U. S. Naval Research Laboratory
Washington, DC 20375-5320

NRL/5340/FR--2022/2

Nonlinear Array Processing

Hatim F. Alqadah
Dan P. Scholnik
Jean de Graaf
Surveillance Technology Branch
Radar Division

April 6, 2022

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION

1. REPORT DATE 03-01-2021	**2. REPORT TYPE** NRL formal report	**3. DATES COVERED**	
		START DATE 01 October 2016	**END DATE** 31 December 2018

4. TITLE AND SUBTITLE
Nonlinear Array Processing

5a. CONTRACT NUMBER	**5b. GRANT NUMBER**	**5c. PROGRAM ELEMENT NUMBER** 61153N
5d. PROJECT NUMBER 53-1E27	**5e. TASK NUMBER** EW021-05-43	**5f. WORK UNIT NUMBER** 1E27

6. AUTHOR(S)
Hatim F. Alqadah, Dan P. Scholnik, and Jean de Graaf

7. PERFORMING ORGANIZATION / AFFILIATION NAME(S) AND ADDRESS(ES) U. S. Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320	**8. PERFORMING ORGANIZATION REPORT NUMBER** NRL/5340/FR--2022/2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research One Liberty Center 875 N. Randolph Street, Suite 1425 Arlington, VA 22203-1995	**10. SPONSOR / MONITOR'S ACRONYM(S)** ONR	**11. SPONSOR / MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: Distribution unlimited.

13. SUPPLEMENTAL NOTES

14. ABSTRACT

This report describes a theoretical and experimental study of a nonlinear array processing framework for sparse arrays that is based on using the difference coarray as a virtual uniform (Nyquist) array. The framework offers the potential to significantly increase the available degrees of freedom of a sparse array and therefore can lead to improved spatial source estimation and/or mitigation. The study examines the feasibility of incorporating such an approach as well as the underlying algorithms capable of exploiting such gains within a conventional monostatic radar system. Validation of the presented concepts is conducted through a 12 channel 3.4 GHz experimental testbed.

15. SUBJECT TERMS
Nonlinear Signal Processing, Coarray Processing, Adaptive Arrays

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT** SAR	**18. NUMBER OF PAGES** 92
a. REPORT Unclassified Unlimited	**b. ABSTRACT** Unclassified Unlimited	**c. THIS PAGE** Unclassified Unlimited		

19a. NAME OF RESPONSIBLE PERSON Hatim F. Alqadah	**19b. PHONE NUMBER** *(Include area code)* (202)-404-1926

This page intentionally left blank

CONTENTS

FIGURES

This page intentionally left blank

EXECUTIVE SUMMARY

Array signal processing for radar traditionally is focused on computing complex weight vectors to be applied linearly to the complex signals received at each array element in order to estimate signals from different directions. The linear framework is quite powerful and flexible, but does suffer from some inherent limitations for sparse arrays, as the degrees of freedom are tied directly to the number of elements. Nonlinear approaches involving second-order statistics have long been used to overcome these limitations when applied to receive-only applications such as direction finding and spectral estimation, but have not been seen as applicable to active sensing as signal coherence is lost.

In recent years, several new works appeared focused on directly processing the covariance, itself, as a virtual signal with support on the difference coarray. This contrasts with older techniques that augment the sparse covariance matrix in order to estimate the equivalent covariance of a filled array, and the papers promised (or at least implied) both higher resolution and greater effective degrees of freedom due to the double length of the coarray. These results prompted a renewed look at the role such techniques might play in both the traditional applications as well as in active sensing.

In this project, we considered the problem from several angles. The primary concern at the outset was sample support; at first glance, it appeared that the number of snapshots required for covariance estimation could prove incompatible with practical radar timelines. Cramer-Rao lower bound analysis for angle estimation showed that the problem was actually worse than expected; resolving a larger number of sources than physical array elements (which is the primary case of interest) requires a prohibitive number of data snapshots for active sensing. Thus, we ultimately focused on direction-of-arrival applications rather than radar, and on the application of single-snapshot covariance estimation approaches to compute an approximate covariance for a decorrelated coarray virtual signal. Our second objective was to compare the coarray-based approaches with the augmented covariance approaches to determine whether the recent works did, in fact, represent something fundamentally new. The results were mixed; the Cramer-Rao lower bound analysis showed that the coarray and augmented-covariance approaches have identical theoretical performance when it comes to estimating source directions. However, the algorithms applied to the coarray did tend to perform better in practice. For the resolution of two closely spaced targets, the longer apparent length of the coarray again appeared to produce better results. Finally, we designed a hardware testbed with which to perform proof-of-concept experiments to verify that the performance was robust to typical signal and hardware errors.

This page intentionally left blank

NONLINEAR ARRAY PROCESSING

1. INTRODUCTION

The 2017 Department of the Navy (DON) 30-year R&D plan outlines a future battle space that is heavily reliant on intelligent, autonomous and distributed sensing systems, for which radar, due to its all-weather and long-range detection capability, will continue to be an essential element. As operational demands increase in complexity, antenna arrays, rather than single physical apertures, will be expected to play a dominant role in next-generation radar systems. Antenna arrays provide the necessary degrees of freedom (DOF) to carry out the sophisticated spatial-processing functions to meet future challenges of the battle space. Examples include rapid target direction finding (DF), intelligent and robust mitigation against electromagnetic interference, and multifunction space-time capabilities providing simultaneous detection, ranging, tracking, and communication. In addition to increased design complexity, a major drawback of arrays in comparison to single-antenna systems is a substantial increase in size, weight, and power (SWAP) demands. As sensing platforms seek to become lighter, more agile, and more sustainable, reducing SWAP requirements of array-based radar systems is critical. This project was launched with the intent to develop algorithms to take advantage of sparse antenna arrays in the hopes of trading increased signal processing for substantially less physical hardware.

1.1 Background: Sparse Arrays

Two important parameters affecting the overall performance of an array are its physical size and the number of individual elements N that constitute the array. The size directly affects the resolution or how finely targets/sources can be resolved in space, while the choice of N defines the available DOF that can be leveraged for processing. Traditional arrays feature elements distributed uniformly across their extent using an interelement spacing on the order of half the wavelength of operation. This variant of the well-known Nyquist criterion assures that no ambiguities (grating lobes) arise within the array's field of view (FOV). Thus, for uniformly spaced arrays, the resolution performance and DOF are directly proportional to each other. Sparse (non-uniform) arrays break this linkage and violate the Nyquist criterion with the aim of using as few elements as possible in order to fill the desired extent. Fewer elements implies a loss in the direct DOF of the array, which limits the performance of conventional spatial signal-processing techniques. However, in some cases, a combination of proper element placement, second-order statistics, and advanced algorithms can buy back these lost DOF.

The recognition that second-order statistics could be used to extend the effective degrees of freedom in antenna arrays is a fairly old one, and possibly originated from the field of radio astronomy (e.g., [1]) where fully populated arrays are often impractical. Moffet introduced the concept of *minimum redundancy arrays* (MRAs) [2], which are sparse arrays with the fewest elements required to provide a complete set of pairwise element separations. The *difference coarray* [3] later formalized the idea of a virtual array defined on these separations. In this context, a weighted sum of the correlation values came to be known as *correlation beamforming* [3, Sec. IV], with the corresponding "pattern" having degrees of freedom based on the length of the coarray ($\mathcal{O}(N^2)$) rather than the number of physical elements (N). A different approach

Manuscript approved April 6, 2022.

based on the same concepts is known as covariance-matrix *augmentation* [4–7]. Here, it was recognized that the covariance matrix of a uniform linear array (ULA) receiving uncorrelated signals has a Toeplitz structure, and thus is highly redundant. (Indeed, such a covariance matrix merely encodes a central segment of the covariance sequence of a stationary spatial process.) Sparse arrays whose element spacings created filled difference coarrays thus provided sufficient correlation values to reconstruct the covariance matrix of a filled ULA. This covariance matrix then could be used in various algorithms as if a full ULA were available. One can extend these virtual-array ideas further to high-order moments [8–10], although this requires non-Gaussian sources and will not be considered further here.

The current revival of difference-coarray-based processing that sparked this project seems to have been led by [11], which is focused on the specific *nested-array* sparse configuration and a matching spatial smoothing technique. A second sparse scheme, *coprime arrays* [12, 13], also has found favor, at least in part on the largely unsubstantiated claim that it mitigates mutual coupling.[1] As in the older correlation-beamforming approach, the analysis revolves around a nonlinear model of the covariance, itself, as a virtual signal with support on the coarray, with signal powers replacing complex amplitudes. In this way, these papers generally claim available degrees of freedom equivalent to a linear array the length of the coarray, which was notable (and, as we will show, questionable).

1.2 Outstanding Questions and Challenges

The discussion above suggests potential benefits that nonlinear array processing (NAP) can offer to sparse array processing in general. However, the recent body of literature has left some significant questions and challenges unresolved. Below, we pose some fundamental questions that guided our research.

NAP degrees of freedom: One of the more confusing concepts related to NAP is determining the true number of DOF that are available to be exploited. Given N sensors in a sparse configuration with a span of M between the outer elements and a filled difference coarray of length $2M - 1$, which of those numbers represents the true degrees of freedom? The short answer turns out be, “It depends.” Context, as always, is key.

Sample-support requirements: One of the primary drawbacks of much of the NAP literature is that the analysis is based on the ideal covariance of the incoming signals. This leads to somewhat unintuitive results, such as noise being deterministic and occupying a single-dimensional subspace. This analysis is not wrong, *per se*, but it also can be quite misleading in the overwhelmingly common case in which the covariance must be estimated using finite sample data. Here, the cross-correlations between nominally uncorrelated sources remain nonzero and thus appear in the NAP signal model as a large number of additional “sources” that threaten to absorb the additional DOF that are NAP’s *raison d’être*. Determining how many data snapshots are required is thus fundamentally important, and yet is largely overlooked. In standard linear-adaptive filtering, the popular “RMB” rule of thumb derived in [14] requires $2N$ data snapshots for an N-element ULA. It will be shown, however, that far more samples typically are needed for NAP, especially in the case of more sources than physical sensors. NAP generally assumes that the underlying signals are uncorrelated; however, in practice, a virtual

[1] This claim derives from the fact that a coprime array has fewer immediately adjacent elements, which are presumed to have greater coupling. However, all sparse arrays depend on having at least one minimum-spaced pair of elements to generate the lag-1 correlation, and coprime arrays of any significant length have multiple such pairs. It is far from clear that their more-scattered distribution vs. other schemes improves performance in practice.

measurement becomes contaminated with contributions from cross terms existing due to limited time support that tend to decay rather slowly. As such, they can be a significant source of interference for the virtual array, as they can leak through the increased sidelobes of the virtual pattern.

