weights is very high (≈ 154 dB), the differ-

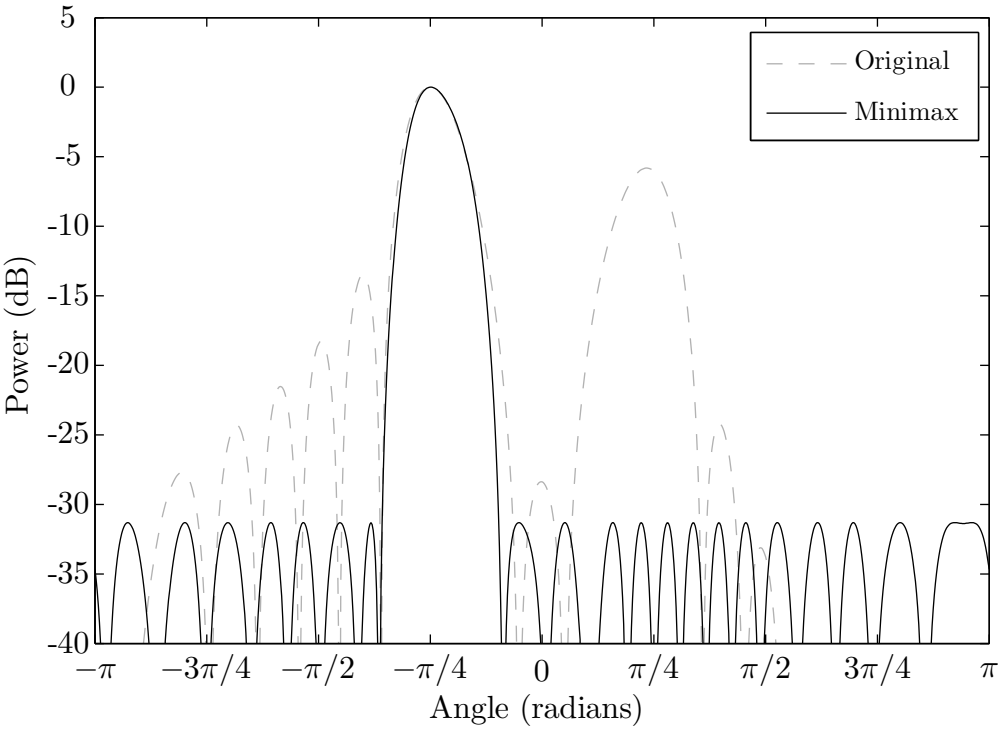

Figure 4.3.2: Example minimax beampattern

ence between the original (conventional) and minimax beampatterns in Figure 4.3.2 is striking. The sidelobes of the minimax beampattern are ≈ 31 dB, extremely low compared to the original beampattern and at least 6 dB lower than any other beampattern in this section. By definition, the maximum sidelobe level of the minimax beampattern is the lowest possible. Furthermore, the minimax beampattern achieves such a low sidelobe level *with a narrower mainlobe than the conventional beampattern.* Minimax beampatterns appear significantly better than the alternatives when only considering the design objectives of mainlobe width and sidelobe level.

Unfortunately, the minimax beampattern shown in Figure 4.3.2 is "too good to be true" in practice. Recall the objectives of fixed weight design: narrow beamwidth, low sidelobe levels, and low sensitivity. Although the minimax beampattern is optimal in terms of the first two objectives, its sensitivity to modeling errors often makes it impractical. Parameters subject to modeling errors are commonly treated as stochastic. Following the work in [15, 1], this thesis uses the extended Gilbert-Morgan mismatch

model, treating the vector-sensor gains, phases, positions, and rotations as zero-mean Gaussian random variables with standard deviations

Parameter	Std. Dev.	Units
Gain	0.1	Unitless
Phase	10	Degrees (°)
Position	0.1	Wavelengths (λ)
Rotation	10	Degrees (°)

.

These parameters represent a scenario with a substantial amount of mismatch. The beampattern with stochastic parameters becomes the *expected* response power as a function of angle, or

$$\begin{aligned} B(\phi) &\triangleq \mathrm{E}\left\{|y(\phi)|^2\right\} \\ &= \mathrm{E}\left\{|\mathbf{w}^H\mathbf{v}(\phi)|^2\right\} \\ &= \mathbf{w}^H\mathbf{R}_{\mathrm{mm}}(\phi)\mathbf{w}, \end{aligned} \tag{4.3.2}$$

where

$$\mathbf{R}_{\mathrm{mm}}(\phi) \triangleq \mathrm{E}\left\{\mathbf{v}(\phi)\mathbf{v}^H(\phi)\right\} \tag{4.3.3}$$

is the covariance matrix of a unit-power signal from angle ϕ. Note that this quadratic form of the beampattern is valid for *any* stochastic model. The minimax and conventional beampatterns from Figure 4.3.2 are shown under this mismatch model in Figure 4.3.3. Note the difference in scale from Figure 4.3.2. In this example, the extreme sensitivity of the minimax beampattern renders it useless for practical array processing. The large magnitude of each element in the minimax weight vector leads to an intuitive understanding of this sensitivity. To obtain the low sidelobe levels and narrow mainlobe shown in Figure 4.3.2, the minimax weights magnify minute differences between the (ideal) responses of the sensors. Errors in the element responses are typically much larger than these differences and are magnified by the minimax weights, resulting in an unpredictable beampattern. One reason that the minimax

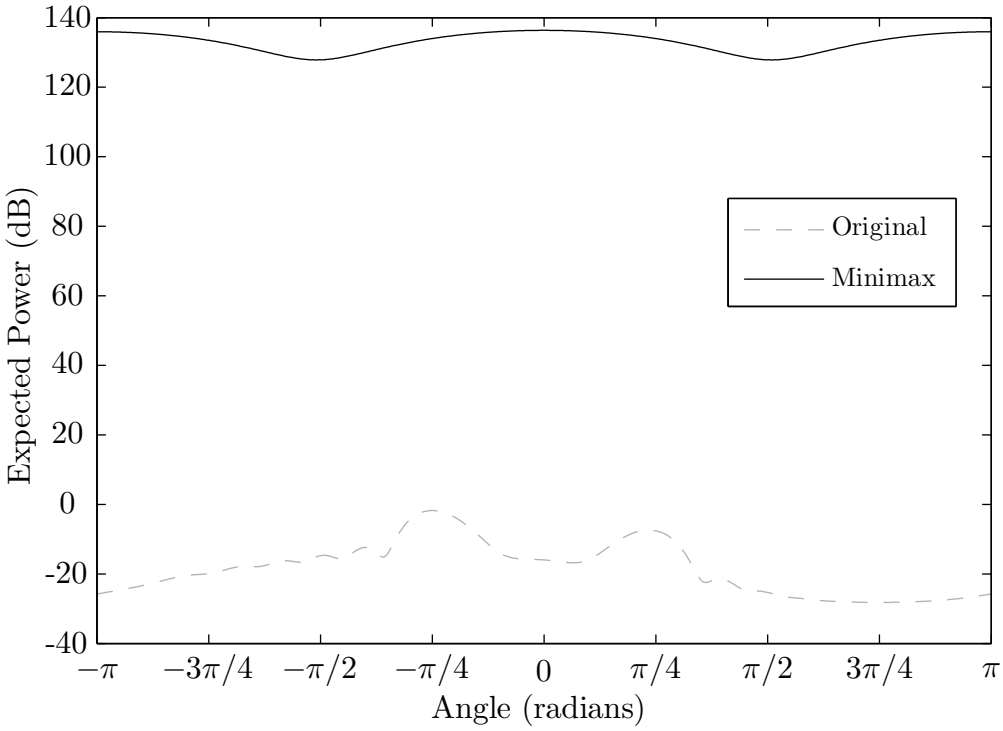

Figure 4.3.3: Effects of mismatch on a minimax beampattern

criterion is sometimes more effective in filter design than beampattern design is the uncertainties involved in array processing. Errors in temporal sampling (*i.e.* clock jitter) are typically much smaller than errors in spatial sampling (*i.e.* position errors).

Although the minimax beampattern is extremely sensitive, it serves two important purposes. First, it provides a bound on the achievable mainlobe and sidelobe levels for any fixed weight. The mainlobe in Figure 4.3.2 is only slightly narrower than the conventional beampattern; the difference is far less than half the beamwidth. This suggests that no VSA beampattern with acceptable sidelobe levels has a mainlobe that is significantly narrower than the conventional. A theoretical basis for this statement appears in the direction-of-arrival bound in [1] (see Figure 1.4.1). Second, the minimax beampattern reveals that vector-sensor array beampatterns sometimes defy conventional wisdom and must be carefully designed. Assuming a typical relationship between mainlobe and sidelobe levels, for instance, results in atypical behavior.

4.3.3 The Minimum Sensitivity Criterion

To avoid the extreme sensitivity demonstrated by the minimax beampatterns, an alternative criterion must balance all three design objectives. This section proposes a "minimum sensitivity" criterion which does just that, resulting in significant improvements and allowing explicit trade-offs between the three objectives. The minimum sensitivity criterion applied to vector-sensor array processing is a key contribution of this work, but the same concepts appear elsewhere in filter design ([23]) and array processing ([24]). The following paragraphs introduce the criterion, provide a few examples, and discuss one important trade-off involved in the design.

The previous section illustrates the need to unify the design objectives of a narrow mainlobe, low sidelobes, and low sensitivity into one design criterion. The key question is how to relate the intuitive notion of a "good" beampattern in terms of these competing objectives. The first step toward answering this question involves "Pareto optimal" solutions. Although a full discussion of multi-objective optimization is outside the scope of this document, Pareto optimal solutions capture the intuition of multi-objective optimality. A solution is Pareto optimal if improving it along one objective must necessarily worsen another. For any solution that is not Pareto optimal, a Pareto optimal solution exists that is better along at least one objective and no worse along the others. Therefore, the set of Pareto optimal solutions includes every preferred solution to the multi-objective problem.

One approach for exploring the set of Pareto optimal solutions is to minimize one objective and constrain the others. For the three-objective beampattern design problem, this yields only three candidate criteria. First, one could minimize the sidelobe level and constrain the beamwidth and sensitivity. This candidate suffers from two problems: 1) sensitivity is the most difficult of the three objectives to interpret and thus constrain, and 2) the criterion allows for undesirable, large variations in sidelobe level from beam to beam. A second candidate criterion is to minimize the beamwidth and constrain the sidelobe level and sensitivity. As with the first candidate, the sensitivity constraint is difficult to interpret. Furthermore, the resulting

minor improvements in beamwidth may not be worthwhile. The third and best candidate criterion is to minimize the sensitivity and constrain the beamwidth and sidelobe level. Choosing reasonable sidelobe and beamwidth constraints is straightforward, so this criterion yields Pareto optimal beampatterns with little tuning.

"Minimum sensitivity" designs do just as the name suggests: they minimize the sensitivity factor subject to a given mainlobe region and sidelobe level. Section 2.2 defines the (normalized) VSA sensitivity factor as

$$\xi \triangleq 2N\ \mathbf{w}^H\mathbf{w} \tag{4.3.4}$$

$$= \overline{\mathbf{w}}^T\mathbf{Q}\overline{\mathbf{w}}. \tag{4.3.5}$$

The diagonal matrix $\mathbf{Q}$ arises because of the transformation from the real weight vector $\overline{\mathbf{w}}$ to the complex weight vector $\mathbf{w}$. The form of $\mathbf{Q}$ depends on the number of elements (odd or even) but is clear from the transformation (see Section 2.1.2); an even number of elements gives $\mathbf{Q} = 4N\cdot\mathbf{I}$. Because the sensitivity factor is a quadratic form, minimum sensitivity weights are the solutions to a quadratic program. Section 4.3.2 demonstrates that the sidelobe constraints are linear. The beamwidth constraint simply determines the width of the mainlobe region in the spatial quantization scheme. The minimum sensitivity problem is therefore

$$\begin{array}{rrcll} \text{minimize} & \overline{\mathbf{w}}^T\mathbf{Q}\overline{\mathbf{w}} & & & \\ \text{subject to} & \overline{\mathbf{v}}^T(\phi_0)\overline{\mathbf{w}} & = & 1 & \\ & \overline{\mathbf{v}}^T(\phi_m)\overline{\mathbf{w}} & \leq & 1 & m \in \mathcal{M} \\ & \overline{\mathbf{v}}^T(\phi_m)\overline{\mathbf{w}} & \leq & \delta & m \in \mathcal{S} \\ & -\overline{\mathbf{v}}^T(\phi_m)\overline{\mathbf{w}} & \leq & \delta & m \in \{\mathcal{M} \cup \mathcal{S}\} \end{array} \quad , \tag{4.3.6}$$

where $\mathcal{M}$ and $\mathcal{S}$ represent the mainlobe and sidelobe regions as in Problem 4.3.1. Note that the optimization is performed only over $\overline{\mathbf{w}}$; the maximum sidelobe level, δ, is fixed. The matrix $\mathbf{Q}$ is positive definite, so the quadratic program is strictly convex with a unique global optimum. Refined algorithms exist that solve Problem