Algorithms to maximize the full potential of NAP: As previously mentioned, the key idea in NAP is to treat the virtual array as if it were a Nyquist array and to transpose over known spatial-processing algorithms. However, this poses a challenge for many conventional direction-of-arrival (DOA) estimation algorithms, such as MUltiple SIgnal Classification (MUSIC), that rely on estimating an array's data covariance matrix from a collection of array snapshots. In the virtual array, a single (virtual) measurement *is* this estimated covariance, so in fact, what would be needed would be the covariance of the covariance. Such a creature is elusive, as the statistics of the sample covariance violate the usual assumptions: It has a nonzero mean (the true data covariance) and is composed of correlated sources (the positive-valued signal powers). So even if we were to have sufficient data to compute multiple virtual snapshots (sample covariances) and then to use those to compute a sample covariance of the covariance, in general, it would not contain the subspace information we seek. At least three potential solutions have presented themselves so-far:

1. Compute multiple sample covariances from successive blocks of data, and form a covariance estimate directly. This requires the blockwise "quasi-stationarity" of [15], where the data within each block is stationary with the signal powers (but not directions) varying between blocks. This has obvious drawbacks due to both the large amount of data required and the tenuous assumptions.
2. Use spatial smoothing to build up the rank of the virtual covariance matrix using only a single virtual measurement, as in [11]. The drawback with this is that spatial smoothing decreases the length of the virtual array by half, which, as we discuss in this report, may reduce the spatial resolution performance.
3. Use single-snapshot methods based on compressive sensing [16] or reiterative minimum-mean-square-error (MMSE) estimation [17]. The latter produces as a byproduct an estimate of the covariance of the random process that generated the snapshot, which then can be used in MUSIC or other algorithms.

We chose to focus our efforts on item 3, as that appeared to be the most promising avenue.

The suitability of NAP for active sensing: The NAP literature is focused almost exclusively on passive, receive-only array processing such as spectral estimation and DOA estimation. Linear adaptive processing (beamforming) is widely used in active sensing such as radar and sonar, and so it is natural to wonder if NAP might provide benefits to such applications. However, there are fundamental differences between active and passive sensing, and between linear-adaptive processing and NAP, which tend to limit NAP's applicability to active sensing. We note that there exists a class of adaptive beamforming algorithms [18, 19] that are hybrids of active and passive methods. These approaches apply augmented-covariance methods in order to enhance DOA and signal-power estimation, then use that information to generate a non-augmented covariance and compute a linear filter to be applied to the original sparse data. These methods do not increase the available degrees of freedom in the beamforming, itself, but may improve the robustness of the beamforming. We mention these approaches for completeness, but will not consider them further here and will focus on adaptive beamforming applied to the coarray, itself [20].

In linear-adaptive processing, the noise-plus-interference covariance ideally is estimated from data without the desired signal present, so-called *auxiliary data*. This covariance then is used either to "whiten" the interference spectrum of the complex data prior to conventional filtering, or equivalently, to design adaptive beamformer weights to remove the interference. The usual effect is to place nulls in the direction of interference sources, and the number of such nulls is limited in linear processing by the number of physical elements. Because the filtering or whitening is a linear operator, it generally can be performed either before or after other coherent linear processing. However, in NAP coarray beamforming, the input "signal" is the sample covariance. Clearly this must contain any desired signals one hopes to detect. For passive receivers, this is not a problem for stationary sources, which, by definition, are persistent. But active systems such as radar generally must limit the duration over which they illuminate a given direction in order to scan the entire volume in a timely manner. This limits the data available with a given target present. What's more, radars generally require coherent integration over the full temporal extent of the waveform in the form of pulse compression and Doppler processing, which leaves no remaining dimension from which to pull independent data snapshots. Thus, one must either extend the signal duration or reduce the coherent integration. Combined with the aforementioned high snapshot requirements, the problem quickly becomes infeasible. Another obstacle is that because the output of NAP processing has lost its phase information, it must come last in the processing chain after any linear processing. Deferring spatial processing until the end may greatly multiply the total computation required.

1.3 Contributions

The challenges itemized above motivate what we see as the main contributions of this work:

Establishing a basic characterization of the necessary time sample support: We take a more in-depth look at understanding the level of error introduced when limited to K snapshots for estimating the virtual array measurements. Although this is done under simplified conditions of uncorrelated Gaussian sources, the insights gained provide for a good rule of thumb for understanding the level of error introduced, and to our knowledge is the only contribution towards this question to date.

Understanding limitations in DOF and resolution performance of the approach: As mentioned above, spatial smoothing has the effect of reducing the size of the virtual array by half, which effectively reduces the available DOF in the virtual array as well as the resolution performance. Looking more deeply using Cramer-Rao lower bound (CRLB) analysis, we can establish that this limitation in DOF is not a specific property of spatial smoothing, but rather an inherent limitation of the information content captured by the sparse array. However, our study finds that this is not necessarily the case for the aperture size in terms of resolution performance. We hypothesize that this is a specific limitation of spatial smoothing.

Evaluating efficacy of non-spatial-smoothing-based algorithms for NAP: We look to compare several state-of-the-art algorithms for the conventional linear model and to understand if they can provide alternatives to spatial smoothing in order to maximize performance gains within the NAP framework. This includes which algorithmic families are more efficient with respect to their time sample support, and which can leverage the full aperture length of the virtual array for resolution performance.

Establishing a suitable experimental testbed for evaluating NAP: Most, if not all, of the NAP studies to date have presented results based on simulation and from a receive-only point of view. One of the key contributions of this work is the construction of a testbed that effectively emulates the receive side of a monostatic radar system using a 12-channel array at a 3.4 GHz center band. In this study, we employed up to eight independent far-field sources that can emulate a linear frequency modulated (LFM) chirp signal or can serve as sources of incoherent interference. Although of most of the analysis in this work was geared toward a 6-element nested array, other sparse arrays, such as minimum redundancy arrays, that fit within the extent of a 12-channel uniform linear array could be analyzed as well.

1.4 Outline of the Report

The remaining chapters of the report are summarized as follows: Chapter 2 serves as a technical overview of the major concepts and notation concerning space-time processing with linear arrays. This includes modeling the array and its received signals and summarizing state-of-the-art beamforming and DOA algorithms for the conventional linear model that later will extend to the NAP framework. Chapter 3 introduces various concepts related to sparse array signal processing, including covariance beamforming, the difference coarray, and the augmented covariance matrix. Chapter 4 extends the algorithms introduced in Chapter 2 to the NAP framework and is where the main theoretical and algorithmic contributions of this work are described. We show that in terms of information content, there is no advantage of NAP over other traditional approaches, such as covariance augmentation, in terms of obtaining more DOF than the number of elements in the array. We then characterize the finite sample support contribution of error in the NAP virtual array as well as consider a number of algorithms and their performance for the DOA estimation problem. We then consider algorithms for the adaptive filtering problem, where we show the detriment of limited time sample support on performance. Our experimental setup is described in Chapter 5, as well as the specific data collections that were performed in this work. Here, we also apply the algorithms of Chapter 4 to selected experimental data, to demonstrate the effects of real-world model mismatch and errors on the algorithms. Final conclusions and recommendations for follow-on work round out the report in Chapter 6.

2. ARRAY-PROCESSING BACKGROUND

In this chapter, we briefly summarize background material on array processing that is prerequisite to the remainder of this report. This includes the basic space-time geometry, the signal model, and common processing including beamforming and various DOA estimation algorithms. The reader familiar with these topics can skip this chapter and return as needed for reference. We will confine ourselves to linear (one-dimensional) array architectures only, since the established concepts can be generalized trivially to multiple dimensions, and this avoids the added complication in notation.

2.1 Linear Arrays

Broadly speaking, an antenna array is a collection of antennas that work together to emulate the function of a single antenna during transmit or receive operations. As mentioned in the introduction, the chief advantage of antenna arrays over continuous-aperture antennas is the increase in DOF. Each individual antenna in an array is called an element of the array. When the elements of the array are laid out across a line, the array is referred to as a linear array. As such, the field of view of a linear array is restricted to only $[-90°, 90°]$ across the polar angle, and has no diversity with respect to azimuth.[2] More general 2D surface arrays enable diversity over the entire half-space (2π steradian). Consider the case of N antenna elements on a line as shown in Fig. 1. The position of the j^{th} antenna is denoted as $\mathbf{r}_j = [u_j, 0, 0]^T$, where $u_j = n(j)\Delta u$ with $n(j)$ taken from some integer set $\mathbb{S} \subset \mathbb{Z}^+$ such that $|\mathbb{S}| = N$. The quantity $\Delta u \in \mathbb{R}^+$ defines the minimum spacing between consecutive elements and typically is taken to be the Nyquist distance of $\Delta u = \lambda/2$. Let's say we have an incoming monochromatic source $e^{i\omega t}$ (such as a scatterer or interference) at angular frequency ω. For most applications of interest, the targets and signal sources are well into the far-field region of the array, and we assume that the electromagnetic field is in the form of a plane wave $\Psi(t, \mathbf{r}) = e^{i(\omega t - \frac{2\pi}{\lambda}\hat{\mathbf{d}}\cdot\mathbf{r})}$, where vector

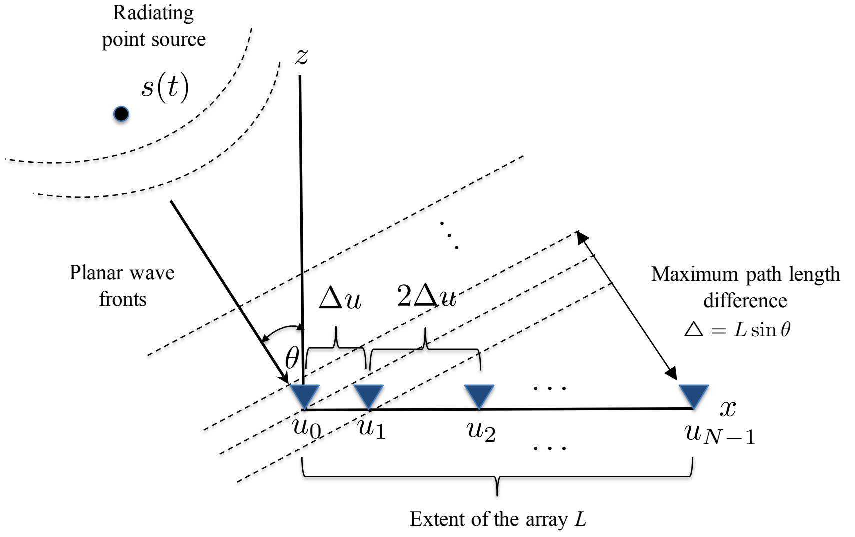


Fig. 1—Basic geometry of a linear array relative to an incoming plane wave

[2] Here, "azimuth" refers to the generic spherical-coordinate angle about the coordinate-system pole, which, for a linear array, is naturally the array axis. This should not be confused with the specific radar meaning of "azimuth," which places the coordinate-system pole normal to the Earth's surface regardless of the array orientation.

$\hat{\mathbf{d}}$ defines the direction of propagation. Furthermore, recall that in the case of a nondispersive homogenous media, we have the relationship $\frac{2\pi}{\lambda} = \omega/c$, with $c > 0$ defining the speed of propagation. According to the coordinate system in Fig. 1, we can write $\hat{\mathbf{d}}(\theta, \phi) = -[\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta]^T$, where θ is the polar angle (the angle with respect to the plane normal to the array axis), and ϕ is the azimuth angle (the angle about the axis). Then the electric field sampled at position $\mathbf{r}_j$ at time t is given as

$$x_j(t) = e^{-i\omega(t-\tau_{n(j)})},$$

where the propagation delays τ_n are given as

$$\tau_n = -\frac{n\Delta u \sin\theta}{c}. \tag{1}$$

More generally, for a signal $f(t) = \displaystyle\int F(\omega)e^{i\omega t}d\omega$ propagating in the form of a plane wave, the array response is given as

$$x_j(t) = f(t - \tau_n),$$

where we have dropped the dependence of n on j for notational brevity. We are usually interested in the case of a bandpass waveform centered over a certain frequency ω_c:

$$f(t) = s(t)e^{i\omega_c t},$$

where $s(t)$ is the complex envelope of the transmitted pulse and is bandlimited to a bandwidth of B. The response at each array element is then written as

$$x_j(t) = s(t - \tau_n)e^{i\omega_c(t-\tau_n)}. \tag{2}$$

If the pulse width is such that it is much larger than the maximum time it takes for the wave to propagate between any two elements in the array Δ/c, then we can make a narrowband approximation, essentially $s(t - \tau_n) \approx s(t)$. Therefore, in conjunction with Eq. (1), we can reduce Eq. (2) to

$$x_j(t) = s(t)e^{i\frac{2\pi}{\lambda}u_n sin\theta}e^{i\omega_c t}.$$

After demodulation to baseband and accounting for noise at each antenna element, we have the following linear model:

$$\mathbf{x}(t) = \mathbf{a}_{\mathbb{S}}(\theta)s(t) + \mathbf{n}(t),$$

relating the array measurements to the source amplitudes. The $N \times 1$ vector $\mathbf{x}(t)$ is the spatial snapshot of the field at time t, the vector function $\mathbf{a}_{\mathbb{S}}(\theta)$ is known as the array steering vector for the array $\mathbb{S}$, and $\mathbf{n}(t)$ is a random additive noise term. For far-field propagation, the steering vector is defined per Eq. (1) as

$$\mathbf{a}_{\mathbb{S}}(\theta) = [e^{i\frac{2\pi}{\lambda}u_0\sin(\theta)}, \ldots, e^{i\frac{2\pi}{\lambda}u_{N-1}\sin(\theta)}]^T. \tag{3}$$

Extending to a more general case of M narrowband sources propagating as plane waves simultaneously impinging on the array with time dependence $\mathbf{s}(t) = [s_0(t), \ldots, s_{M-1}(t)]$ and arrival angles $\boldsymbol{\theta}^* = [\theta_1, \ldots, \theta_M]$, we can describe the received signal as the coherent sum of all the sources

$$\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\theta}^*)\mathbf{s}(t) + \mathbf{n}(t). \tag{4}$$

Here, $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}_{\mathbb{S}}(\theta_1), \ldots, \mathbf{a}_{\mathbb{S}}(\theta_M)]$ defines a matrix function of $\boldsymbol{\theta}$ that we will refer to as the array manifold and depends on the array architecture $\mathbb{S}$. When these signals are zero-mean wide-sense-stationary (WSS) random processes, they can be characterized via their covariance matrix:

$$\mathbf{R_x} \triangleq E\left[\mathbf{x}\mathbf{x}^H\right] = \mathbf{A}\mathbf{R_s}\mathbf{A}^H + \mathbf{R_n}. \tag{5}$$

Here, we have assumed that the noise is uncorrelated with the other signals, and by convention have dropped the explicit dependence of the manifold $\mathbf{A}$ on $\boldsymbol{\theta}$ for notational simplicity.

For digital array processing, in most cases, the output of each array element is sampled with respect to a fixed sampling rate Δt. In radar systems. there are a number of time scales in which time samples could be taken across. In a typical pulse-Doppler system, a periodic series of pulses is transmitted according to the radar's pulse repetition interval (PRI). Time samples obtained during each individual pulse correspond to a fast-time scale and typically are reserved for range processing such as pulse compression. Time samples taken across a group of pulses within a coherent processing interval (CPI) are referenced to a much slower sampling rate and commonly are referred to as slow-time samples. Coherent integration operations, such as those in array or Doppler processing, usually are performed with respect to the slow-time dimension. It is during a CPI in which the WSS assumption of the underlying sources can be considered reasonable. Time samples also can be taken across a group of CPIs (an entire radar dwell), in which for most radar applications, WSS assumptions typically do not hold. In any case, a discretized model based on a sampling rate of the appropriate scale is given as

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k), \tag{6}$$

where $\mathbf{x}(k) = \mathbf{x}(k\Delta t)$ for time snapshot k. Broadly speaking, array-processing algorithms can be classified as either those that can work readily with a single measurement $\mathbf{x}(k)$ at each time instant k, or those that rely on multiple array snapshots in order to render output at time instant k. The former will be referred to as single-measurement-vector (SMV)-capable methods. For the latter type of methods, multiple snapshot data consisting of K snapshots can be organized into the matrix

$$\mathbf{X}_K(k) = [\mathbf{x}(k), \mathbf{x}(k-1), \ldots, \mathbf{x}(k-K+1)]^T\,,$$

which we will refer to as multiple measurement vector (MMV) data.

2.2 Beamforming

Focusing the radar's antenna towards a particular direction can be accomplished by choosing an appropriate linear combination of the array channels,

$$y(k) = \mathbf{w}^H\mathbf{x}(k), \tag{7}$$

where **w** is the weight vector. Substituting Eq. (4) into Eq. (7) yields

$$\begin{aligned} y(k) &= \sum_{m=0}^{M-1} \mathbf{w}^H \mathbf{a}_{\mathbb{S}}(\theta_m) s_m(k) + \mathbf{w}^H \mathbf{n}(k) \\ &= \sum_{m=0}^{M-1} \mathcal{W}(\theta_m) s_m(k) + \mathbf{w}^H \mathbf{n}(k). \end{aligned} \tag{8}$$

where $\mathcal{W}(\theta) \triangleq \mathbf{w}^h \mathbf{a}_{\mathbb{S}}(\theta)$ defines the *array pattern.* For far-field propagation in Eq. (3), this becomes

$$\begin{aligned} \mathcal{W}(\theta) &= \sum_{n=0}^{N-1} w_n^* e^{i \frac{2\pi}{\lambda} u_n \sin(\theta)} \\ &= W^*\Big(\frac{\sin(\theta_m)}{\lambda}\Big), \end{aligned} \tag{9}$$

where W is the spatial Fourier transform of the weight vector:

$$W(v) = \sum_{n=0}^{N-1} w_n e^{-i2\pi u_n v}.$$

The array pattern is a fundamental property of a linear beamformer and represents its response to a single incoming plane wave in the absence of noise. The argument $v = \sin(\theta)/\lambda$ is a spatial frequency, with units of reciprocal distance. The standard response of the array is a simple sum across the channels, i.e., a unity weighting across all the array elements. This corresponds to the unadapted pattern of the array and can be characterized by a main lobe at boresight and a finite number of sidelobes at various points within the field of view. The width of the main lobe defines the Rayleigh diffraction limit B_w of the array, which is approximately proportional to the inverse of the array length:

$$B_w \approx \lambda/L.$$

Thus, for a Nyquist ULA, the resolution of the array is dictated by the number of elements N. More generally, we seek a choice of **w** (and thus W) to spatially filter the FOV, i.e., we seek to enhance the response of a desired signal or direction, and when appropriate, to suppress energy from other directions. The Fourier beamformer,

$$y(t) = \mathbf{a}_{\mathbb{S}}^H(\theta)\mathbf{x}(t),$$

is a simple example of a beamformer that enhances the signal from a particular direction against overall noise, and is the spatial equivalent of matched filtering. Indeed, through a simple Cauchy-Schwarz argument, it can be shown that a Fourier beamformer maximizes SNR for a single signal from direction θ in white noise, providing a SNR gain of N relative to a single sensor. Fourier beamforming, however, is still subject to the Rayleigh resolution limit and also features a number of relatively high sidelobes that can admit interference from undesired directions. The sidelobes can be improved via amplitude tapering, but this also has the effect of reducing SNR.

More sophisticated beamformers seek not only to improve the overall array's response in the desired direction, but also to null out undesired directions containing potential interference. If we consider a scenario in which $M-1$ jammers/interferers are impinging on the array from the directions $\theta_1, \ldots, \theta_{M-1}$, respectively, the array signal can be decomposed as

$$\mathbf{x}(t) = \mathbf{t}(t) + \mathbf{i}(t) + \mathbf{n}(t),$$

where $\mathbf{t}(t) = \mathbf{a}_{\mathbb{S}}(\theta_1)s_1(t)$ corresponds to the desired signal (target), $\mathbf{i}(t) = \sum_{m=2}^{M} \mathbf{a}_{\mathbb{S}}(\theta_m)s_m(t)$ corresponds to the interference component, and $\mathbf{n}(t)$ represents ambient noise. Note that subsequently, we simply will use the notation $\mathbf{a}(\theta)$ when referring to the array steering vector, and only use the full notation $\mathbf{a}_{\mathbb{S}}$ when the context of the discussion requires an explicit reference to the array's geometric structure. If the directions of the sources and the interference are known, a beamformer $\mathbf{w}$ can be designed so that the overall beam matches the response $\mathbf{b}(\boldsymbol{\theta}) = [1, 0, \ldots, 0]^T$ in a least-square sense. So for the underdetermined case $M \leq N$, one exact (but non-unique when $M < N$) solution is

$$\mathbf{w} = \mathbf{A}\left(\mathbf{A}^H\mathbf{A}\right)^{-1}\mathbf{b}, \tag{10}$$

while for the overdetermined case $M > N$, the closest (inexact) solution is

$$\mathbf{w} = \left(\mathbf{A}\mathbf{A}^H\right)^{-1}\mathbf{A}\mathbf{b}, \tag{11}$$

where again, $\mathbf{A}$ is the array manifold defined for the vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]^T$. Thus, we see for an N element array, we can specify no more than $N-1$ explicit nulls. In this sense, the DOF of the array also plays an important role in guarding against being overwhelmed by a large number of interference sources.

When the directions of the interference sources are not known *a priori*, adaptive approaches such as the minimum-variance distortionless response (MVDR) beamformer can be considered. The MVDR weights are designed to maximize the signal-to-interference-plus-noise ratio (SINR) and can be shown to be

$$\mathbf{w}_{\text{mvdr}} = \frac{\mathbf{R}_{i+n}^{-1}\mathbf{a}(\theta_1)}{\mathbf{a}^H(\theta_1)\mathbf{R}_{i+n}^{-1}\mathbf{a}(\theta_1)}, \tag{12}$$

with $\mathbf{R}_{i+n}$ denoting the covariance matrix with respect to the interference and noise components only (interested readers may refer to Appendix B for a brief discussion on the derivation). Another adaptive approach of interest is the minimum mean-square error (MMSE) filter, where the mean-square error is minimized with respect to the underlying target signal amplitude $s_1(t)$. These weights can be shown to be computed as

$$\mathbf{w}_{\text{mmse}} = \sigma_1^2\left(\mathbf{R}_{i+n} + \sigma_1^2\mathbf{a}(\theta_1)\mathbf{a}^H(\theta_1)\right)^{-1}\mathbf{a}(\theta_1), \tag{13}$$

where $\sigma_1^2 = E\left[|s_1|^2\right]$. In fact, we further can show that the MMSE weights are actually a shrinking factor of the MVDR solution,

$$\mathbf{w}_{\text{mmse}} = \beta(\sigma_1)\mathbf{w}_{\text{mvdr}}, \tag{14}$$

with $0 \leq \beta(s) \leq 1$ and monotonically increasing with σ_1^2. Thus, the MMSE beamformer also maximizes SINR (when $\mathbf{R}_{i+n}$ is known), but is known to provide a better estimate of $s_1(k)$ as it is scaled to minimize the estimation error with respect to this. An obvious but important note is that the MMSE filter requires in addition to $\mathbf{R}_{i+n}$ that the target power be known *a priori*.