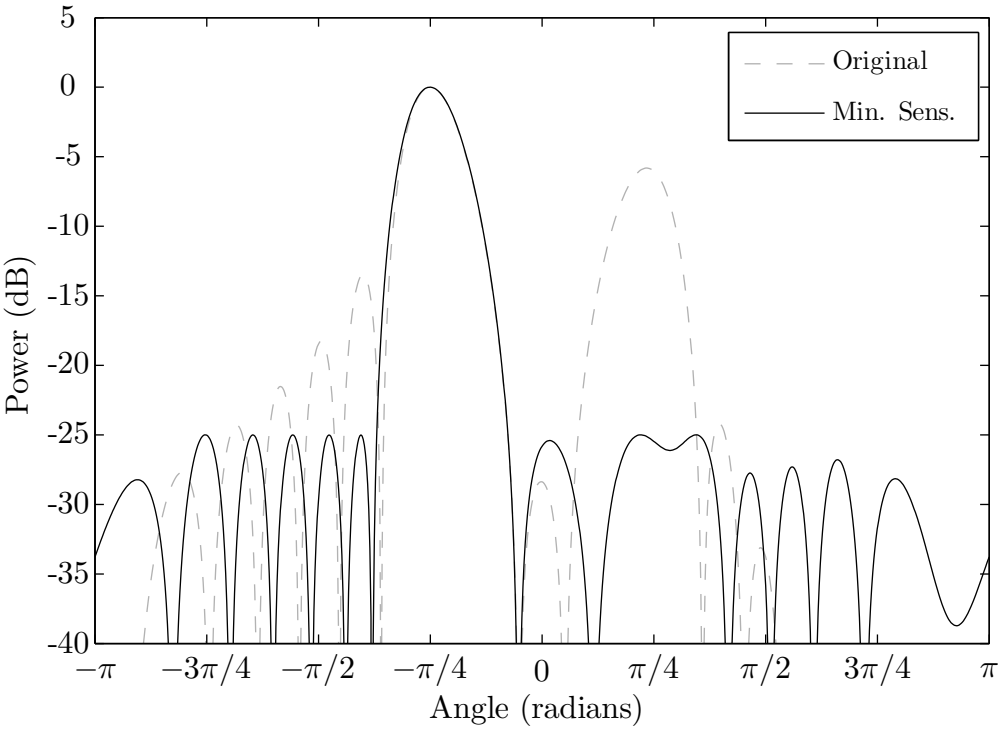

Figure 4.3.4: Example minimum sensitivity beampattern

4.3.6 quickly (in worst-case polynomial time) and to high precision [26, 28]. Section 4.3.4 discusses one such algorithm that solves Problem 4.3.6 efficiently by leveraging the special structure of the array processing problem.

Figure 4.3.4 provides an example of a minimum sensitivity beampattern. As before, the beampattern is for an $N = 10$ element vector-sensor array steered to $\phi_0 = -\pi/4$ at frequency $f = 5/7 f_d$. The maximum sidelobe constraint is -25 dB. The mainlobe region is determined by the width of a corresponding pressure-sensor array beampattern using a 25 dB Taylor taper. As expected, the sidelobes and backlobe are low (no higher than -25 dB) and the mainlobe is comparable to the conventional beampattern. By reducing the backlobe to a low sidelobe level, the minimum sensitivity beampattern in Figure 4.3.4 resolves the pressure-sensor ambiguity *as effectively as it resolves signals from any other direction.*

Figure 4.3.4 is an excellent result, but the true strength of the minimum sensitivity criterion is illustrated in Figure 4.3.5. Under the same mismatch scenario

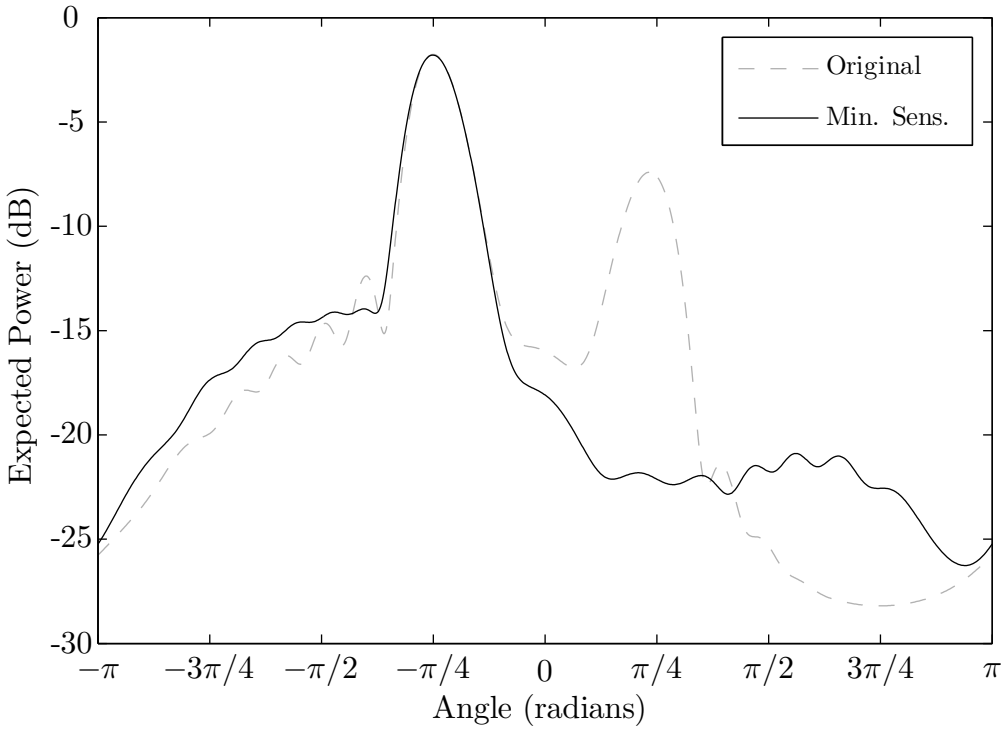

Figure 4.3.5: Effects of mismatch on a minimum sensitivity beampattern

as Section 4.3.2, the minimum sensitivity weights from Figure 4.3.4 still effectively resolve the pressure-sensor ambiguity. The effect of the mismatch is similar to beampattern modulation with directional additive noise (see [1, 15]). As the sensitivity factor increases, the noise level increases and the signal gain decreases. The minimum sensitivity weights reduce the level of additive noise close to the conventional beampattern and maintain the same level of gain. In short, the minimum sensitivity beampattern is *approximately as robust as the conventional beampattern but without the backlobe*. For comparison, the normalized sensitivity factor is $\xi \approx 154$ dB for the minimax weights in Figure 4.3.3 and $\xi \approx 1.37$ dB for the minimum sensitivity weights in Figure 4.3.5. The minimum value of ξ is unity, or 0 dB.

Minimum sensitivity beampatterns have a number of interesting properties that merit further discussion. Figures 4.3.6 and 4.3.7 illustrate a second example beampattern in ideal and mismatched scenarios. The beampatterns in these figures arise from a $N = 20$ element vector-sensor array at frequency $f = 5/7 f_d$, steered near endfire at

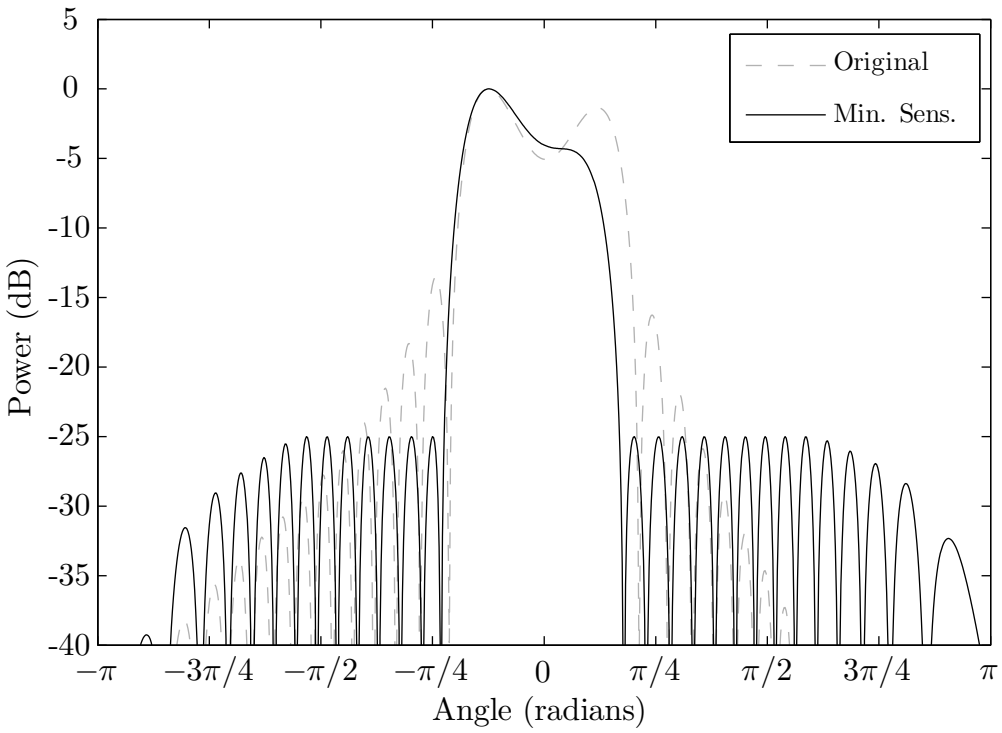

Figure 4.3.6: Example endfire minimum sensitivity beampattern

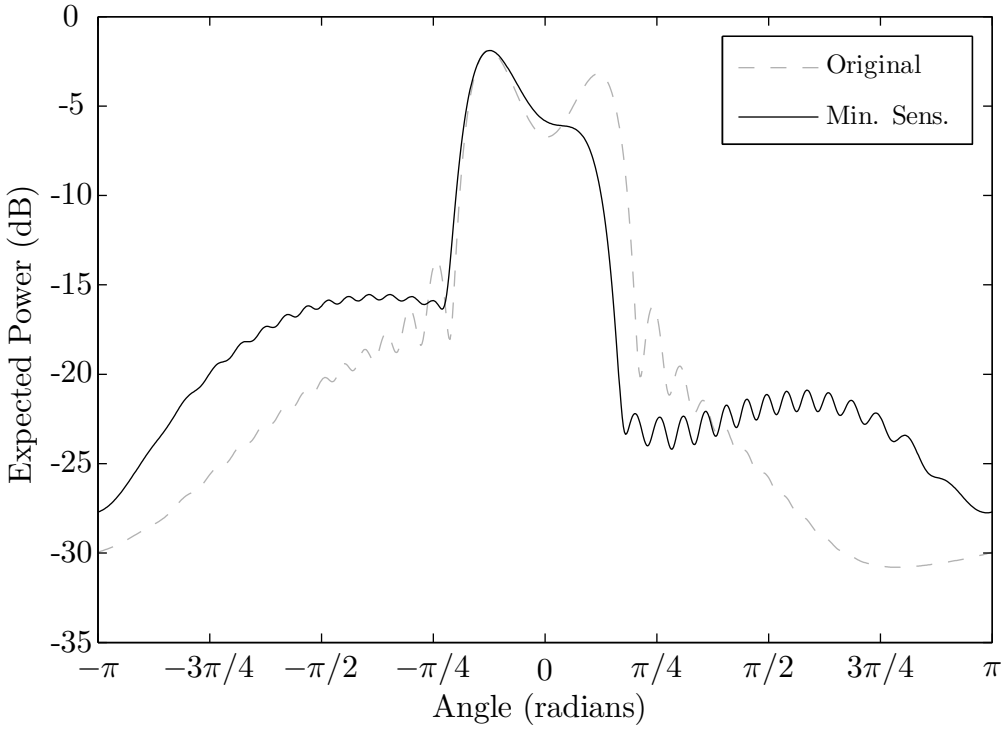

Figure 4.3.7: Effects of mismatch on an endfire minimum sensitivity beampattern

$\phi_0 = -\pi/8$. The mismatch parameters are the same as before; the sidelobe constraint remains -25 dB. Figure 4.3.6 reveals that the mainlobe and backlobe are merged when the look direction is within a beamwidth of array endfire. Attempting to reduce the backlobe to a low level is often infeasible or the resulting weights are extremely sensitive. Chapter 3 provides a theoretical basis for this endfire performance problem. The elegant solution, illustrated by the minimum sensitivity beampattern in Figure 4.3.6, extends the mainlobe region into the backlobe to widen the beamwidth until the weights become less sensitive. For the example shown, the mainlobe is extended until the sensitivity factor fell to $\xi = 3$ dB. Note that the resulting minimum sensitivity beampattern improves the ambiguity resolution, maintains a narrow beamwidth, and achieves low sidelobe levels. Another property demonstrated in Figures 4.3.6 and 4.3.7 is sidelobe roll-off. In this respect, the minimum sensitivity beampatterns are very similar to the popular Taylor taper. As the sidelobe constraint becomes more restrictive, more sidelobes meet it with equality. If the sidelobe constraint is very high (not binding), the minimum sensitivity weight equals the conventional weight. With the lowest sidelobe level, the minimum sensitivity weight is the minimax weight.

One important issue in the design of minimum sensitivity weights is the trade-off between sensitivity and maximum sidelobe level. As the maximum sidelobe level decreases, the sensitivity increases monotonically. Figure 4.3.8 illustrates a typical curve of minimum sensitivity versus sidelobe level for the familiar case with $N = 10$, $f = 5/7 f_d$, and $\phi_0 = -\pi/4$. The minimum sensitivity weights delineate the gray region in Figure 4.3.8. This region represents unachievable combinations of sidelobe level and sensitivity. At one extreme, the minimax weights achieve the lowest absolute sidelobe level. At the other extreme, the conventional weights achieve the lowest absolute sensitivity. The minimum sensitivity curve connecting these points divides into two regimes. The first regime in Figure 4.3.8 occurs for sidelobe levels above ≈ -25 dB, where a marginal decrease in sidelobe level changes the sensitivity very little.[2] The weights in this regime are generally well-behaved: coarse spatial quan-

[2]The "boundary" between the two regimes depends on the expected level of mismatch. The choice of ≈ -25 dB is explained in Section 4.3.4.

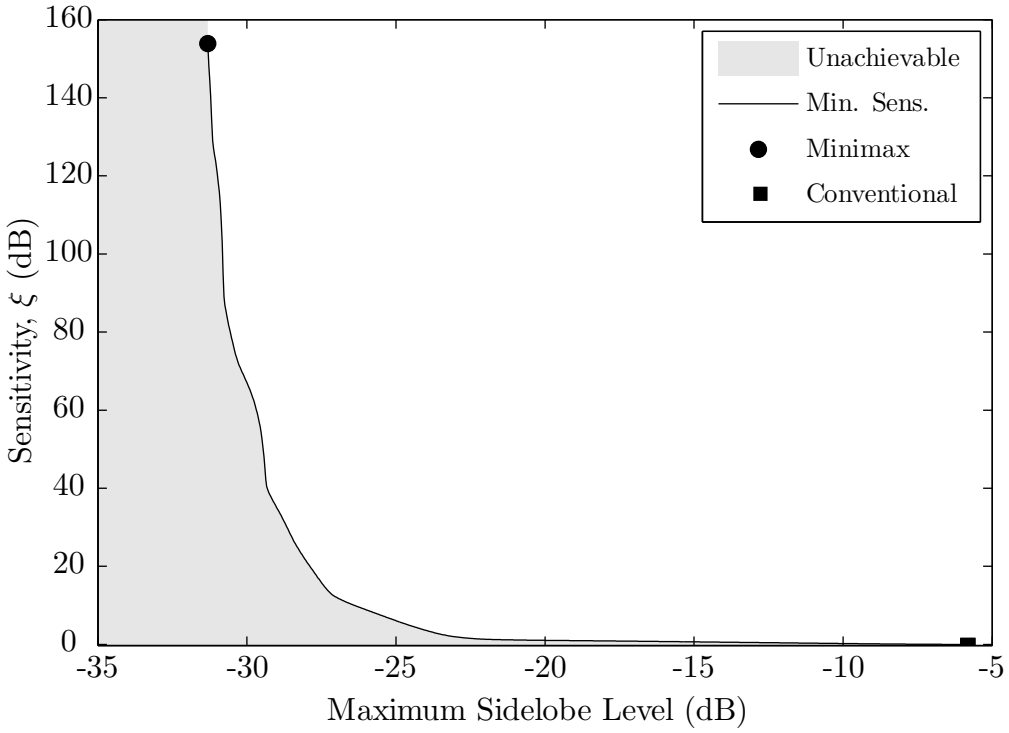

Figure 4.3.8: Minimum sensitivity versus maximum sidelobe level

tization yields acceptable weights, solving Problem 4.3.6 is numerically stable and efficient in practice, and the beampatterns are robust. The second regime in Figure 4.3.8 occurs for sidelobe levels below ≈ -25 dB, where a marginal decrease in sidelobe level significantly increases the sensitivity. Decreasing the sidelobe level of the ideal beampattern in this regime actually *increases* the expected sidelobe level under a mismatch scenario. The weights in this second regime are sometimes numerically unstable, difficult to compute in practice, and require a fine spatial quantization. Figure 4.3.8 reveals the importance of considering both sensitivity and sidelobe level in the design process. Although it is not shown here, the curve of minimum sensitivity versus beamwidth exhibits similar behavior.

4.3.4 Advanced Optimization Topics

The criteria discussed in Sections 4.3.2 and 4.3.3 provide a gentle introduction to fixed weight design using convex optimization. Several related optimization topics

are worth mentioning here, but a detailed discussion is outside the scope of the thesis. This subsection first provides heuristics and implementation details for the VSA minimum sensitivity problem. It then highlights the role of convex optimization in alternative design criteria.

Section 4.3.3 notes that the minimum sensitivity criterion requires "reasonable" choices for sidelobe level and beamwidth. In practice, simple heuristics aid in choosing both parameters. The approach taken here sets the maximum sidelobe level slightly below the average expected sidelobes of a mismatched, conventional beampattern (see [10], [1, §3.1], and [15]). Intuitively, this choice makes the sidelobes negligible compared to the noise introduced by the mismatch. A reasonable mainlobe region is chosen based upon a standard beamwidth, then expanded as necessary until the sensitivity factor falls below a threshold.

Many algorithms are capable of solving quadratic programs like Problem 4.3.6, but the special structure of minimum sensitivity design favors a modified active-set method [28, §16.6]. A properly implemented active-set algorithm for this problem is more precise, uses less memory, and requires orders of magnitude less computation than a general purpose solver. Active-set methods are similar in structure to the Remez exchange algorithm: they solve quadratic programs by determining the small set of constraints "active" at the optimum and solving a simpler, equality-constrained problem. Each iteration of the active-set method requires little computation. Only a few iterations are typically required to solve the problem from a good, feasible starting point. Adding an (exact) ℓ^∞ penalty, commonly referred to as the "big M" method, allows starting from any initial guess. Weights computed at nearby frequencies or beams often provide an excellent "warm-start" to the algorithm.

The two criteria discussed in this chapter are not the only criteria possible with convex optimization. Alternative criteria which seem intractable have efficient numerical solutions as long as the problems are convex [25]. One example is weight design based upon the expected beampattern. Consider an arbitrary array (including any pressure- or vector-sensor array) under any deterministic or stochastic model. Each point on the expected beampattern is a positive semidefinite quadratic form

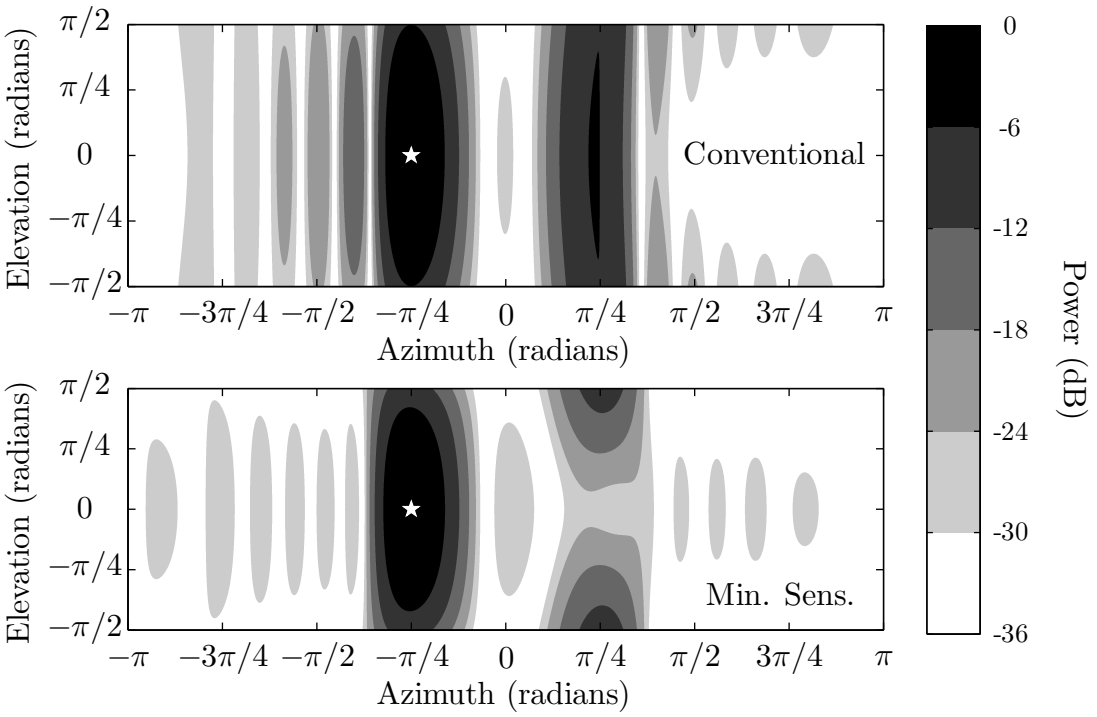

Figure 4.3.9: VSA beampatterns in azimuth and elevation

(like Equation 4.3.2). The minimax criterion applied to the expected beampattern yields a convex, second-order cone program (SOCP) that is efficient to solve [24, 26].

4.3.5 Beampatterns in Azimuth and Elevation

The spatial quantization scheme discussed in Section 4.3.1 only constrains the beampattern of a horizontal vector-sensor array at zero elevation, *i.e* in the horizontal plane. The following provides an example and brief argument that additional elevation constraints are unnecessary. The ability of a horizontal VSA to resolve signals at the same conical angle is limited. This resolution arises only from the directional elements; there is no vertical aperture to a horizontal array. Constraining the beampattern in the horizontal plane applies two constraints per conical angle, so the constrained beampattern has little freedom in the elevation dimension. Figure 4.3.9 confirms this behavior with 2-D contours of the beampatterns from Figure 4.3.4. The look direction is marked with a star. The minimum sensitivity beampattern is lower

than the conventional beampattern almost everywhere, even at nonzero elevation where there are no constraints. Constraining the beampattern of a horizontal VSA in the horizontal plane produces a weight that is well-behaved at all elevation angles.

4.4 Performance of Fixed Weights

The stated goal of this chapter is the design of fixed weights that achieve good performance like that predicted in Chapter 3. Section 4.3.3 describes the design of minimum sensitivity weights achieving 1) a narrow mainlobe, 2) low sidelobe levels, and 3) low sensitivity. Properties (2) and (3) ensure that these weights achieve performance close to the estimation bound in Section 3.2 for all but high-power interference.

For any fixed weight, the NRMSE performance as described in Chapter 3 depends only on the sensitivity factor and the backlobe rejection. The mean-squared error for the estimation scenario illustrated in Figure 3.2.1 is

$$\begin{aligned} \mathrm{MSE}_{\mathrm{fixed}} &= \mathrm{E}\left\{(\alpha_0 - \widehat{\alpha}_0)^2\right\} \\ &= \alpha_0^2 - 2\alpha_0 \mathrm{E}\left\{\widehat{\alpha}_0\right\} + \mathrm{E}\left\{\widehat{\alpha}_0^2\right\}. \end{aligned} \tag{4.4.1}$$

The power estimate, $\widehat{\alpha}_0$, obtained by any fixed weight is a scaled Chi-square random variable with two degrees of freedom, so $\mathrm{E}\left\{\widehat{\alpha}_0^2\right\} = 2\mathrm{E}\left\{\widehat{\alpha}_0\right\}^2$ and

$$\mathrm{MSE}_{\mathrm{fixed}} = \alpha_0^2 - 2\alpha_0 \mathrm{E}\left\{\widehat{\alpha}_0\right\} + 2\mathrm{E}\left\{\widehat{\alpha}_0\right\}^2. \tag{4.4.2}$$

The beam output power for the white noise scenario in Section 3.2 is

$$\mathrm{E}\left\{\widehat{\alpha}_0\right\} = \alpha_0 + \alpha_n \mathbf{w}^H \mathbf{w} + \alpha_1 |\mathbf{w}^H \mathbf{v}(-\phi_0)|^2. \tag{4.4.3}$$

The first term in Equation 4.4.3 is the target power, which remains undistorted by the fixed weight. The second term is the output power due to white noise and does not depend on the interference. The third term represents the backlobe interference

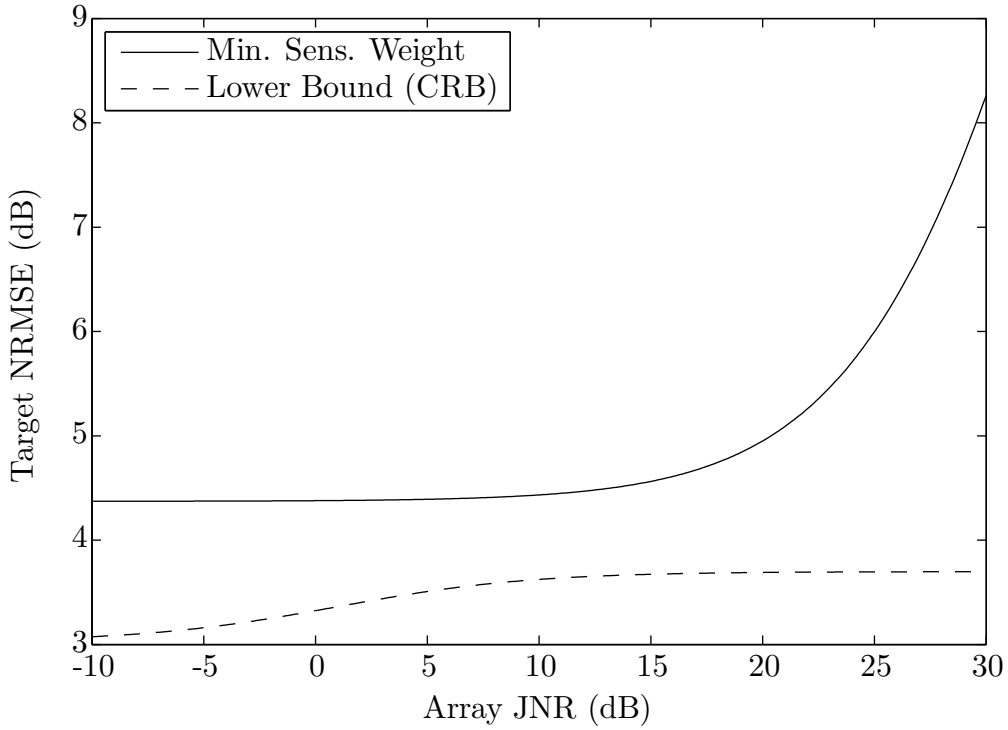

Figure 4.4.1: Efficiency of an example fixed weighting

"leakage." When the array JNR is small compared to the backlobe rejection, *e.g.*

$$\frac{\alpha_1}{\alpha_n} < \frac{1}{2} \cdot \frac{\mathbf{w}^H \mathbf{w}}{|\mathbf{w}^H \mathbf{v}(-\phi_0)|^2}, \tag{4.4.4}$$

the interference leakage is negligible and the NRMSE is nearly constant with JNR. For interference power above this threshold, the NRMSE increases quickly and diverges from the lower bound. The CRB is not a strict bound in this case because the estimator is biased, but it still commonly used for comparison [5, Chapters 8 and 9].