Spatial filters like MVDR and MMSE achieve their respective optimal performance under the assumption that $\mathbf{R}_{i+n}$ is known exactly. In practice, however, the covariance data needs to be estimated from MMV data. This usually entails using the sample covariance matrix,

$$\mathbf{S}_x = \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}(k)\mathbf{x}^H(k), \tag{15}$$

in place of the ideal covariance matrix. The choice of K then plays an important role; K should be chosen large enough so that $\mathbf{S}_x$ is full-rank and sufficiently converges to the underlying covariance matrix; however, K should be small enough to satisfy the stationary assumption and to adhere to radar system performance requirements. In practice, a good rule of thumb (known as the Reed-Mallat-Brennan (RMB) rule) [21] is about twice the number of elements to ensure that the average SNR loss of an adaptive beamformer output is no more than 3 dB. Another issue of practical importance specifically to spatial filters like MVDR and MMSE beamforming is that in many cases, the signal of interest is also present in the MMV data, in which case directly using the associated sample covariance matrix often leads to significant performance degradation. Although this issue is outside the scope of this report, we do note that there are various techniques, such as diagonal loading the sample covariance matrix, that can help mitigate such effects. We refer interested readers toward [22] for a comprehensive discussion on the topic.

2.3 DOA Estimation

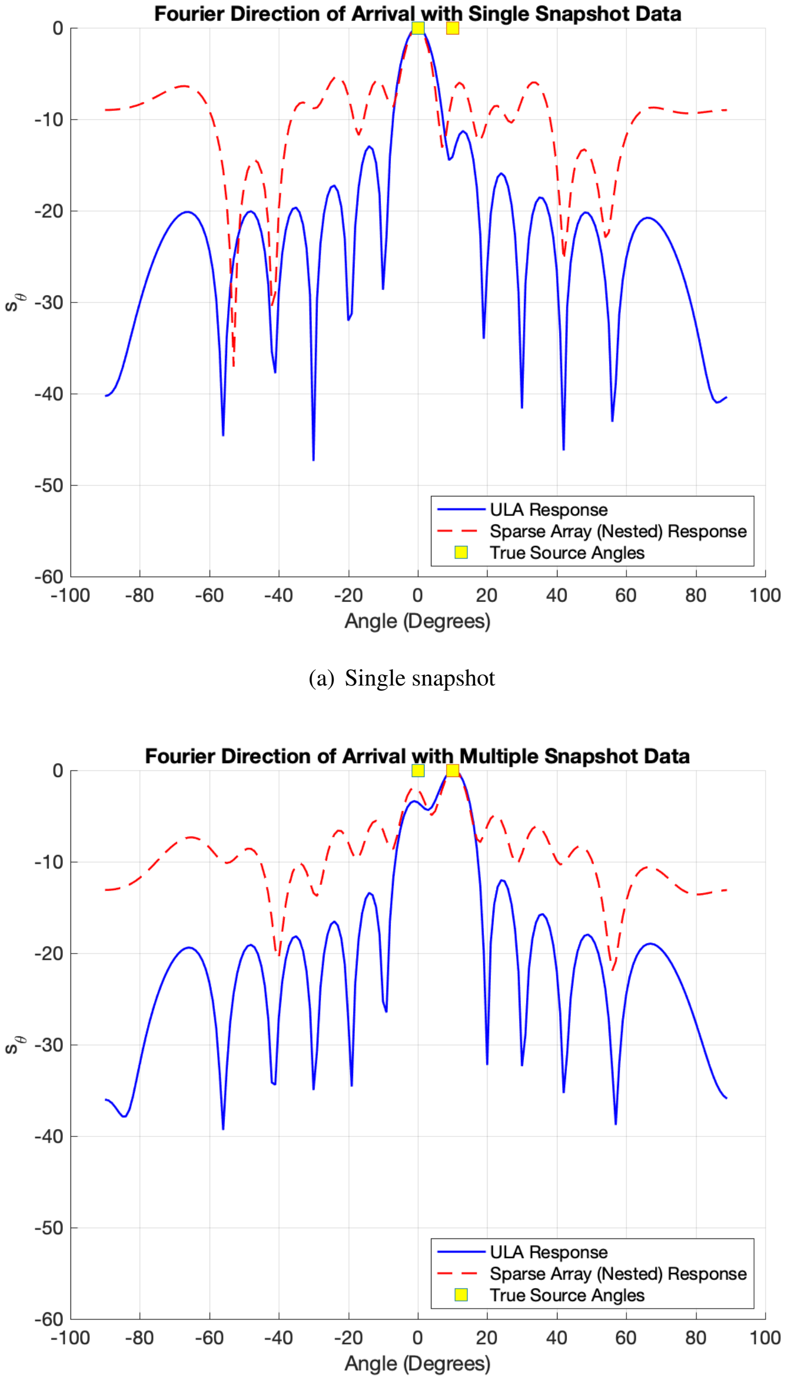
The problem of identifying the direction of arrival of various plane wave sources is referred to as the DOA estimation problem. Beamforming techniques such as the ones discussed above can serve readily as DOA estimators by sweeping the viewing angle θ across the FOV of the array,

$$p(\theta) = \left|\mathbf{w}^H(\theta)\mathbf{x}(k)\right|^2 .$$

For the case of a discrete angular search grid vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_P]^T$, beamforming-based DOA estimation can be described by a $N \times P$ matrix (filter bank) $\mathbf{W}$, whose spectrum output is given as

$$p(\boldsymbol{\theta}) = |\mathbf{W}^H \mathbf{x}(k)|^2.$$

Fourier beamforming, where we have $\mathbf{W} = \mathbf{A}$, has the advantage of computational simplicity as well as being easily adaptable to arbitrary array architectures. Furthermore, Fourier DOA estimation also is suitable for SMV data. The main drawbacks of the Fourier approach are again the Rayleigh resolution limit as well as the sidelobes that can mask potential sources. In fact, sidelobes are particularly problematic with sparse arrays, as we can see in Fig. 2. In this example, we see the nested array exhibiting sidelobes roughly 10 to 12 dB larger than those using a sufficiently sampled ULA. Distinguishing sources from sidelobes, however, can be aided through non-coherent integration across MMV data. For example, in Fig. 2, we have a case where the nearby source potentially could be confused with a sidelobe, but when integrating across multiple snapshots,



(a) Single snapshot

(b) Multiple snapshots

Fig. 2—Illustrating Fourier-based DOA estimation with SMV and MMV data for both a ULA and a nested array. In the SMV case, the source at 10° appears as a sidelobe for both arrays. Non-coherent averaging with multiple snapshots helps bring out the second source but does not improve resolution.

the second source is significantly enhanced. However, note that the use of MMV data did not help with improving the overall resolution of the Fourier estimator. Adaptive beamforming techniques such as MVDR beamforming, on the other hand, rely on MMV data in order to compute the necessary sample covariance matrices. In this scenario, unlike the beamforming problem, covariance from derived from the MMV data should capture all the signals. The advantage of these types of beamforming-based DOA estimators is that they can achieve some degree of super-resolution. Another important point regarding beamforming-based DOA estimation is that beamforming is a linear operator with respect to the array data, thus the output signal of the filter bank retains phase information, and thus DOA estimation can be applied prior to any range or Doppler processing. This is not generally the case for DOA estimation techniques, as more sophisticated algorithms are nonlinear and thus do not preserve any phase information for subsequent coherent processing.

2.3.1 Subspace Methods

Subspace methods are among the most recognized super-resolution DOA estimation approaches in the current literature. These approaches are based on an assumed structured covariance model that arises under the following assumptions:

Assumption 1. *The additive noise term in Eq.* (6) *is stationary, and* $\mathbf{n}(k) \sim \mathcal{CN}\left(\mathbf{0}, \sigma_0^2\mathbf{I}\right)$.

Assumption 2. *The sources* $\mathbf{s}(k)$ *are wide-sense stationary.*

Assumption 3. *The sources are zero mean and uncorrelated with each other, i.e.,*

$$\mathbf{\Sigma} = E\left[\mathbf{s}(k)\mathbf{s}^H(k)\right] = diag\left[\sigma_1^2, \ldots, \sigma_M^2\right].$$

Under these assumptions, the covariance data matrix defined in Eq. (5) can be represented as

$$\mathbf{R}_x = \mathbf{A\Sigma A}^H + \sigma_0^2\mathbf{I}. \tag{16}$$

The MUSIC algorithm is one of the most established methods in the class of subspace techniques. These methods rely on an eigendecomposition of the correlation matrix $\mathbf{R}_x$ to isolate the noise/interference subspace from the signal subspace, and exploiting the orthogonality of the two. More specifically, if $\mathbf{\Sigma}$ is full rank and $M \leq N$, then the rank of $\mathbf{A\Sigma A}^H$ is M, with the smallest $N - M$ of its eigenvalues being zero. Therefore, by performing an eigendecomposition and ranking the eigenvalues and their associated eigenvectors from largest to least, we find that the top M Eigenvectors $\{\mathbf{u}_1, \ldots \mathbf{u}_M\}$ correspond to the signal subspace, while the remaining $N - M$ eigenvectors $\{\mathbf{u}_{M+1}, \ldots, \mathbf{u}_N\}$ form a basis of the noise subspace with repeated eigenvalue of σ_0^2. Since the signal and noise subspaces form orthogonal complements of each other, and since $\mathbf{a}(\theta)$ is a basis of the signal subspace, then

$$p(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{U}_{M-N}\mathbf{U}_{M-N}^H\mathbf{a}(\theta)}$$

can be used as an indicator function for the DOAs, where $\mathbf{U}_{M-N} = [\mathbf{u}_1, \ldots \mathbf{u}_M]$. Indeed, since $\mathbf{a}^H(\theta)\mathbf{u}_j = 0$, for any $j = M+1, \ldots, N$, the MUSIC spectrum becomes infinite (or very large in practice) when θ coincides with the DOA of one of the sources. Other examples that are in this class include ESPIRIT [23], which is a computational and storage-reduction improvement over MUSIC, and root-MUSIC [24], which further improves on resolution by looking for the roots of the spectrum explicitly.