Because minimum sensitivity weights have low sidelobes and low sensitivity, their NRMSE performance remains close to the bound unless the JNR is high. Figure 4.4.1 illustrates the NRMSE performance of the minimum sensitivity weight from Figures 4.3.4 and 4.3.5. The performance of the fixed weight only diverges from the bound at JNR above the -25 dB sidelobe level. Fixed weights exist with better NRMSE performance under this specific scenario, but minimum sensitivity weights perform well for *any* sidelobe interferer without a priori knowledge of interference location.

Chapter 5

Subspace Adaptive Processing

Chapter 4 describes the design of robust, fixed weights for vector-sensor array processing. The minimum sensitivity beampatterns in Section 4.3.3 achieve substantial improvements over existing techniques, but fixed-weight beamforming is fundamentally limited in two important areas. First, the beamformer resolution is restricted by the beamwidth. Second, the interference rejection is restricted by the sidelobe levels. Improving performance in either area motivates the use of data adaptive beamforming.

Adaptive beamforming (ABF) improves resolution and interference rejection by adjusting the array weights based upon observed data. A thorough introduction to adaptive processing is provided in [5]. Adaptive processing typically proceeds in two steps. First, the second-order statistics of the data are estimated in the form of a covariance matrix. This chapter focuses on estimation using the "sample covariance matrix,"

$$\widehat{\mathbf{R}} \triangleq \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k \mathbf{x}_k^H , \tag{5.0.1}$$

because of its rapid convergence [9]. Recall from Section 3.1 that K is the number of observations, or "snapshots." The second step in adaptive processing computes weight vectors by substituting the sample covariance matrix for the true, or theoretical, covariance matrix. This step solves the MVDR problem in Equations 4.2.1 and 4.2.2 using the sample covariance matrix, $\widehat{\mathbf{R}}$.

The key problem in adaptive vector-sensor array processing is large dimension. The data dimension, D_0, increases from $D_0 = N$ with a pressure-sensor array to $D_0 = 4N$ with a vector-sensor array. As mentioned in Section 1.4.3, adaptive processing requires $\mathrm{O}(D^3)$ computation and $\mathrm{O}(D)$ training time. Increasing D by a factor of four necessitates algorithms that require less computation and converge more quickly. Processing power is easily increased, but training time is fundamentally limited by the stationarity of the data (see Section 1.4.3).

The work in this chapter does not overcome the fundamental problems of large dimension, it *circumvents* them by reducing the problem dimension without noticeably affecting optimum performance. Sections 5.1 and 5.2 introduce and formalize subspace processing. Section 5.3 derives an "optimum" subspace appropriate for beamforming an entire region. After a brief discussion of the mathematics behind subspace design, the chapter concludes by analyzing the performance of adaptive processing. Key contributions of this chapter are 1) a theoretical framework for subspace design, 2) a derivation of eigenbeam transformations within this framework, and 3) an approximation revealing the substantial dimension reduction achieved. The techniques in this chapter apply to any array, but the analysis and results are VSA-specific.

5.1 Introduction to Subspace Techniques

Subspace techniques reduce both computation and training time by performing standard adaptive processing in a low dimensional subspace. In standard adaptive processing, the input data is fed directly into some adaptive processor. The dimension of this "element-space" adaptive problem equals the data dimension, or $D = D_0$. Subspace adaptive processing, illustrated by the block diagram in Figure 5.1.1, projects the data into a low dimensional subspace before applying adaptive processing. The replica vectors used in subspace adaptive processing pass through the same transformation. The dimension of the adaptive problem is reduced to the dimension of the subspace, $D \ll D_0$. The reduced-dimension adaptive problem requires less computation and, more importantly, less training time than the element-space scheme.

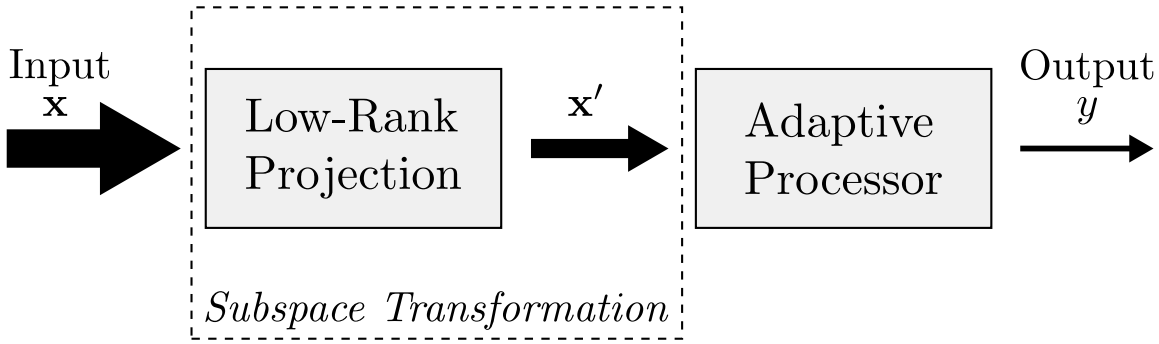

Figure 5.1.1: Adaptive processing in low-dimensional subspaces

Subspace adaptive processing has several advantages over other techniques for dealing with large dimension. First, it involves a simple linear transformation of the data. The standard processing techniques apply *without modification* to the transformed data: jointly Gaussian data remains jointly Gaussian, conventional and adaptive processing retain their properties, and optimum processing takes the same form. Second, it is compatible with other techniques. Proven techniques such as diagonal loading and dominant mode rejection remain effective [10, 11]; promising new techniques such as PCML still apply to the transformed data [12]. Third, subspace processing is computationally efficient, especially for static arrays. For transformations computed offline, the cost of applying the transformation is offset by the savings of inverting a smaller matrix.

Before discussing the fundamentals of subspace design, the relationship between subspaces and orthonormal transformation matrices is important to clarify. The set of all D-dimensional subspaces in $\mathbb{C}^{D_0}$ is called the (complex) Grassmann manifold, $\mathbb{G}(D, D_0)$. Each subspace can have infinitely many orthonormal bases, related by rotation, so there is generally not a one-to-one correspondence between $\mathbb{G}(D, D_0)$ and the orthonormal matrices $\left\{\mathbf{P} \in \mathbb{C}^{D_0 \times D} \mid \mathbf{P}^H\mathbf{P} = \mathbf{I}_D\right\}$. The loose notation $\mathbf{P} \in \mathbb{G}(D, D_0)$ indicates that $\mathbf{P}$ is an orthonormal basis for one subspace in $\mathbb{G}(D, D_0)$. The optimal subspace is often unique, but the transformation matrix is not. For a full treatment of optimization on the Grassmann manifold, see [29].

5.2 Inner Product Preservation

The primary obstacle to developing a theory of subspace processing is the lack of a useful optimality criterion. Many proven techniques such as beamspace processing [5, §7.10] and subarray processing are easily framed as subspace methods, but their choice of subspace is based on intuition. Comparing different subspaces requires a definition of optimality, an ideal subspace. The goal of subspace processing is to reduce dimension but leave the optimum output unchanged. Under what conditions is the subspace transformation "lossless"?

Subspace optimum processing is equivalent to element-space optimum processing if and only if the subspace transformation is inner product preserving. To ensure equivalent processing for the signals in $\mathbb{V}$, an orthonormal transformation matrix $\mathbf{P}$ must only satisfy

$$\langle \mathbf{v}_0 \, , \, \mathbf{v}_1 \rangle = \langle \mathbf{P}^H \mathbf{v}_0 \, , \, \mathbf{P}^H \mathbf{v}_1 \rangle \qquad \forall \; \mathbf{v}_0, \mathbf{v}_1 \in \mathbb{V}. \tag{5.2.1}$$

An informal proof of Equation 5.2.1 is straightforward: inner product preservation is equivalent to $\mathbb{V} \subseteq \text{span}(\mathbf{P})$, so applying the transformation performs a change of basis and leaves the optimum output unaltered.

Many conditions are equivalent to Equation 5.2.1, but the chosen form is most useful for several reasons. First, it provides a quantitative measure of subspace performance that is useful for comparison and design. Second, it provides intuition about how errors affect the output. If the norm of a vector is not preserved, the output signal-to-noise ratio is lowered. If the inner product between two vectors is not fully preserved, the ability to distinguish between the two signals (*e.g.* nulling) is affected. Third, inner products naturally extend concepts from filter design to the multidimensional case. The one-dimensional subspace design problem is closely related to filter design; norms are equivalent to the magnitude response of the filter.

The power of Equation 5.2.1 lies in approximation. An ideal subspace satisfying Equation 5.2.1 with equality is often not useful in practice. The ideal subspace for many problems is simply element-space, or $\mathbf{P} = \mathbf{I}$. Although it sometimes takes

the full dimension to satisfy with equality, inner product preservation is often well-approximated with low dimensional subspaces.

5.3 Regional Projections

A natural starting point for subspace design appears in the definition of inner product preservation. How can Equation 5.2.1 be approximated to support low-dimensional processing over an entire region? This section provides one answer in the form of "eigenbeam" transformations. Eigenbeam transformations are not new to array processing (see [30]), but the justification given for their use is often vague. This section derives eigenbeam transformations as the subspaces minimizing the squared error of the inner products.

5.3.1 Problem Description

Section 5.2 derives conditions corresponding to the "ideal" subspace. The design problem is now reduced to approximation, or selecting a subspace that minimizes some measure of deviation from the ideal. There are many useful error metrics, each of which defines an "optimal" subspace. This section describes the problem of minimizing a common and tractable metric, the total squared error.

Applying the minimum squared error criterion to subspace design involves formally defining the error and stating the problem. Section 5.2 reveals that the error between the ideal subspace and any orthonormal transformation, $\mathbf{P}$, is captured by the error in the inner products. For any two vectors $\{\mathbf{v}_0, \mathbf{v}_1\}$, the error in the inner product is

$$\epsilon\{\mathbf{v}_0, \mathbf{v}_1\} \triangleq \mathbf{v}_0^H \mathbf{v}_1 - \mathbf{v}_0^H \mathbf{P}\mathbf{P}^H \mathbf{v}_1. \tag{5.3.1}$$

The error implicitly depends on the transformation $\mathbf{P}$. Equation 5.3.1 must be considered over all pairs of vectors in some region, $\mathcal{K}$, of the manifold. The $\mathcal{K}$ considered in this chapter are regions in cosine-space, or u-space, but the derivation is valid for regions defined in any parameterization of the manifold (angular-space, wavenumber-

space, *etc.*). The error is minimized over the set of manifold vectors in the region,

$$\mathbb{V} \triangleq \{\mathbf{v}(\mathbf{k}) \mid \mathbf{k} \in \mathcal{K}\}. \tag{5.3.2}$$

Writing the minimization in terms of the region, $\mathcal{K}$, gives the optimization problem

$$\min_{\mathbf{P} \in \mathbb{G}(D, D_0)} \int_{\mathcal{K}} \int_{\mathcal{K}} \left| \epsilon \left\{ \mathbf{v}(\mathbf{k}_0), \mathbf{v}(\mathbf{k}_1) \right\} \right|^2 \, d\mathbf{k}_0 \, d\mathbf{k}_1. \tag{5.3.3}$$

Using the loose notation $\mathbf{v}_0 = \mathbf{v}(\mathbf{k}_0)$ and $\mathbf{v}_1 = \mathbf{v}(\mathbf{k}_1)$ makes the dependence on $\mathbf{k}$ implicit and produces a compact form of the problem:

$$\min_{\mathbf{P} \in \mathbb{G}(D, D_0)} \int_{\mathcal{K}} \int_{\mathcal{K}} \left| \epsilon \left\{ \mathbf{v}_0, \mathbf{v}_1 \right\} \right|^2 \, d\mathbf{k}_0 \, d\mathbf{k}_1. \tag{5.3.4}$$

The double integral over $\mathcal{K}$ in Equation 5.3.4 captures every pair of inner products.

5.3.2 Solution: Eigenbeams

The optimization problem in Equation 5.3.4 seems very difficult at first glance. It is non-convex, requires a search over complex manifolds, and involves difficult integrals. The solution, however, is powerful and elegant. The global optimum is easily computed by a singular value decomposition to high numerical precision.

The global optimum to Problem 5.3.4 is derived in three steps. First, the problem is modified to search over the orthogonal complement of the desired subspace. The inner product error in Equation 5.3.1 is easily written as

$$\epsilon\{\mathbf{v}_0, \mathbf{v}_1\} = \mathbf{v}_0^H \mathbf{P}_\perp \mathbf{P}_\perp^H \mathbf{v}_1, \tag{5.3.5}$$

where $\mathbf{P}_\perp \in \mathbb{G}(D_0 - D, D_0)$ is an orthonormal basis for the null-space of $\mathbf{P}$. Finding $\mathbf{P}$ in Problem 5.3.4 is equivalent to finding $\mathbf{P}_\perp$ in

$$\min_{\mathbf{P}_\perp \in \mathbb{G}(D_0 - D, D_0)} \int_{\mathcal{K}} \int_{\mathcal{K}} \left| \mathbf{v}_0^H \mathbf{P}_\perp \mathbf{P}_\perp^H \mathbf{v}_1 \right|^2 \, d\mathbf{k}_0 \, d\mathbf{k}_1. \tag{5.3.6}$$

Second, the integrals are removed from the problem. Expanding the objective in Equation 5.3.6 gives

$$\begin{aligned}
&\int_{\mathcal{K}}\int_{\mathcal{K}} \mathbf{v}_0^H \mathbf{P}_\perp \mathbf{P}_\perp^H \mathbf{v}_1 \; \mathbf{v}_1^H \mathbf{P}_\perp \mathbf{P}_\perp^H \mathbf{v}_0 \, d\mathbf{k}_0 \, d\mathbf{k}_1 \\
&= \int_{\mathcal{K}}\int_{\mathcal{K}} \mathrm{tr}\left(\mathbf{P}_\perp^H \mathbf{v}_0 \mathbf{v}_0^H \mathbf{P}_\perp \; \mathbf{P}_\perp^H \mathbf{v}_1 \mathbf{v}_1^H \mathbf{P}_\perp\right) \, d\mathbf{k}_0 \, d\mathbf{k}_1 \\
&= \int_{\mathcal{K}} \mathrm{tr}\left[\mathbf{P}_\perp^H \left(\int_{\mathcal{K}} \mathbf{v}_0 \mathbf{v}_0^H \, d\mathbf{k}_0\right) \mathbf{P}_\perp \; \mathbf{P}_\perp^H \mathbf{v}_1 \mathbf{v}_1^H \mathbf{P}_\perp\right] \, d\mathbf{k}_1 \\
&= \mathrm{tr}\left[\mathbf{P}_\perp^H \left(\int_{\mathcal{K}} \mathbf{v}_0 \mathbf{v}_0^H \, d\mathbf{k}_0\right) \mathbf{P}_\perp \; \mathbf{P}_\perp^H \left(\int_{\mathcal{K}} \mathbf{v}_1 \mathbf{v}_1^H \, d\mathbf{k}_1\right) \mathbf{P}_\perp\right] \\
&= \mathrm{tr}\left[\left(\mathbf{P}_\perp^H \mathbf{R}_{\mathcal{K}} \mathbf{P}_\perp\right)^2\right], \qquad (5.3.7)
\end{aligned}$$

where $\mathbf{R}_{\mathcal{K}}$ is the covariance matrix

$$\mathbf{R}_{\mathcal{K}} \triangleq \int_{\mathcal{K}} \mathbf{v}(\mathbf{k}) \mathbf{v}^H(\mathbf{k}) \, d\mathbf{k}. \qquad (5.3.8)$$

The first step above treats the scalar integrand as the trace of a 1×1 matrix and uses the trace identity $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$; the remaining steps utilize the linearity of the trace function. Note that $\mathbf{R}_{\mathcal{K}}$ is the covariance matrix of isotropic noise over $\mathcal{K}$.

The third and final step to solving Problem 5.3.4 determines the global optimum using the Poincaré separation theorem [31, §4.3]. Let $\lambda_i(\mathbf{R})$ represent the i^{th} largest eigenvalue of the $n \times n$ Hermitian matrix $\mathbf{R}$. For any $n \times r$ orthonormal transformation matrix $\mathbf{P}$, the Poincaré separation theorem states that

$$\lambda_i(\mathbf{R}) \geq \lambda_i(\mathbf{P}^H \mathbf{R} \mathbf{P}) \geq \lambda_{n-r+i}(\mathbf{R}) \qquad i = 1, \ldots, r. \qquad (5.3.9)$$

The upper and lower bounds in Equation 5.3.9 are achieved *simultaneously for all eigenvalues* when $\mathbf{P}$ spans the dominant and sub-dominant subspace of $\mathbf{R}$, respec-

tively.[1] One implication of Equation 5.3.9 is that

$$\sum_{i=1}^{r} f\left[\lambda_i(\mathbf{P}^H\mathbf{R}\mathbf{P})\right] \tag{5.3.10}$$

is maximized/minimized for any monotonic function $f(\lambda)$ when $\mathbf{P}$ spans the dominant/sub-dominant subspace of $\mathbf{R}$. Substituting Equation 5.3.7 into Problem 5.3.6 yields

$$\min_{\mathbf{P}_\perp \in \mathbb{G}(D_0-D,D_0)} \sum_{i=1}^{D} \lambda_i^2\left(\mathbf{P}_\perp^H\mathbf{R}_\mathcal{K}\mathbf{P}_\perp\right). \tag{5.3.11}$$

Because $\mathbf{R}_\mathcal{K}$ is positive semidefinite, each eigenvalue is nonnegative and $f(\lambda) = \lambda^2$ is monotonic. Using Equation 5.3.10, the global optimum is achieved when $\mathbf{P}_\perp$ spans the sub-dominant subspace of $\mathbf{R}_\mathcal{K}$, or when $\mathbf{P}$ spans the dominant subspace. The global optimum subspace is always unique when the eigenvalues of $\mathbf{R}_\mathcal{K}$ are unique.

The above derivation is complex, but obtaining the optimum transformation is relatively simple. An orthonormal transformation to the optimum D-dimensional subspace is found in three steps:

1. Form the "isotropic" covariance matrix over $\mathcal{K}$:

$$\mathbf{R}_\mathcal{K} \triangleq \int_\mathcal{K} \mathbf{v}(\mathbf{k})\mathbf{v}^H(\mathbf{k})\, d\mathbf{k}$$

2. Compute the singular value decomposition of $\mathbf{R}_\mathcal{K}$:

$$\mathbf{R}_\mathcal{K} = \mathbf{U}\mathbf{S}\mathbf{U}^H$$

3. Concatenate the dominant eigenvectors from $\mathbf{U}$:

$$\mathbf{P}_\mathrm{E} \triangleq \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_D \end{bmatrix}$$

[1] The eigenvectors with the largest eigenvalues span the dominant subspace; the eigenvectors with the smallest eigenvalues span the sub-dominant subspace.

Note that the first two steps do not depend on the subspace dimension. Applying the transformation matrix is "beamforming" with the eigenvectors of $\mathbf{R}_{\mathcal{K}}$, so $\mathbf{P}_{\mathrm{E}}$ is commonly referred to as an "eigenbeam" transformation. Often, the most difficult step in computing the eigenbeam transformation is integrating to form $\mathbf{R}_{\mathcal{K}}$.

5.3.3 Analysis

The procedure in Section 5.3.2 constructs orthonormal transformations to the optimal least-squares subspaces of any chosen dimension. Section 5.3.2 only proves that these subspaces minimize the least-squares criterion; it does not indicate how much dimension reduction is possible.

This subsection analyzes the behavior of eigenbeam subspaces and demonstrates a significant reduction in dimension with near-optimal performance. First, it characterizes the error behavior of eigenbeam transformations and provides a rule for choosing an acceptable subspace dimension. Second, it shows that the required dimension is often very small and grows linearly with region size and frequency. Third, it provides an example of the performance improvements achieved with subspace adaptive processing.

The integrated squared error from Equation 5.3.4 provides a natural metric for choosing the dimension of eigenbeam subspaces. Just as the length of a filter is chosen to satisfy a maximum error criterion, the dimension of an eigenbeam subspace is easily chosen by the integrated error. Because the eigenbeams span the dominant subspace of $\mathbf{R}_{\mathcal{K}}$, the integrated error of a D-dimensional eigenbeam subspace is determined by the sub-dominant eigenvalues of $\mathbf{R}_{\mathcal{K}}$:

$$\varepsilon(D) = \sum_{i=D+1}^{D_0} \lambda_i^2 \left(\mathbf{R}_{\mathcal{K}}\right). \tag{5.3.12}$$

Recall that D_0 is the dimension of the full space, or the data dimension. The function $\varepsilon(D)$ is easily evaluated from the existing singular value decomposition. A more

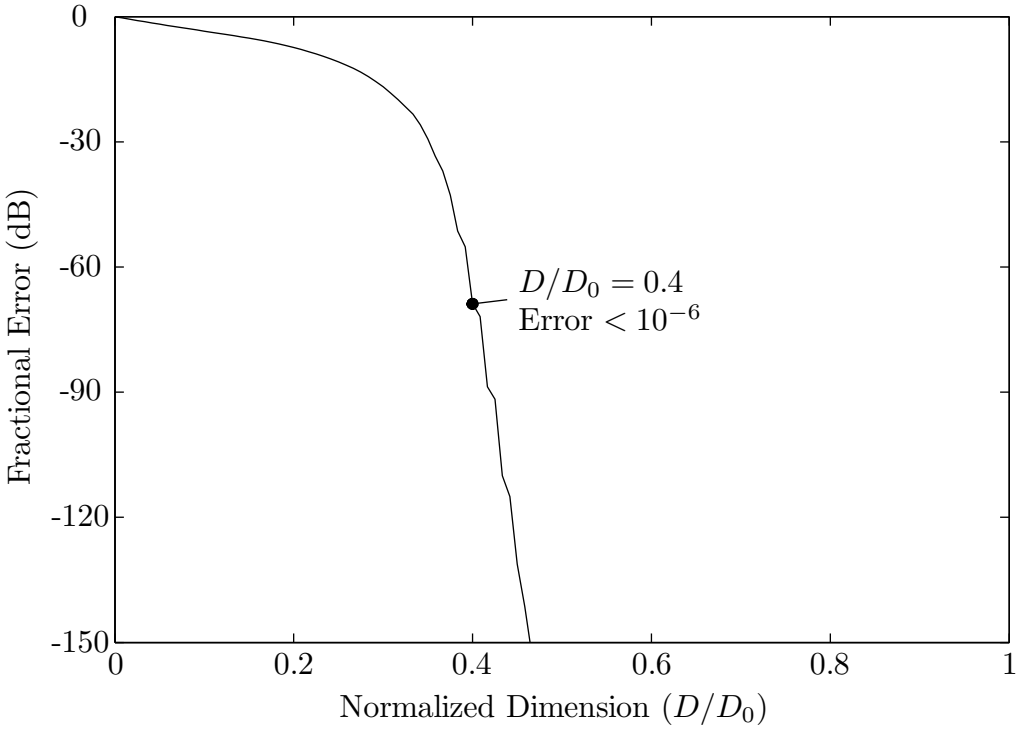

Figure 5.3.1: Integrated error versus eigenbeam dimension

universal quantity is the relative, or fractional, error formed by the ratio

$$\begin{aligned} \eta(D) &\triangleq \frac{\varepsilon(D)}{\varepsilon(0)} \\ &= \frac{\sum_{i=D+1}^{D_0} \lambda_i^2 \left(\mathbf{R}_{\mathcal{K}}\right)}{\sum_{i=1}^{D_0} \lambda_i^2 \left(\mathbf{R}_{\mathcal{K}}\right)}. \end{aligned} \tag{5.3.13}$$

By definition, the fractional error lies in the range $0 \leq \eta(D) \leq 1$ and is monotonically decreasing with D. Figure 5.3.1 provides an example of fractional error (in decibels) versus subspace dimension. The curve in Figure 5.3.1 corresponds to an $N = 30$ element VSA at $f = 5/7 f_d$; the region $\mathcal{K}$ corresponds to the entire visible region in u-space. Figure 5.3.1 reveals that the fractional error exhibits threshold behavior at a critical dimension that is much less than D_0. Beyond this critical dimension, $\eta(D)$ decreases rapidly. A rule-of-thumb for choosing the subspace dimension is to select

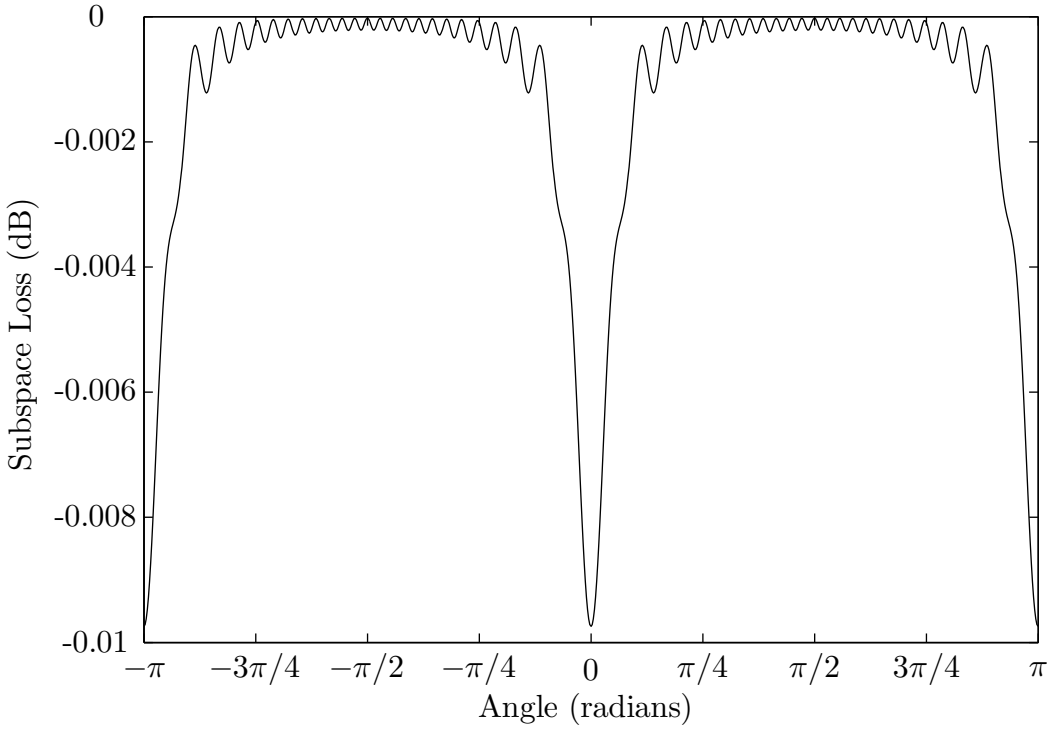

Figure 5.3.2: Loss incurred with an eigenbeam subspace: $D/D_0 = 0.4$

the minimum dimension that reduces the fractional error below a given threshold, or

$$D_{\mathrm{E}} \triangleq \inf \{D \mid \eta(D) < \epsilon\} \tag{5.3.14}$$

for some $\epsilon \ll 1$. The threshold behavior of $\eta(D)$ implies that *the particular choice of ϵ has little effect on the resulting dimension, D_E.* The remainder of this section assumes a conservative threshold of $\epsilon = 10^{-6}$. Applying this threshold to the example in Figure 5.3.1 gives $D/D_0 = 0.4$, a 60% reduction in dimension. To visualize the resulting errors, consider the "subspace loss" of a manifold vector $\mathbf{v}$ as the ratio

$$\frac{\mathbf{v}^H \mathbf{P}\mathbf{P}^H \mathbf{v}}{\mathbf{v}^H \mathbf{v}} \leq 1. \tag{5.3.15}$$

As its name implies, the subspace loss is always less than unity when $\mathbf{P}$ is orthonormal. Figure 5.3.2 confirms that the subspace loss is negligible (< 0.01 dB) for this example. The figure also reveals that subspace loss increases near array endfire.