2.3.2 Reiterative Superresolution Algorithms

As we have discussed above, the MMSE beamformer attempts to estimate the underlying signal of interest in a least-mean-square sense. Under the assumed covariance structure Eq. (16), the associated MMSE filter bank then is given as

$$\mathbf{W}_{\text{mmse}} = \left(\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^H + \sigma_0^2\mathbf{I}\right)^{-1}\mathbf{A}\boldsymbol{\Sigma}. \tag{17}$$

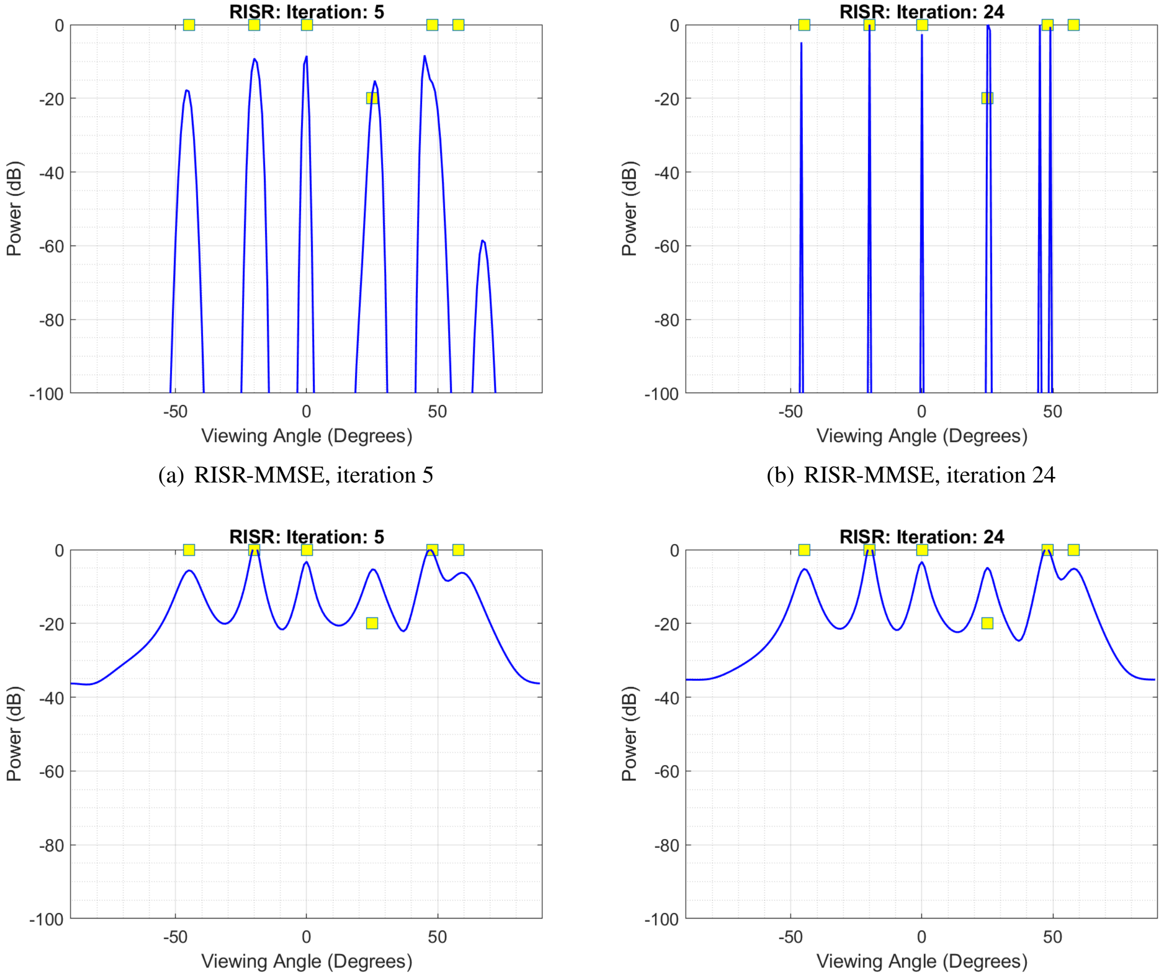
Like we mentioned before, the MMSE estimator Eq. (17) assumes knowledge of the source covariance matrix $\boldsymbol{\Sigma}$, which is generally unknown *a priori*. Blunt proposed a recursive scheme for updating the MMSE filter bank using successive estimates of the source covariance matrix that he termed as a reiterative superresolution (RISR) algorithm [25]. More specifically, for a given initial estimate $\boldsymbol{\Sigma}^{(0)}$, the technique can be summarized as follows:

$$\begin{aligned}
\mathbf{W}^{(p+1)} &= \left(\mathbf{A}\boldsymbol{\Sigma}^{(p)}\mathbf{A}^H + \sigma_0^2\mathbf{I}\right)^{-1}\mathbf{A}\boldsymbol{\Sigma}^{(p)} \\
\boldsymbol{\Sigma}^{(p+1)} &= \operatorname{diag}\left[\frac{1}{K}\sum_{k=1}^{K}\mathbf{W}^{H(p+1)}\mathbf{x}(k)\right] \odot \mathbf{I},
\end{aligned}$$

iterating until some convergence criteria is satisfied. While to our knowledge, a rigorous study of the convergence properties of the RISR method has not been performed, we observed in our experimental studies that RISR tends to approach a solution rather quickly and admits high-resolution DOA estimates similar to those computed by subspace techniques like MUSIC. However, the method also tended to oscillate around a saddle point and/or to exhibit false modes at the expense of missing true sources entirely. As an example, we consider Fig. 3, where we can see at iteration 24 RISR mistakenly places two peaks at the source located at 48°. A variant of the RISR algorithm, as proposed in [26], is based on replacing the MMSE filter bank with a MVDR filter bank, in which case the RISR iterations become

$$\begin{aligned}
\mathbf{R}_x^{(p+1)} &= \left(\mathbf{A}\boldsymbol{\Sigma}^{(p)}\mathbf{A}^H + \sigma_0^2\mathbf{I}\right) \\
\mathbf{W}^{(p+1)} &= \frac{\left(\mathbf{R}_x^{(p+1)}\right)^{-1}\mathbf{A}}{\mathbf{1}_N\operatorname{diag}\left[\mathbf{A}^H\left(\mathbf{R}_x^{(p+1)}\right)^{-1}\mathbf{A}\right]} \\
\boldsymbol{\Sigma}^{(p+1)} &= \operatorname{diag}\left[\frac{1}{K}\sum_{k=1}^{K}\mathbf{W}^{H(p+1)}\mathbf{x}(k)\right] \odot \mathbf{I},
\end{aligned}$$

where $\mathbf{1}_N$ is the $N \times 1$ vector of ones. As we know with MVDR beamformers, the source covariance matrix does not factor in explicitly into the filter update equation; rather, it's contained implicitly in the measurement covariance matrix. The effect of this that we observed in our numerical and experimental study is that using the MVDR update provided more overall stability at the expense of a moderate decrease in resolution as compared to the MMSE version. Figure 3 shows an example of this, in which we see that convergence of RISR-MVDR was achieved as early as iteration 5. Furthermore, the sources were identified correctly as opposed to the MMSE version; however, a significant degradation in resolution also can be seen when compared to the resolution performance provided by the MMSE version of RISR.



(a) RISR-MMSE, iteration 5

(b) RISR-MMSE, iteration 24

(c) RISR-MVDR, iteration 5

(d) RISR-MVDR, iteration 24

Fig. 3—An example to highlight some of the differences in behavior between the two RISR variants. Here, we apply both RISR DOA estimation algorithms to a 12-element ULA with six sources. The MMSE version exhibits much sharper resolution, but tends to miss sources and/or to display false peaks. Using the MVDR weight update seems to provide better stability overall at the expense of resolution performance loss.

2.3.3 Maximum Likelihood Estimation and Expectation Maximization

Taking a stochastic view of the array data, each snapshot $\mathbf{x}(k)$ constitutes an independent observation of the random vector $\boldsymbol{x}$, for which we denote $f_{\boldsymbol{x}}(\mathbf{x}|\boldsymbol{\alpha})$ as its underlying probability distribution function (PDF) conditioned on a set of parameters $\boldsymbol{\alpha}$. The parameter vector $\boldsymbol{\alpha}$ includes the DOAs as well as the amplitudes of the sources. Based on the assumed covariance structure Eq. (16), the problem of finding the maximum likelihood estimate (MLE) with respect to each parameter α_k (as shown in Appendix C) can be reduced to solving the following equation:

$$K\mathrm{tr}\left[\left(\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1} - \mathbf{R}_x^{-1}\right)\frac{\partial \mathbf{R}_x}{\partial \alpha_k}\right] = 0. \tag{18}$$

We see that in general, finding the roots of the left-hand side in Eq. (18) is a nonlinear optimization problem that would require a numerical approach such as Newton's method, which is too computationally complex for most practical applications. In practice, a more efficient but suboptimal method can be used instead to obtain a meaningful solution. For example, expectation maximization (EM) is a paradigm that is known to converge to some local stable point of the log-likelihood function. EM assumes a framework in which we have an incomplete data space $\mathcal{X}$ and a complete (latent) data space $\mathcal{S}$ for which there is a many-to-one mapping from the complete data space to the incomplete data space. As such, the incomplete data is uniquely determined by the complete data. In our case here, the source snapshots $\{\mathbf{s}(k), \ldots, \mathbf{s}(k+K-1)\}$ form K observations of the complete data space having diagonal covariance matrix $\boldsymbol{\Sigma}$. The incomplete data is given by the many-to-one mapping $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$. With Gaussian sources, the likelihood functions of both spaces are given as

$$L_{cd}\left(\boldsymbol{\Sigma}|\mathbf{S}_s\right) = -K\log|\boldsymbol{\Sigma}| - \mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}\mathbf{S}_s\right] \tag{19}$$

and

$$L_{id}\left(\mathbf{R}_x|\mathbf{S}_x\right) = -K\log|\mathbf{R}_x| - \mathrm{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right]. \tag{20}$$

The EM algorithm is an iterative recipe with two basic steps:

1. The "E" step, where, at the current iteration, we compute the conditional expected value of the complete data with respect to the observed data $\mathbf{X}_k$ and the current estimate of the latent parameter $\boldsymbol{\Sigma}^{(p)}$,

$$E\left[L_{cd}(\boldsymbol{\Sigma})|\mathbf{X}_k, \boldsymbol{\Sigma}^{(k)}\right] = -K\log|\boldsymbol{\Sigma}| - \mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}E\left[\mathbf{S}_s|\mathbf{X}_k, \boldsymbol{\Sigma}^{(p)}\right]\right]. \tag{21}$$

2. The "M" step, where, $\boldsymbol{\Sigma}^{(p+1)}$ is computed by maximizing Eq. (21) with respect to $\boldsymbol{\Sigma}^{(p)}$,

$$\boldsymbol{\Sigma}^{(p+1)} = \arg\max_{\boldsymbol{\Sigma}^{(p)}}\left\{E\left[L_{cd}(\boldsymbol{\Sigma})|\mathbf{X}_k, \boldsymbol{\Sigma}^{(p)}\right]\right\}.$$

The work of Barton and Fuhrman [27] provided explicit computations of the above two steps, which can be boiled down to two convenient formulas:

$$\begin{aligned}\boldsymbol{\Sigma}^{(p+1)} &= \left(\boldsymbol{\Sigma}^{(p)} + \boldsymbol{\Sigma}^{(p)}\mathbf{B}^{(p)}\boldsymbol{\Sigma}^{(p)}\right) \odot \mathbf{I}\\ \mathbf{R}_x^{(p)} &= \mathbf{A}\boldsymbol{\Sigma}^{(p)}\mathbf{A} + \sigma_0^2\mathbf{I},\end{aligned}$$

where $\odot$ denotes the Hadamard product and

$$\mathbf{B}^{(p)} = \mathbf{A}^H \mathbf{R}_x^{-1(p)} \left(\mathbf{S}_x - \mathbf{R}_x\right) \mathbf{R}_x^{-1(p)} \mathbf{A}.$$

On the surface, this approach seems only applicable with MMV data, however there is no explicit requirement for this and the method extends to the single-snapshot case without any modification. Furthermore, it is important to note that the EM method also can be extended to handle non-Gaussian noise characterizations by replacing $\sigma_0^2 \mathbf{I}$ with a more general noise covariance matrix, $\mathbf{R}_n$.