Eigenbeam subspaces yield substantial dimension reduction when applied to vector-sensor arrays. Below the design frequency, the required dimension approximately equals the number of critical beams required to cover the region. Critical beams are conventional beams spaced at the Rayleigh resolution limit, *i.e.* at the peak-to-null width. In u-space, the VSA peak-to-null width is

$$(\Delta u)_{\mathrm{PN}} = \frac{2}{N} \cdot \left(\frac{f_d}{f} \right), \tag{5.3.16}$$

so the number of critical beams required to cover some region Δu on both sides of the array is

$$\begin{aligned} B_{\mathrm{crit}} &= 2 \cdot \frac{(\Delta u)}{(\Delta u)_{\mathrm{PN}}} \\ &= 2N \cdot \left(\frac{f}{f_d} \right) \cdot \left(\frac{\Delta u}{2} \right). \end{aligned} \tag{5.3.17}$$

The additional factor of two accounts for the two sides of the array. Figure 5.3.3 illustrates the eigenbeam dimension versus frequency and region size for a very long array ($N = 201$). The subspace dimension is well approximated by the number of critical beams, or

$$\begin{aligned} \left(\frac{D}{D_0} \right) &\approx \left(\frac{B_{\mathrm{crit}}}{D_0} \right) \\ &= \frac{1}{2} \cdot \left(\frac{f}{f_d} \right) \cdot \left(\frac{\Delta u}{2} \right). \end{aligned} \tag{5.3.18}$$

Equation 5.3.18 is written in terms of normalized quantities to illustrate its simple bi-linear form. The subspace dimension is often slightly greater than the approximation, depending on the threshold, but the approximation becomes tighter as the number of elements increases. Equation 5.3.18 is not intended to replace the numerical method in Equation 5.3.14, only to illustrate its dependencies. The power of subspace processing is evident in Equation 5.3.18 and Figure 5.3.3: compared to element-space, eigenbeam processing reduces the dimension by at least half and often much more.

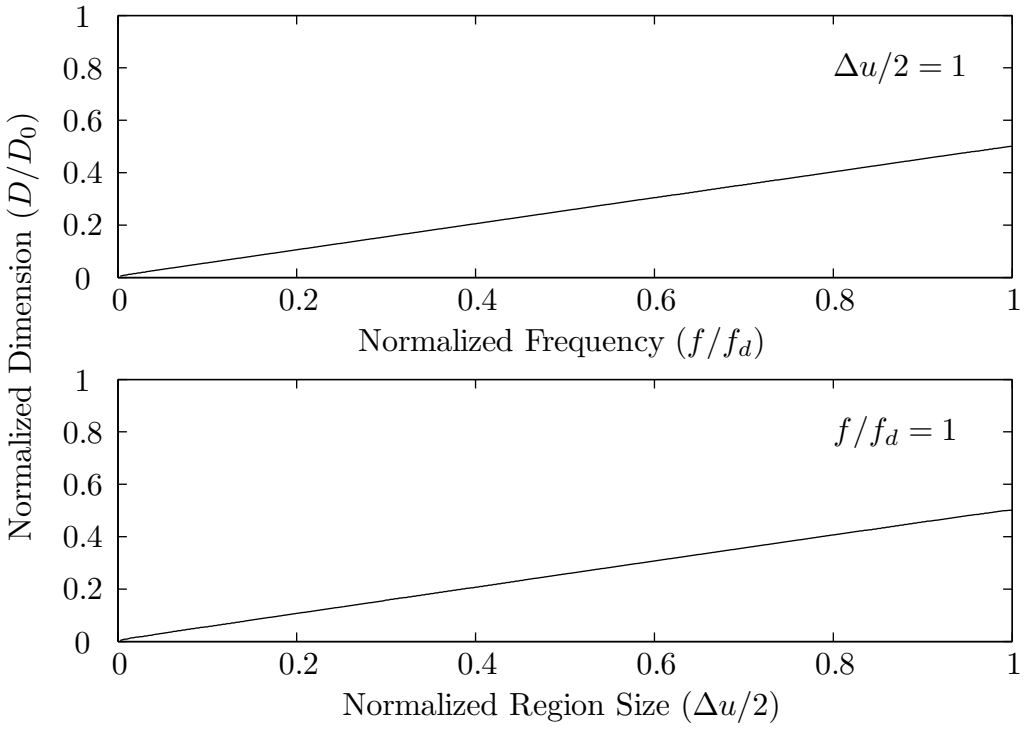

Figure 5.3.3: Eigenbeam dimension versus frequency and region size

The dimension reduction achieved with eigenbeam processing allows for dramatic improvements when the training is limited. As mentioned in Section 1.4.3, the number of data snapshots is often too few for element-space processing. Figure 5.3.4 illustrates one such scenario with an $N = 30$ element vector-sensor array at $f = 5/7 f_d$. The simulation involves white noise and four sources at $\phi \approx \{-0.73\pi,\ -0.39\pi,\ 0.27\pi,\ 0.61\pi\}$ radians with array signal-to-noise ratios ASNR $= \{6, 12, 18, 24\}$ dB, respectively. The two sources nearest endfire leak into beams on the opposing side of the array; note the small "backlobe" peaks near $\phi \approx -0.27\pi$ and $\phi \approx 0.73\pi$. The eigenbeam subspace is the same subspace illustrated in Figures 5.3.1 and 5.3.2 with the same dimension, $D = 48$. The number of snapshots is $K = 3 \times D$, enough to guarantee the eigenbeam covariance is well-estimated [9]. The ABF power estimates are non-Gaussian random variables, so their median values and the 95% confidence region are indicated as determined from 10,000 Monte-Carlo trials. The element-space processor is severely affected by the low sample support, yielding a large power bias and unreliable esti-

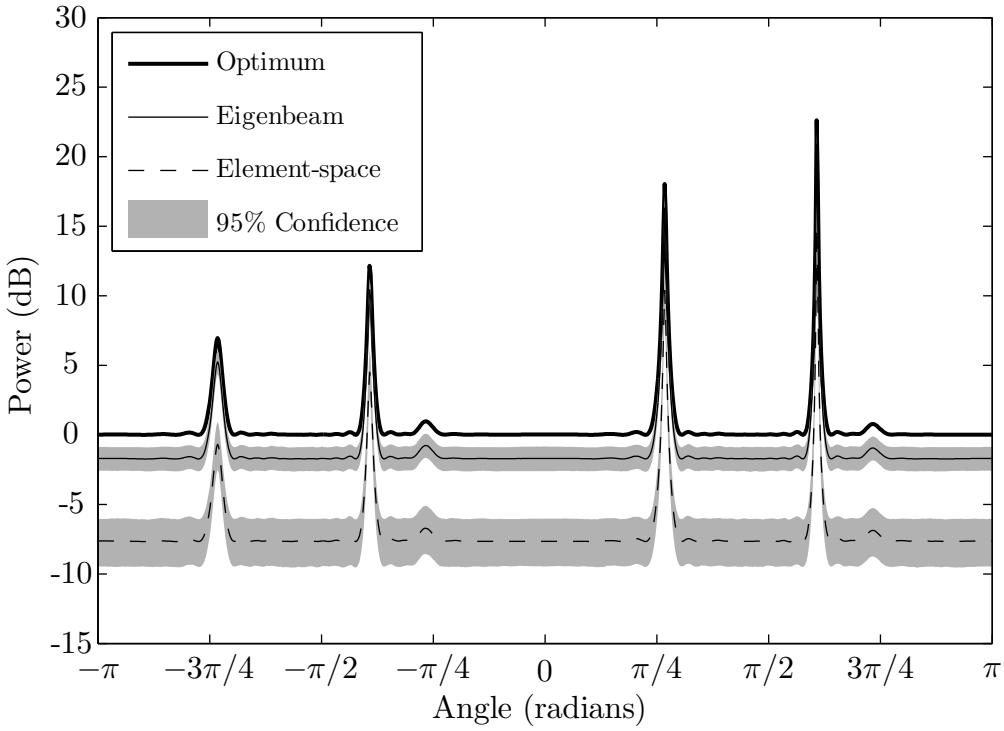

Figure 5.3.4: ABF comparison with $K = 3 \times D$ snapshots

mates. By contrast, the eigenbeam processor is near optimum. Eigenbeam processing reduces the power bias by ≈ 6 dB and yields more reliable power estimates. The comparison in Figure 5.3.4 is impossible in many scenarios because the training is too short for element-space processing. For example, Figure 1.4.3 predicts only $K \approx 108$ available snapshots in the scenario described above. Eigenbeam adaptive processing is well-conditioned with this support, but element-space adaptive processing is impossible without modification.

One benefit of dimension reduction is an improvement in output bias, or a diminished "loss." Sample matrix inverse (SMI) processing estimates a covariance matrix from a finite number of snapshots. The output power of the processor is biased low because the covariance estimate is imperfect [9]. The power bias (in decibels) for a D-dimensional problem with $K > D$ snapshots is

$$10 \log_{10} \left(\frac{K+1}{K+2-D} \right). \tag{5.3.19}$$

Reducing the problem dimension from D_0 to D improves this bias by

$$10 \log_{10} \left(\frac{K + 2 - D}{K + 2 - D_0} \right). \tag{5.3.20}$$

The improvement is most dramatic when the dimension is significantly reduced and the number of snapshots is limited, as illustrated in Figure 5.3.4.

The eigenbeam techniques developed in this chapter have important applications in sector and local subspace processing. This chapter primarily analyzes eigenbeam processing over all of u-space. It is possible to segment u-space into different regions, or "sectors," and perform eigenbeam processing within each sector. It is also possible to process each beam in its own local subspace. Partitioning u-space into multiple regions produces many adaptive processing problems, each of which has smaller dimension than the original. Sector and local subspace techniques are discussed further in [5, §3.10].

5.4 Differential Geometry of Subspace Design

The previous section describes the performance improvements possible with subspace processing. A key result in the discussion is the observation in Equation 5.3.18 that the eigenbeam dimension grows linearly with both frequency and region size and remains smaller than the full data dimension. Although a detailed discussion is beyond the scope of this thesis, Equation 5.3.18 hints at the special structure of the vector-sensor array manifold. Parameterized by azimuth angle, the manifold $\mathbf{v}(\phi)$ represents a smooth curve on a radius-$\sqrt{2N}$ sphere in the high dimensional space $\mathbb{C}^{4N}$. Although $\mathbf{v}(\phi)$ moves around the sphere (as indicated by the sidelobes in the beampattern), it stays almost exclusively within the low-dimensional hyperplane given by the eigenbeam subspace. In this sense, $\mathbf{v}(\phi)$ moves around the sphere near the "equator." The curve appears flat on a small scale, but its dimension grows linearly as the scale increases (see Figure 5.3.3). Subspace design is closely tied to this geometric picture and other concepts from differential geometry.