2.3.4 Compressive Sensing

A number of recent DOA estimation algorithms have been based on the concept of compressive sensing (CS) [16, 28, 29] that are aimed specifically towards sparse arrays. Typically, a CS framework seeks to exploit sparsity as a regularization prior to recover physically meaningful signals from under-determined linear systems. For the DOA estimation problem, a sparsity prior can be assumed if the number of sources are coming from only a few directions relative to the total number of directions in a discretized search grid $\boldsymbol{\theta}$. It is important to note that while the other DOA estimation methods discussed above all use a discretized search grid for computing their solutions, the core assumptions of these algorithms are independent of the chosen grid. This is not the case in conventional CS-based methods, and as we discuss below, the choice of grid can have direct impact on underlying sparsity assumptions.

For the case of SMV data, it is known that the maximum number of sources that can be estimated by leveraging sparsity is $N/2 - 1$, about half of what can be expected with DOA estimation techniques such as MUSIC. More specifically, when the matrix $\mathbf{A}$ satisfies a Kruskal rank condition (Theorem 2 in Appendix D), it can be assured that the null space of $\mathbf{A}$ is free of vectors of a specified sparsity; thus, when restricting solutions to such a space, the under-determined problem becomes well-posed. Under a sparsity prior, DOA estimation is then ideally accomplished through a ℓ_0-based optimization program such as

$$\begin{aligned} &\min \|\mathbf{s}\|_0 \text{ subject to} \\ &\qquad \|\mathbf{A}\mathbf{s}(k) - \mathbf{x}(k)\| < \sigma_0^2. \end{aligned}$$

In practice, such ℓ_0 approaches are not feasible, since these programs are known to be non-convex and fall under the class of NP-hard problems (i.e., finding a solution polynomial time is not guaranteed). However, using an alternative based on the use of the ℓ_1 norm in place of the ℓ_0 norm has been recognized as a better approach. This is because not only are ℓ_1-based programs convex, and thus computational feasibility can be assured, but also under some mildly stricter conditions on the structure of $\mathbf{A}$, it can be shown that solutions admitted by this method correspond (or very closely correspond) to sparse solutions yielded by ℓ_0 techniques. Some example ℓ_1 recovery algorithms are the basis pursuit denoising (BPDN) approach,

$$\min \|\mathbf{s}\|_1 \text{ subject to} \tag{22}$$

$$\|\mathbf{A}\mathbf{s}(k) - \mathbf{x}(k)\|_2 < \sigma_0^2, \tag{23}$$

and its unconstrained equivalent form, the least absolute shrinkage and selection operator (LASSO) program,

$$\min \frac{1}{2} \|\mathbf{A}\mathbf{s}(k) - \mathbf{x}(k)\|_2^2 + \alpha \|\mathbf{s}\|_1. \tag{24}$$

Another class of sparsity-based approaches includes greedy methods, which are not based on structured convex optimization. Greedy methods such as orthogonal matching pursuit (OMP) are advantageous for larger-scale problems, as they are computationally more efficient than ℓ_1-based algorithms. However, any assurances of such techniques converging to optimal sparse solutions can be shown only under much stricter conditions on $\mathbf{A}$ [30].

When MMV data is available and is full rank, the maximum number of sources that potentially can be estimated through sparsity approaches becomes $N - 1$ (Appendix D, theorem 3), which is on par with conventional DOA estimation methods. For the MMV case, ℓ_1-based algorithms are based on a so-called joint-sparsity assumption, where (like the other techniques discussed here) the underlying source directions are fixed across the time span of the MMV data. A joint-sparsity version of LASSO, for example, is

$$\min_{\mathbf{S}\in\mathbb{C}^{M\times K}} \frac{1}{2}\|\mathbf{A}\mathbf{S}(k) - \mathbf{X}(k)\|_F^2 + \alpha\|\mathbf{S}\|_{1,r},$$

where $\|\cdot\|_F$ is the Frobenius norm and

$$\|\mathbf{S}\|_{1,r} = \sum_{m=1}^{M}\left(\sum_{k=1}^{K}|[\mathbf{S}]_{mk}|^2\right)^{1/2}.$$

This norm generalizes the concept of an ℓ_1 norm by viewing the temporal dimension of the underlying signal as a multi-component signal. Thus, the joint-sparsity LASSO penalizes an ℓ_1 norm across the spatial dimension of $\mathbf{S}$.

For either SMV or MMV sparsity-based DOA estimation, the gridding problem is one of the biggest drawbacks [16] of these approaches. As we mentioned, the matrix $\mathbf{A}$ relies on an explicit grid; however, the underlying assumed sparsity is taken with respect to this grid. In reality, the probability that the true DOAs align exactly with the chosen grid is zero. This is the case no matter how finely we sample our grid, and thus there will always be an imbalance between our assumed steering vectors and the true DOAs of the sources. This imbalance can be highly detrimental to the performance of sparsity-based methods. An explanation for this lies in the fact that the mismatched steering vectors leads to a violation of our assumed sparsity [31], since energy will be distributed across all the coefficients. Furthermore, as the number of columns in the matrix $\mathbf{A}$ increases, the steering vectors in the matrix $\mathbf{A}$ become more mutually coherent, which decreases the suitability of leveraging sparsity as an effective prior. Consequently, the maximum number of sources that algorithms like BP, LASSO, or OMP can identify reliably will decrease. Recent works on compressive DOA estimation have thus focused on so-called “gridless” methods. These methods are on a more general concept of signal parsimony rather than sparsity; in other words, the signal we wish to receive is somehow low-dimensional with respect to building blocks of atomic functions. This motivated the development of compressive sensing based on the use of the atomic norm [32], which can be seen as a generalization of the ℓ_1 norm for recovering sparse vectors or the nuclear norm for low-rank matrices. In essence, the work in [33] showed that through employing atomic norm minimization in the form of a semi-definite program (SDP), one can uniquely recover the underlying sources with overwhelming probability and with infinite precision! This is shown to be the case when the maximal spacing of the spatial frequencies satisfy

$$\Delta\nu > 4/N,$$

for the single-snapshot case. In other words, the maximal number of sources that can be recovered using such an approach is $N/4$, about half of what can be expected with a conventional ℓ_1 penalty under completely matched steering vectors. Thus, such approaches in addition to being computationally expensive, seem to imply a rather inefficient use of the DOF offered by the array.

3. SPARSE ARRAYS, THE DIFFERENCE COARRAY, AND THE AUGMENTED COVARIANCE MATRIX

In the previous chapter, we discussed a number of conventional spatial-processing algorithms that are commonly employed for ULAs. In particular, we showed that generally for the DOA estimation problem, estimating $N-1$ sources serves as an upper bound on the number of sources that can be estimated. This upper bound is one of the major drawbacks of sparse arrays, i.e., the lack of spatial DOF as compared to dense arrays. As discussed in Chapter 1, there are approaches based on the second-order statics of the array signals that allow for greater DOF to be extracted from certain sparse arrays. Two examples of such techniques that are prevalent in the current literature both rely on forming a so-called virtual uniform linear array (VULA) by exploiting covariance information. The first approach is based on augmenting the measured covariance matrix [4, 5] to estimate the larger measurement covariance matrix corresponding to the ULA of the same outer dimension as the sparse one. This augmented ULA is referred to as the VULA and thus allows for employing spatial processing algorithms that rely on covariance data to be employed with no special modification. The second approach, which is the main focus of this chapter, is based on viewing the central ULA segment of the difference coarray as a VULA in itself, and treating the corresponding covariance data as a signal received at this virtual array. Linear beamforming of this virtual signal, referred to as "correlation beamforming" in [3, Sec. IV], appears to be the earliest example of this approach. We will refer to the first approach as covariance augmentation and the second approach as NAP. This chapter proceeds as follows: We first consider linear beamforming of the covariance matrix of a sparse array, as this naturally leads to the derivation of the difference coarray as well as the assumptions that underlie and enable much of what is to come. We then compare this to the augmented-covariance approach, which uses the same underlying data and assumptions but in a different format. We use these results to compare some different classes of sparse arrays that are appropriate for extending their respective DOFs within the two frameworks discussed.

3.1 Linear Beamforming of the Covariance Matrix

Let us return to the N-element linear array signal model of Eq. (4) and its corresponding $N\times N$ covariance matrix of Eq. (5). We wish to apply a linear beamformer (filter) to the covariance, which we represent here as the $N \times N$ complex matrix $\mathbf{W}$. We can write the filter output as

$$\begin{aligned}\left\langle \mathbf{R}_x, \mathbf{W}\right\rangle &= \left\langle \mathbf{A}\mathbf{R}_s\mathbf{A}^H, \mathbf{W}\right\rangle + \left\langle \mathbf{R}_n, \mathbf{W}\right\rangle \\ &= \left\langle \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} \mathbf{a}_{\mathbb{S}}(\theta_m)\mathbf{a}_{\mathbb{S}}^H(\theta_n)[\mathbf{R}_\mathbf{s}]_{m,n}, \mathbf{W}\right\rangle + \left\langle \mathbf{R}_n, \mathbf{W}\right\rangle \\ &= \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} \mathbf{a}_{\mathbb{S}}^H(\theta_n)\mathbf{W}^H\mathbf{a}_{\mathbb{S}}(\theta_m)[\mathbf{R}_\mathbf{s}]_{m,n} + \left\langle \mathbf{R}_n, \mathbf{W}\right\rangle, \end{aligned} \tag{25}$$

where the third line uses the identity $\left\langle \mathbf{a}\mathbf{b}^H, \mathbf{W}\right\rangle = \mathbf{b}^H\mathbf{W}\mathbf{a}$. From this, we see that the quantity $\mathbf{a}_{\mathbb{S}}^H(\theta_n)\mathbf{W}^H\mathbf{a}_{\mathbb{S}}(\theta_m)$ determines the gain applied to the signal cross-covariance term $[\mathbf{R}_\mathbf{s}]_{m,n}$. Defining the two-input *array cross-pattern* function

$$\mathcal{W}(\theta,\theta') \triangleq \mathbf{a}_{\mathbb{S}}^H(\theta')\mathbf{W}^H\mathbf{a}_{\mathbb{S}}(\theta), \tag{26}$$

we see that it serves as the second-order equivalent of the linear array pattern $\mathbf{w}^H \mathbf{a}_{\mathbb{S}}(\theta)$. Indeed, if $\mathbf{W}$ is defined as the rank-1 outer product $\mathbf{w}\mathbf{w}^H$ then we have

$$\begin{aligned} \mathcal{W}(\theta, \theta') &= \mathbf{a}_{\mathbb{S}}^H(\theta')\mathbf{w}\mathbf{w}^H \mathbf{a}_{\mathbb{S}}(\theta) \\ &= (\mathbf{w}^H \mathbf{a}_{\mathbb{S}}(\theta'))^*(\mathbf{w}^H \mathbf{a}_{\mathbb{S}}(\theta)), \end{aligned}$$

the product of linear beamformer responses in the two directions. Substituting the steering-vector expression Eq. (3) into Eq. (26) yields

$$\begin{aligned} \mathcal{W}(\theta, \theta') &= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} w_{m,n}^* \, e^{-j\frac{2\pi}{\lambda}(x_n \sin(\theta') - x_m \sin(\theta))} && (27) \\ &= W^*(\sin(\theta)/\lambda, -\sin(\theta')/\lambda), && (28) \end{aligned}$$

where W is the two-dimensional spatial Fourier transform of filter $\mathbf{W}$.

The cross-pattern function is linear in the N^2 coefficients of $\mathbf{W}$ and thus represents N^2 degrees of freedom, compared to N for the linear beamformer. On the other hand, M sources can result in up to M^2 cross-terms in the signal covariance $\mathbf{R}_\mathbf{s}$, and so in general, a beamformer based on second-order statistics cannot resolve any more sources than a linear one. It is only when some of the sources are uncorrelated, so that their corresponding cross-terms are zero, that these degrees of freedom can be used to resolve additional sources. Indeed, a most-common case is to assume that all of the sources are uncorrelated. Let us revisit Eq. (25), assuming that $[\mathbf{R}_\mathbf{s}]_{m,n} = 0$ for $m \neq n$:

$$\begin{aligned} \langle \mathbf{R}_x, \mathbf{W} \rangle &= \sum_{m=0}^{M-1} \mathcal{W}(\theta_m, \theta_m)[\mathbf{R}_\mathbf{s}]_{m,m} + \langle \mathbf{R}_n, \mathbf{W} \rangle \\ &= \sum_{m=0}^{M-1} \mathcal{C}(\theta_m)[\mathbf{R}_\mathbf{s}]_{m,m} + \langle \mathbf{R}_n, \mathbf{W} \rangle. && (29) \end{aligned}$$

Now we can see that the cross-pattern derived in the previous section has embedded within it the array response to such signals by evaluating it at the same angle for both arguments:

$$\mathcal{C}(\theta) \triangleq \mathcal{W}(\theta, \theta) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} w_{m,n}^* \, e^{-j\frac{2\pi}{\lambda}(x_n - x_m)\sin(\theta)}. \quad (30)$$

We see that this diagonal cut through the cross-pattern is the Fourier transform of the sequence of values $\{w_{m,n}^*\}$ on the locations $\{\mathbf{x}_n - \mathbf{x}_m\}$ representing all of the pairwise differences of array element positions. The set of pairwise differences is known as the *difference coarray* [3], and it represents the locations of elements in a virtual array whose pattern $\mathcal{C}$ we will call the *coarray pattern*.

Comparing Eqs. (29) and (30) to Eqs. (8) and (9), we see that the signal term of Eq. (29) is analogous to that of a conventional linear beamformer, with the signal powers $\{[\mathbf{R}_\mathbf{s}]_{m,m}\}$ taking the place of complex signal amplitudes $\{\tilde{s}_m\}$, and the coarray pattern taking the place of the linear array pattern. Since the coarray is larger and less sparse than the physical array, second-order beamforming can provide increased resolution for uncorrelated sources. More generally, (Eq.) (29) implies that in appropriate contexts, we can treat the covariance of the linear array as a virtual "signal," itself, with support on the coarray. This idea will be formalized and developed further in the following sections.

3.2 Sparse Array Architectures

There are many ways to arrange sparse array elements, but in this report, we are specifically interested in sparse array architectures that admit a full (or nearly full) difference coarray. The hypothesis is that these types of sparse arrays would contain the covariance information captured by a ULA of the same length. Before we discuss specific examples of such arrays, we formalize our definition of the difference coarray and introduce a few related concepts.

Definition 1 (Difference Coarray)**.** *Let $\mathbb{S}$ be the set of relative antenna locations. The set of all spatial lags associated with $\mathbb{S}$ is defined as*

$$\mathbb{D}_l = \{n_i - n_j \mid n_i, n_j \in \mathbb{S}\}$$

and the difference coarray is the set $\mathbb{D}_c$ containing all unique elements of $\mathbb{D}_l$.

Note in this definition we allow the set $\mathbb{D}_l$ to have repeated elements. An important property of both the sets $\mathbb{D}_l$ and $\mathbb{D}_c$ is that their elements are symmetric about zero. In other words, if $m \in \mathbb{D}_l$, then $-m \in \mathbb{D}_l$ as well, and the same holds for $\mathbb{D}_c$. the following definitions aid in developing a means to measure how complete a certain sparse array is in reference to a ULA:

Definition 2 (Central ULA Segment)**.** *Let $\mathbb{D}_c$ be the difference coarray associated with an array $\mathbb{S}$, and let $\tilde{N}$ be the largest integer in $\mathbb{D}_c$ such that the set $\mathbb{U} = \{0, \pm 1, \ldots, \pm \tilde{N}\}$ is also contained in $\mathbb{D}_c$. Then we say that $\mathbb{U}$ is the central ULA segment of $\mathbb{D}_c$.*

We say that a certain N-element sparse array is *fully augmentable* when its central ULA segment coincides with its coarray. In other words, there are no gaps in $\mathbb{D}_c$. The second measure is related to the amount of redundancy captured by a given array.

Definition 3 (Weight Function)**.** *For a given N-element array with spatial lags $\mathbb{D}_l$ and coarray $\mathbb{D}_c$, we define the weight function $w : \mathbb{D}_c \longrightarrow \mathbb{Z}^+$ as*

$$w(m) = |\{m \in \mathbb{D}_l\}|. \tag{31}$$

The weight function is essentially a histogram of the spatial lags contained in the coarray giving a count of repeated spatial lags. It is not difficult to show that the weight function adheres to the following properties:

The weight function of some arbitrary array $\mathbb{S}$ with $|\mathbb{S}| = N$ satisfies:

1. $w(0) = N$.
2. $1 \leq w(m) \leq N$, for all $m \in \mathbb{D}_c$.

3. $w(m) = w(-m)$, for all $m \in \mathbb{D}_c$.

4. $\sum_{m\in\mathbb{D}_c\setminus\{0\}} w(m) = N(N-1)$.

In particular, properties 1, 2, and 4 imply we can aim to achieve a longer central ULA segment of a particular array by minimizing the amount of redundancy in $\mathbb{D}_l$. In fact, a completely non-redundant array would yield the weight function

$$w_{\text{nonredundant}}(m) = \begin{cases} N, & \text{for } m = 0 \\ 1, & \text{otherwise} \end{cases}, \tag{32}$$

which yields a central ULA segment of length $N(N-1)+1$. Interestingly, one can show that any ULA is completely redundant and always admits a central ULA segment of length $2N-1$. As a simple example, the non-redundant 4-element array given as $\mathbb{S} = \{0, 1, 4, 6\}$ admits a central ULA segment that coincides with a 7-element ULA, which we can see in Fig. 4.

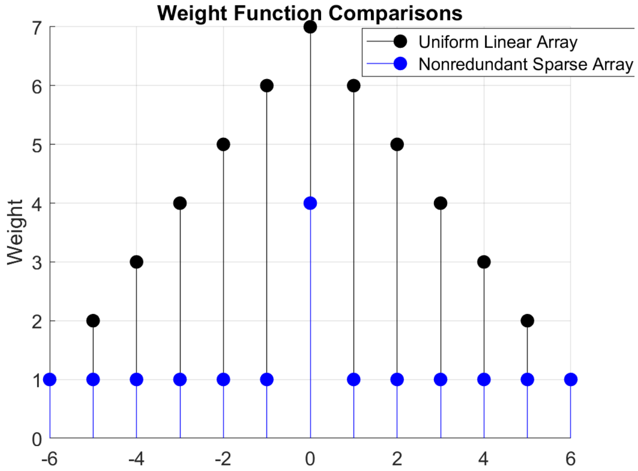


Fig. 4—Comparing the weight functions of a 4-element non-redundant array with a 7-element ULA of equivalent length

We now can discuss specific sparse architectures that admit favorable characteristics in terms of redundancy and the length of their central ULA segments. One of the earliest sparse architectures is a minimum-redundancy array (MRA) [2, 34] in which N elements are positioned to yield as few redundancies as possible but subject to a fully augmentable constraint. Since the number of redundancies is minimized, as we discussed, this results in the maximal length of the coarray $\tilde{N}$ and is ensured to have no gaps; i.e., its central ULA segment coincides with $\mathbb{D}_c$. The element positions satisfying this objective and constraint typically are found through computational optimization algorithms. Another relevant sparse architecture is a minimum-hole array (MHA) [35], which relaxes the no-gap constraint of MRAs. In other words, MHAs have no redundancies but are allowed to have holes with the aim of keeping the number of gaps to a minimum. As such, it admits an array that has the minimum aperture possible without introducing redundancy. Like MRAs, there are no closed-form formulas for finding the optimal element placing. Nested [11] and coprime arrays [36] are more recent sparse architectures, both of which have explicit formulas for placing N elements; however this comes at a cost of not being able to adhere to the minimal redundancy or hole criteria as is done with MRAs and MHAs. Nested arrays can be thought of as the concatenation of two ULAs: a dense one and a sparse one. Coprime arrays are composed of two sparse arrays: one that is of length N_1 and one that is of length N_2, where $N_1 < N_2$ and such that they are coprime integers. Both of these structures have the advantage of closed-form formulas for determining the element positions, which could be important for applications where one needs to reconfigure the array elements rapidly. It is claimed in [37] that coprime arrays admit a reduced mutual coupling between sensor elements as compared to nested arrays, which could be the case, given that coprime arrays have fewer immediately adjacent elements, but this is not substantiated. Figure 5 illustrates an example of all four architectures for the case of $N = 6$ and their respective difference coarrays. Notice that in contrast with coprime arrays, nested arrays are fully augmentable.