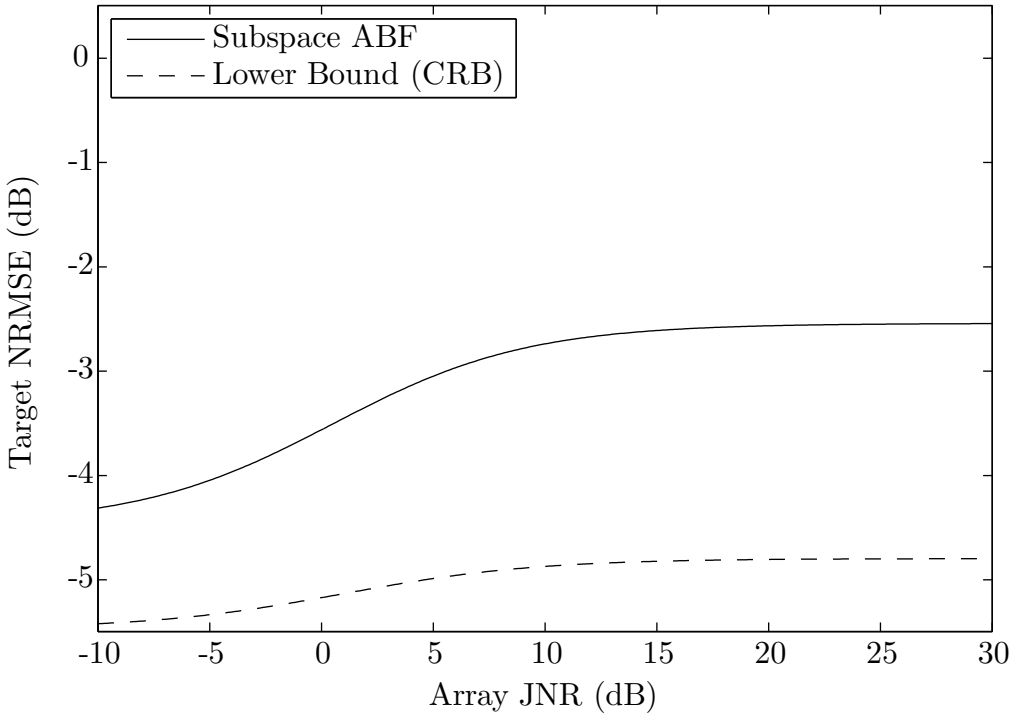

Figure 5.5.1: Efficiency of an example subspace processor

5.5 Performance of Adaptive Processing

The adaptive processing techniques described in this chapter approach the performance predicted in Chapter 3 under many circumstances. Unlike the fixed weights designed in Chapter 4, adaptive processors perform well even in the presence of strong interference. Figure 5.5.1 illustrates this behavior with the standard $N = 10$ element VSA steered to $\phi_0 = -\pi/4$ at frequency $f = 5/7 f_d$. The NRMSE metric and Cramér-Rao bound are discussed in Section 3.2. The eigenbeam processor displayed in Figure 5.5.1 utilizes $D = 19$ dimensions and $K = 50$ snapshots. The subspace ABF is not unbiased, but the Cramér-Rao bound is still helpful. Comparing Figures 5.5.1 and 4.4.1 reveals that the left/right performance of adaptive processing is substantially better than fixed weight processing when the interference is strong. Furthermore, advanced adaptive processors likely achieve better NRMSE performance than the simple SMI processor shown in the figure. Figure 5.3.4 reveals that adaptive processing also increases resolution, an improvement not captured by the NRMSE metric.

Chapter 6

Conclusion

The analyses and techniques presented in this thesis enable substantial improvements in vector-sensor array processing. Building on the fundamentals enumerated in Chapter 2, the thesis bounds vector-sensor performance and describes near-optimal processing techniques.

6.1 Summary of Processing Improvements

Each chapter of the thesis focuses on improving a different branch of vector-sensor array processing as described in Section 1.5. The performance bounds developed in Chapter 3 tie the processing techniques together in Figures 4.4.1 and 5.5.1. These figures quantify the left/right rejection of both techniques, but they do not give a sense of overall performance. Figure 6.1.1 illustrates the improved VSA processing achieved by this work. The four-source scenario described in Section 5.3.3 is simulated with the standard $N = 10$ element vector-sensor array at $f = 5/7 f_d$. The number of observations is $K = 50$.

The top axis in Figure 6.1.1 displays conventional VSA processing. With conventional processing, strong sources on one side ($\phi > 0$) of the array have high sidelobes and backlobes that interfere with sources on the other side ($\phi < 0$). False peaks at $\phi \approx -\pi/4$ and $\phi \approx -5\pi/8$ make detecting the true sources difficult. It is impossible to determine the location and number of sources from conventional processing alone.

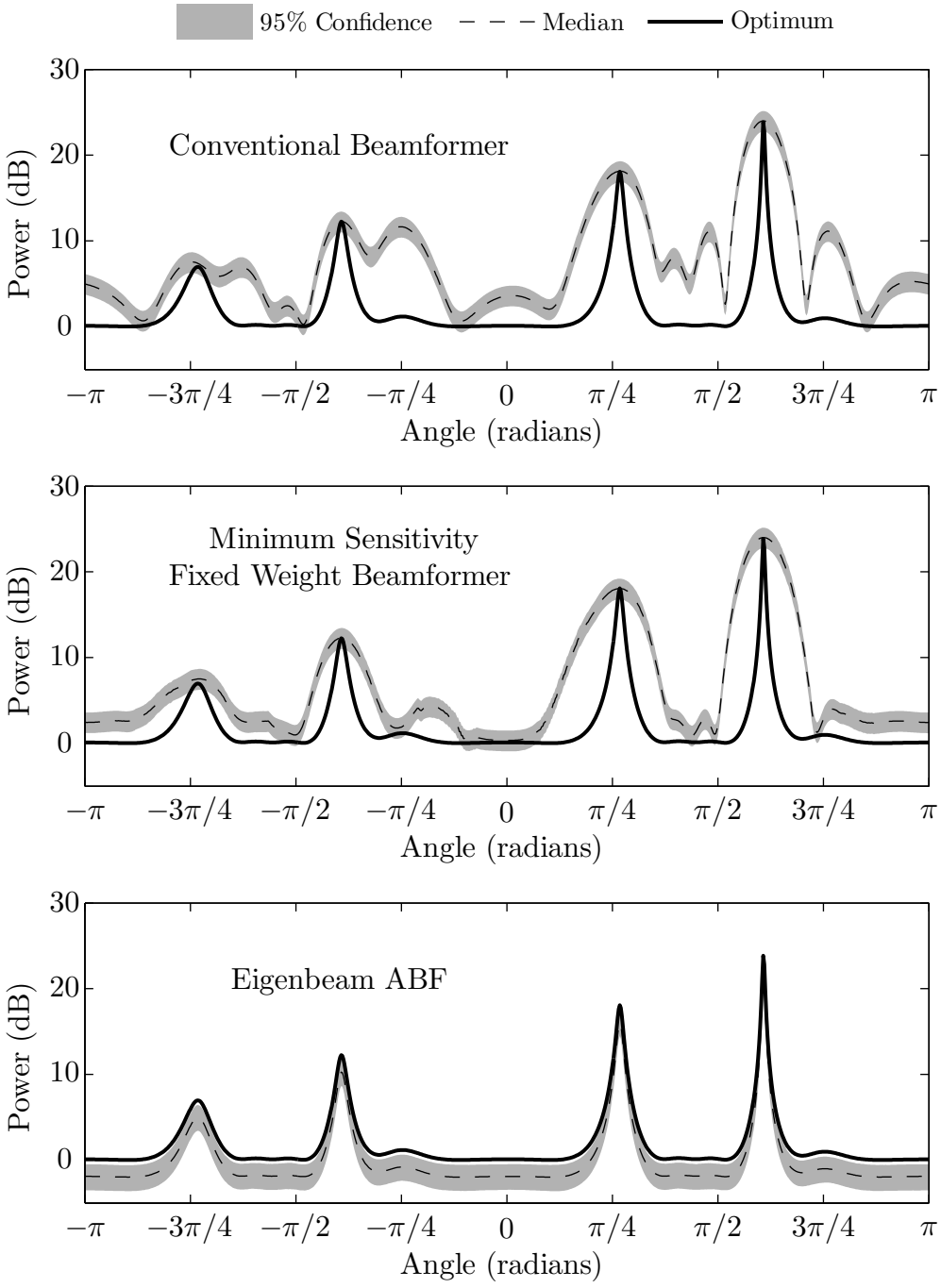

Figure 6.1.1: Example of improved VSA processing

The center axis in Figure 6.1.1 displays the output of a beamformer using the minimum sensitivity fixed weights described in Section 4.3.3. The weights have a maximum sidelobe level of -25 dB. Minimum sensitivity weights reject sidelobe interference without sacrificing much resolution, so the four true sources are clearly visible without strong false peaks. The left/right resolution is reasonably close to the optimum result. The limits of fixed weights are also illustrated: weak sidelobes are visible on the strongest source and the array resolution is far from optimal.

The bottom axis in Figure 6.1.1 displays the eigenbeam adaptive processing described in Chapter 5. Adaptive processing in the low-dimensional ($D = 19$) subspace yields fast convergence and near-optimum results. Similar processing in element-space produces a biased and unreliable output. Eigenbeam adaptive processing reduces the left/right ambiguities more than fixed weight processing, but the most significant improvement is the increased resolution.

6.2 Future Work in VSA Processing

The last chapter of this thesis is not the final chapter in vector-sensor array research. The doors opened by this work lead to many unexplored and interesting areas within array processing:

- *Extending the convex optimization algorithms described in Chapter 4 to arbitrary arrays and stochastic models.* A generalized beampattern design algorithm could be easily constructed around a second-order cone solver (see Section 4.3.4). Such an algorithm would be a powerful tool for arrays that are nonlinear, mismatched, or both.

- *Deriving optimal "local" subspaces for each beam.* The eigenbeam subspaces in Chapter 5 support processing of signals and interference within a given region. One alternative problem is designing a subspace for each beam. A weighted least squares approach yields a modified eigenbeam technique. The dimension of such

local subspaces is likely to be small and may only depend weakly on the array length, but computational load and processing artifacts may be problematic.

A number of interesting topics specific to vector-sensor arrays also arise:

- *Extending this work to resolve pressure-sensor grating lobes.* Directional information allows for unambiguous vector-sensors array processing at all angles and frequencies. This work focused on resolving left/right pressure ambiguities. The same techniques may resolve pressure-sensor spatial aliasing above the array design frequency (see [5, §2.4] and [1]).

- *Matched field processing with vector-sensor arrays.* The directional measurements provided by a horizontal vector-sensor array allow limited vertical resolution. Leveraging this vertical resolution could reduce the problematic ambiguities in matched field processing (see [21]).

- *Computationally efficient adaptive processing.* The point null approach in Sections 4.1 and 4.2 is easily transformed into an adaptive sidelobe canceller. Adapting only to the backlobe and grating lobes requires little computation but provides less benefit than fully-adaptive beamforming.

The techniques developed in this thesis provide a foundation for further research and indicate the bright future ahead for acoustic vector-sensor arrays.

Appendix A

Supplemental Material

A.1 Symmetric Noise Distributions

This appendix briefly proves a statement made in Section 3.1: for "left/right symmetric" noise distributions, the probability of error is a function of *only* the number of snapshots and the K-L divergence. The formal definition of "left/right symmetric" is that

$$\mathbf{v}_0^H \mathbf{R}_n^{-1} \mathbf{v}_0 = \mathbf{v}_1^H \mathbf{R}_n^{-1} \mathbf{v}_1 \tag{A.1.1}$$

for all azimuth angles. Recall that $\mathbf{v}_0$ and $\mathbf{v}_1$ are replica vectors for opposing sides of a linear vector-sensor array. This definition agrees with intuition and implies the only condition necessary for this proof:

$$\begin{aligned} |\mathbf{R}_1 \mathbf{R}_0^{-1}| &= \frac{|\mathbf{R}_n + \sigma^2 \mathbf{v}_1 \mathbf{v}_1^H|}{|\mathbf{R}_n + \sigma^2 \mathbf{v}_0 \mathbf{v}_0^H|} \\ &= \frac{|\mathbf{R}_n (\mathbf{I} + \sigma^2 \mathbf{R}_n^{-1} \mathbf{v}_1 \mathbf{v}_1^H)|}{|\mathbf{R}_n (\mathbf{I} + \sigma^2 \mathbf{R}_n^{-1} \mathbf{v}_0 \mathbf{v}_0^H)|} \\ &= \frac{1 + \sigma^2 \mathbf{v}_1^H \mathbf{R}_n^{-1} \mathbf{v}_1}{1 + \sigma^2 \mathbf{v}_0^H \mathbf{R}_n^{-1} \mathbf{v}_0} \\ &= 1. \end{aligned} \tag{A.1.2}$$

For any noise distribution, Section 3.1 shows that the probability of error only depends on the eigenvalues of the matrix $\mathbf{R}_1\mathbf{R}_0^{-1}$ and the number of snapshots K. Applying the matrix inversion lemma gives

$$\begin{aligned}
\mathbf{R}_1\mathbf{R}_0^{-1} &= (\mathbf{R}_n + \sigma^2\mathbf{v}_1\mathbf{v}_1^H)(\mathbf{R}_n + \sigma^2\mathbf{v}_0\mathbf{v}_0^H)^{-1} \\
&= (\mathbf{R}_n + \sigma^2\mathbf{v}_1\mathbf{v}_1^H)(\mathbf{R}_n^{-1} - \alpha\mathbf{R}_n^{-1}\mathbf{v}_0\mathbf{v}_0^H\mathbf{R}_n^{-1}) \\
&= \mathbf{I} - \alpha\mathbf{v}_0\mathbf{v}_0^H\mathbf{R}_n^{-1} + \sigma^2\mathbf{v}_1\mathbf{v}_1^H\mathbf{R}_n^{-1} - \sigma^2\alpha\beta\mathbf{v}_1\mathbf{v}_0^H\mathbf{R}_n^{-1} \qquad \text{(A.1.3)}
\end{aligned}$$

where

$$\alpha \triangleq (\sigma^{-2} + \mathbf{v}_0^H\mathbf{R}_n^{-1}\mathbf{v}_0)^{-1} \qquad \text{(A.1.4)}$$

$$\beta \triangleq \mathbf{v}_1^H\mathbf{R}_n^{-1}\mathbf{v}_0. \qquad \text{(A.1.5)}$$

The matrix in Equation A.1.3 can be written as the identity matrix plus a matrix with rank no greater than two. Thus, it has no more than two non-unity eigenvalues. Because the determinant is one, either the two non-unity eigenvalues are a reciprocal pair or all eigenvalues are unity. In either case, the relationship between the two (possibly) non-unity eigenvalues means that the trace of the matrix $\mathbf{R}_1\mathbf{R}_0^{-1}$ fully specifies its eigenvalues. Therefore, the trace together with the number of snapshots specifies the probability of error. Recall that the K-L divergence between to zero-mean Wishart distributions is

$$\mathrm{D}(p_0||p_1) = \frac{K}{2}\left[\ln\frac{|\mathbf{R}_1|}{|\mathbf{R}_0|} + \mathrm{tr}\left(\mathbf{R}_1^{-1}\mathbf{R}_0\right) - 4N\right]. \qquad \text{(A.1.6)}$$

The determinants are equal for left/right symmetric noise distributions, so the K-L divergence with the number of snapshots also characterizes the probability of error.

Although not required for the proof, notice that the K-L divergence has a simple form similar to Equation 3.1.29 for *any* left/right symmetric noise distribution. This is shown by applying the matrix inversion lemma to A.1.6, expanding terms, and evaluating the trace.

A.2 Weights and the Array Manifold

Array processing often deals with array manifolds exhibiting a given property. That is, every replica vector $\mathbf{v}(\boldsymbol{\Theta})$ that forms the array manifold exhibits the same property. It seems natural that any weight $\mathbf{w}$ applied to the manifold might inherit such a property, *e.g.*

P1 Conjugate symmetric weights may be sufficient for a conjugate symmetric manifold.

P2 Weights with element-wise linear phase may be sufficient for a manifold whose replicas have element-wise linear phase.

As trivial and intuitive as these assumptions seem, they are difficult to prove and some are false. An incorrect assumption is overly restrictive and may lead to suboptimal weights. This appendix proves that **P1** is generally true and suggests that **P2** is generally not. More importantly, it provides weak sufficient conditions for any property to be transferred from the array manifold to the weights.

Throughout the discussion, "$\mathbf{P}$" denotes the property or the set of vectors exhibiting the property. The following set of conditions guarantees that a property, $\mathbf{P}$, of the array manifold is transferred a weight:

C1 The weight is a solution to a convex optimization problem.

C2 The gradients of all objective and constraint functions exhibit $\mathbf{P}$ at any point exhibiting $\mathbf{P}$. Formally, for any gradient or constraint function $f(\mathbf{w})$, this requires $\left[\nabla_{\mathbf{w}^H} f(\mathbf{w})\right]_{\mathbf{w}_0} \in \mathbf{P} \quad \forall \mathbf{w}_0 \in \mathbf{P}$.

C3 The property is preserved under real, linear combination. Formally, if $\mathbf{x}_1 \in \mathbf{P}$ and $\mathbf{x}_2 \in \mathbf{P}$ then $a\mathbf{x}_1 + b\mathbf{x}_2 \in \mathbf{P}$ for all real a and b.

Note that the first condition applies only to the problem, the second applies to both the problem and the property, and the third applies only to the property.

The above conditions are deemed "weak" because none is overly restrictive in practice. The first condition applies to the most useful set of problems, those for which

optimality is easily proved and local extrema do not exist. The third condition, **C3**, is weak because any property not satisfying **C3** forms a concave set. For such properties, the original (convex) optimization problem under the additional constraint $\mathbf{w} \in \mathbf{P}$ is non-convex and likely more difficult. Thus, although the solution may exhibit $\mathbf{P}$, this information is not necessarily helpful. The second condition is weak because it is satisfied by many problems and properties of interest when **C3** is satisfied. This includes the beampattern synthesis, optimum beamforming, and weight design problems studied in this thesis.

The proof consists of showing that there exists a sequence of weights $\{\mathbf{w}_0, \mathbf{w_1}, \ldots\}$, $\mathbf{w}_n \in \mathbf{P}$, that converges to a global optimum. Under the conditions **C1**–**C3** above, the proof is trivial thanks to the convergence of various first-order optimization algorithms. In solving the canonical convex optimization problem

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{w}) \\ \text{subject to} \quad & g_n(\mathbf{w}) \leq 0 \quad n = 1, 2, \ldots, N \\ & h_m(\mathbf{w}) = 0 \quad m = 1, 2, \ldots, M \end{aligned} \quad , \tag{A.2.1}$$

a number of first-order optimization algorithms with proven convergence iterate by taking steps in the direction opposite a generalized gradient. These algorithms include first-order variants of non-differentiable exact penalty and Lagrangian methods. In this case, the set of generalized gradients $\widetilde{\nabla}_{\mathbf{w}^H}$ is contained in the convex hull formed from the individual gradients of the objective and constraint functions:

$$\begin{aligned} \widetilde{\nabla}_{\mathbf{w}^H} \quad \subset \quad & \Big\{ \lambda \nabla_{\mathbf{w}^H} f(\mathbf{w}) + \textstyle\sum_{n=1}^{N} \alpha_n \nabla_{\mathbf{w}^H} g_n(\mathbf{w}) + \sum_{m=1}^{M} \beta_m \nabla_{\mathbf{w}^H} h_m(\mathbf{w}) \quad \Big| \\ & \lambda + \textstyle\sum_{n=1}^{N} \alpha_n + \sum_{m=1}^{M} \beta_m = 1, \quad \lambda \geq 0, \quad \alpha_n \geq 0, \quad \beta_m \geq 0 \Big\}. \end{aligned} \tag{A.2.2}$$

The conditions **C2** and **C3** imply that the step direction exhibits $\mathbf{P}$ because it is a real, linear combination of objective and constraint gradients exhibiting $\mathbf{P}$. For the algorithms mentioned, condition **C1** guarantees convergence starting from any point. Choosing a starting point $\mathbf{w}_0 \in \mathbf{P}$ without loss of generality, condition **C3** guarantees that a step along any generalized gradient also satisfies $\mathbf{P}$. Thus, every weight in the

sequence of iterates satisfies **P**. Because the sequence converges to a global optimum, that global optimum must satisfy **P**.

The above proof deserves several comments. First, although the proof involves first-order algorithms moving along generalized gradients, this is only to show that a global optimum exists exhibiting **P**; it does not constrain the type of algorithm used in practice. Second, if the problem is strictly convex (**C1** is strengthened), the unique optimum is proved to exhibit **P**. Third, if the problem is non-convex (**C1** is weakened), there exist local optima exhibiting **P** to which the first-order algorithms above will converge.

Testing the two properties in the first paragraph, **P1** and **P2**, for the conditions reveals that **P1** satisfies **C3** but **P2** does not. Thus, **P1** is generally true but **P2** need not be.

Appendix B

Nomenclature

B.1 Acronyms

Acronym	Description
ABF	Adaptive Beamforming
ASNR	Array Signal to Noise Ratio
CBF	Conventional Beamforming
CRB	Cramér-Rao Bound
DOA	Direction of Arrival
FIR	Finite Impulse Response
JNR	Jammer to Noise Ratio
LP	Linear Program
MSE	Mean Squared Error
MVDR	Minimum Variance Distortionless Response
NRMSE	Normalized Root Mean Squared Error
PCML	Physically Constrained Maximum Likelihood
PSA	Pressure-sensor Array
RMSE	Root Mean Squared Error
SINR	Signal to Interference-Plus-Noise Ratio
SMI	Sample Matrix Inverse
SNR	Signal to Noise Ratio
VSA	Vector-sensor Array

B.2 Notation

Notation	Description	Example
a	Scalar variable	Eqn. 3.1.31
$\mathbf{a}$	Vector variable	Eqn. 1.7.1
$\mathbf{a}^H$	Conjugate (or Hermitian) transpose	Eqn. 2.1.1
$\mathbf{a}^*$	Conjugation	Eqn. 3.2.16
$\mathbf{a}^T$	Transpose	Eqn. 1.7.1
$\mathbf{a} \odot \mathbf{b}$	Element-wise (or Hadamard) product	Eqn. 2.4.11
$\mathbf{a} \otimes \mathbf{b}$	Tensor (or Kronecker) product	Eqn. 4.2.7
$[\mathbf{a}\ ,\ \mathbf{b}]$ or $[\mathbf{a}\ \ \mathbf{b}]$	Horizontal concatenation	Eqns. 1.7.1 or 3.1.2
$\mathrm{D}(p_0 \| \| p_1)$	Kullback-Leibler divergence	Eqn. 3.1.24
$\mathbf{A} \succeq 0$	Matrix $\mathbf{A}$ is positive semidefinite	Eqn. 3.2.1
$[\mathbf{A}]_{ij}$	i,j^{th} element of the matrix $\mathbf{A}$	Eqn. 3.2.2
$\mathrm{E}\{x\}$	Expectation of random variable x	Eqn. 4.4.3
$a \triangleq 0$	Definition of a	Eqn. 5.3.1
$\mathbb{R}^N$ or $\mathbb{C}^N$	Real or complex N-dimensional space	Sec. 5.1
$\mathbb{G}(D, D_0)$	Grassmann manifold	Sec. 5.1
$\langle \mathbf{a},\ \mathbf{b} \rangle$	Inner product	Eqn. 5.2.1
$\lambda_i(\mathbf{R})$	i^{th} largest eigenvalue of $\mathbf{R}$	Eqn. 5.3.9
$\mathrm{tr}(\mathbf{A})$	Trace of the matrix $\mathbf{A}$	Eqn. 5.3.7
$\lvert\mathbf{A}\rvert$	Determinant of the matrix $\mathbf{A}$	Eqn. 3.1.6
$f^{(n)}(x)$	n^{th} derivative of $f(x)$	Eqn. 2.4.15
$\mathrm{diag}(\mathbf{A})$	Main diagonal of $\mathbf{A}$	Eqn. 3.2.17
$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$	Vertical concatenation	Eqn. 1.7.8

Bibliography

[1] J. P. Kitchens. Acoustic vector-sensor array performance. Master's thesis, Massachusetts Institute of Technology, June 2008.

[2] A. Nehorai and E. Paldi. Acoustic vector-sensor array processing. *IEEE Transactions on Signal Processing*, 42(9):2481–2491, September 1994.

[3] H. Cox and H. Lai. Performance of Line Arrays of Vector and Higher Order Sensors. *Conference Record of the Forty-First Asilomar Conference on Signals, Systems, and Computers*, pages 1231–1236, November 2007.

[4] H. Lai. *Optimum and Adaptive Processing of Acoustic Vector and Higher-order Sensors*. PhD thesis, George Mason University, Spring 2008.

[5] H. L. Van Trees. *Optimum Array Processing. Part IV of Detection, Estimation, and Modulation Theory*. John Wiley & Sons, Inc., New York, NY, 2002.

[6] M. Hawkes and A. Nehorai. Effects of sensor placement on acoustic vector-sensor array performance. *IEEE Journal of Oceanic Engineering*, 24(1):33–40, January 1999.

[7] B. A. Cray and A. H. Nuttall. Directivity factors for linear arrays of velocity sensors. *Journal of the Acoustical Society of America*, 110(1):324–331, July 2001.

[8] A. B. Baggeroer and H. Cox. Passive sonar limits upon nulling multiple moving ships with large aperture arrays. *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, 1:103–108, October 1999.

[9] L. Brennan, I. Reed, and J. Mallat. Rapid convergence rate in adaptive arrays. *IEEE Transactions on Aerospace and Electronic Systems*, 10(6):853–863, November 1974.

[10] S. M Kogon. Eigenvectors, diagonal loading and white noise gain constraints for robust adaptive beamforming. *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems, and Computers*, 2:1853–1857, November 2003.

[11] H. Cox and R. Pitre. Robust DMR and multi-rate adaptive beamforming. *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems, and Computers*, 1:920–924, November 1997.

[12] A. L. Kraay and Baggeroer A. B. A physically constrained maximum-likelihood method for snapshot-deficient adaptive array processing. *IEEE Transactions on Signal Processing*, 55(8):4048–4063, August 2007.

[13] M. Hawkes and A. Nehorai. Acoustic vector-sensor beamforming and capon direction estimation. *IEEE Transactions on Signal Processing*, 46(9):2291–2304, September 1998.

[14] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt. *Computational Ocean Acoustics*. Springer-Verlag New York, Inc., Fifth Avenue, New York, NY 10010, 2000.

[15] A. J. Poulsen and A. B. Baggeroer. Vector sensor array sensitivity and mismatch: generalization of the Gilbert-Morgan formula. In *Fourth Joint Meeting of the Acoustical Society of America and the Acoustical Society of Japan*, December 2006.

[16] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. John Wiley and Sons, New York, NY, 1972.

[17] L. L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley Publishing Company, Reading, MA, 1991.

[18] A. H. Nuttall. Numerical evaluation of cumulative probability distribution functions directly from characteristic functions. Technical Report 1032, Navy Underwater Sound Laboratory, 1969.

[19] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, NY, 1 edition, 1991.

[20] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 1998.

[21] W. Xu. *Performance Bounds on Matched-Field Methods for Source Localization and Estimation of Ocean Environmental Parameters*. PhD thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, June 2001. MIT/WHOI Joint Program.

[22] I. Richards and H. Youn. *Theory of Distributions: a non-technical introduction*. Cambridge University Press, New York, NY, 1990.

[23] J. W. Adams. FIR Digital Filters with Least-Squares Stopbands Subject to Peak-Gain Constraints. *IEEE Transactions on Circuits and Systems*, 39(4):376–388, April 1991.

[24] H. Lebret and S. Boyd. Antenna array pattern synthesis via convex optimization. *IEEE Transactions on Signal Processing*, 45(3):526–532, March 1997.

[25] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[26] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[27] D. Wei. Non-convex optimization for the design of sparse FIR filters. *IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 117–120, 2009.

[28] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, 2 edition, 2006.

[29] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[30] P. Forster and G. Vezzosi. Application of spheroidal sequences to array processing. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 12:2268–2271, April 1987.

[31] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.