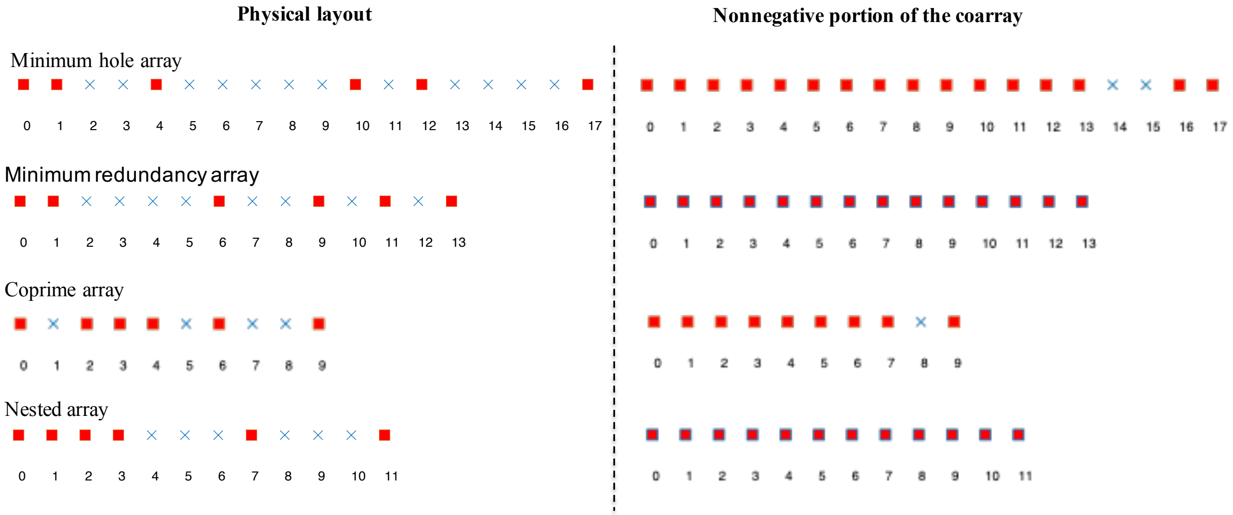


Fig. 5—Illustration of the element positioning of the four different sparse arrays discussed for the case of $N = 6$ as well as the non-negative portions of their respective difference coarrays

3.3 Extending the DOF of Sparse Arrays through the Difference Coarray

For simplicity of the discussion, we will assume throughout this section (and the remainder of the report) that the sparse arrays under consideration induce a filled coarray without gaps, which corresponds to the "restricted" class in [2]. While much of what we discuss also could apply to non-fully augmentable arrays (such as coprime arrays) by either focusing solely on its central ULA or through some interpolation scheme, we choose to focus on the fully augmentable case to focus on the conceptual aspects of the approach.

3.3.1 Covariance Matrix Augmentation

To get more specific about the two approaches, let $\tilde{N} \leq (N)(N-1)+1$ denote the length of the central ULA segment of a given sparse array $\mathbb{S}$. Covariance augmentation seeks to estimate a covariance matrix $\tilde{\mathbf{R}}_x$ of size $\frac{(\tilde{N}+1)}{2} \times \frac{(\tilde{N}+1)}{2}$ corresponding to the $\frac{(\tilde{N}+1)}{2}$ ULA. Since the sparse array captures all possible spatial lags of this ULA, we can construct an initial guess $\tilde{\mathbf{S}}_x$ for $\tilde{\mathbf{R}}_x$ by filling in the covariance data corresponding to the lags $m = 0, \ldots, \frac{(\tilde{N}-1)}{2}$. Note that repeated elements can be filled by an averaging scheme. However, this resulting matrix does not necessarily admit the properties of a covariance matrix, mainly that of being Toeplitz and being positive semidefinite. To ensure these crucial properties, one then can find the closest Hermitian, Toeplitz, and positive definite $\frac{(\tilde{N}+1)}{2} \times \frac{(\tilde{N}+1)}{2}$ matrix to this initial estimate. Let us denote the vector space $\mathbb{H}^{(\tilde{N}+1)/2}$ as the space containing all Hermitian matrices of size $\frac{(\tilde{N}+1)}{2} \times \frac{(\tilde{N}+1)}{2}$. Note that $\mathbb{H}^{(\tilde{N}+1)/2}$ in fact constitutes a Hilbert space endowed with the inner product $\langle \mathbf{A}, \mathbf{B} \rangle_{\mathbb{H}} = \mathrm{tr}\,[\mathbf{AB}]$ for all $\mathbf{A}, \mathbf{B} \in \mathbb{H}^{(\tilde{N}+1)/2}$. Now define the sets $\mathbb{P}^{(\tilde{N}+1)/2} \subset \mathbb{H}^{(\tilde{N}+1)/2}$ and $\mathbb{T}^{(\tilde{N}+1)/2} \subset \mathbb{H}^{(\tilde{N}+1)/2}$ as

$$\mathbb{P}^{(\tilde{N}+1)/2} = \{\mathbf{A} \in \mathbb{H}^{(\tilde{N}+1)/2} \mid \mathbf{x}^H \mathbf{A} \mathbf{x} \geq 0\}, \text{ for all } \mathbf{x} \in \mathbb{C}^{(\tilde{N}+1)/2} \tag{33}$$

and

$$\mathbb{T}^{(\tilde{N}+1)/2} = \{\mathbf{A} \in \mathbb{H}^{(\tilde{N}+1)/2} \mid \mathbf{A} \text{ is Toeplitz}\}. \tag{34}$$

In plain English, $\mathbb{P}^{(\tilde{N}+1)/2}$ is the set of Hermitian positive semidefinite matrices and $\mathbb{T}^{(\tilde{N}+1)/2}$ is the set of Hermitian Toeplitz matrices. It is not hard to establish the following:

1. The sets $\mathbb{P}^{(\tilde{N}+1)/2}$ and $\mathbb{T}^{(\tilde{N}+1)/2}$ are convex.
2. The sets $\mathbb{P}^{(\tilde{N}+1)/2}$ and $\mathbb{T}^{(\tilde{N}+1)/2}$ are closed.
3. The intersection $\mathbb{M}^{(\tilde{N}+1)/2} = \mathbb{P}^{(\tilde{N}+1)/2} \cap \mathbb{T}^{(\tilde{N}+1)/2}$ is a closed and convex set.

It is known that for any closed and convex subset $\mathbb{W}$ of a Hilbert space $\mathbb{H}$, for every element $u \in \mathbb{H}$, there exists a *unique best approximation* $w \in \mathbb{W}$ to u. This is known as the projection of u onto the set $\mathbb{W}$ and is denoted as $\mathrm{proj}_{\mathbb{W}}(u)$. For our specific case, we seek the projection of $\tilde{\mathbf{S}}_x$ onto the set $\mathbb{M}^{(\tilde{N}+1)/2}$. This can be solved formally as

$$\mathrm{proj}_{\mathbb{M}^{(\tilde{N}+1)/2}}\left(\tilde{\mathbf{S}}_x\right) = \arg \inf_{\mathbf{R} \in \mathbb{H}^{(\tilde{N}+1)/2}} \|\mathbf{R} - \tilde{\mathbf{S}}_x\| \text{ subject to } \mathbf{R} \in \mathbb{M}^{(\tilde{N}+1)/2}. \tag{35}$$

This can be solved efficiently using Dykstra's alternating projection onto convex sets (POCS) algorithm [38]. Once a solution is obtained, we can apply conventional DOA estimation techniques with respect to the estimated larger covariance data $\tilde{\mathbf{R}}_x = \mathrm{proj}_{\mathbb{M}^{(\tilde{N}+1)/2}}\left(\tilde{\mathbf{S}}_x\right)$. Based on our discussion of such algorithms, it is clear, then, the maximum number of sources that we can expect to estimate via this approach is $(\tilde{N}-1)/2$.

3.3.2 The NAP Virtual Array Model

In the NAP approach, the VULA is obtained directly by vectorizing the covariance model given in Eq. (5). As discussed in Appendix A, we can define a transformation $\mathrm{VEC}(\cdot) : \mathbb{C}^{N \times N} \to \mathbb{C}^{N^2 \times 1}$ that maps the N^2 variables $\{R_{11}, R_{21}, \ldots, R_{NN}\}$ (columnwise order) into the vector $\mathbf{z} = [z_1, z_2, \ldots, z_{N^2}]^T$. Applying $\mathrm{VEC}(\cdot)$ to both sides of Eq. (5) and after some algebraic manipulation, we can derive the (redundant) NAP VULA model

$$\mathbf{z} = (\mathbf{A}^* \circledast \mathbf{A})\, \mathbf{p} + \sigma_0^2 \mathbf{1}. \tag{36}$$

Here, $\mathbf{p} = [\sigma_1^2, \ldots, \sigma_M^2]^T]$, $\mathbf{1} = \text{VEC}\,(\mathbf{I})$, and $\circledast$ is the Khatri-Rao product. Due to the multiplicative property of exponential functions, the matrix $(\mathbf{A}^* \circledast \mathbf{A})$ resembles the structure of an array manifold corresponding to an array whose antenna element positions coincide with the set $\mathbb{D}_l$. Denoting this (virtual) array manifold as $\mathbf{A}_{\mathbb{D}_l}$ we can write Eq. (36) more succinctly as

$$\mathbf{z} = \mathbf{A}_{\mathbb{D}_l}\mathbf{p} + \sigma_0^2\mathbf{1}. \tag{37}$$

Our desire is to transform the model Eq. (37) into one that further resembles a physical array. This can be accomplished by sorting the spatial lags and then removing the redundant elements of D_l according to some predefined rule (such as retaining only the first occurrence), or more formally by defining a (nonsquare) permutation matrix $\mathbf{P} \in \mathbb{R}^{\tilde{N} \times N^2}$ that implements the sorting and selection.

Example 1. *Consider the array $\mathbb{S} = \{0, 1, 2, 5\}$ that admits the coarray*

$$\mathbb{D}_l = \{0, 1, 2, 5, -1, 0, 1, 4, -2, -1, 0, 3, -5, -4, -3, 0\}.$$

The covariance matrix associated with the array has the structure

$$\mathbf{R}_x = \begin{bmatrix} r(0) & r(-1) & r(-2) & r(-5) \\ r(1) & r(0) & r(-1) & r(-4) \\ r(2) & r(1) & r(0) & r(-3) \\ r(5) & r(4) & r(3) & r(0) \end{bmatrix}$$

with $r(j)$ being the covariance data for spatial lag $j \in \mathbb{D}_l$. The permutation matrix

$$\mathbf{P} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \tag{38}$$

maps the vector

$$\mathbf{z} = \mathit{VEC}\,(\mathbf{R}_x) = [r(0), r(1), r(2), r(5), \ldots, r(-3), r(0)]^T$$

into the vector

$$\tilde{\mathbf{z}} = [r(-5), r(-4), \ldots, r(0), \ldots, r(5)]^T$$

containing no redundant elements.

If we apply this operator to both sides of Eq.(36), we have

$$\mathbf{Pz} = \tilde{\mathbf{z}} = \mathbf{A}_{\mathbb{D}_c}\mathbf{p} + \sigma_0^2\mathbf{1}_c, \tag{39}$$

where $\mathbf{A}_{\mathbb{D}_c} = \mathbf{PA}_{\mathbb{D}_l}$ is the array manifold corresponding to the array elements taken from the coarray $\mathbb{D}_c$, and $\mathbf{1}_c = \mathbf{P1}$. When the array is fully augmentable, this form of NAP then resembles a uniform linear array with unique antenna elements distributed across the aperture $-(\tilde{N}-1)/2, \ldots, (\tilde{N}-1)/2$.

As we begin to analyze some characteristics of the NAP VULA in Chapter 4, we need a formal means of mapping back and forth between the various covariance structures in Eqs. (5), (37), and (39). For mappings between the covariance model Eq. (5) to the redundant coarray data in Eq. (37), this is not an issue because the operator VEC $(\cdot)$ is a bijection (one to one and onto mapping) between $\mathbb{C}^{N\times N}$ and $\mathbb{C}^{N^2\times 1}$ and also preserves the Euclidean distance between points in the respective spaces.

Proposition 1. *The transformation* $\mathbf{z} = VEC\left(\mathbf{R}_x\right)$ *is a linear isometry.*

Proof. Clearly, VEC $(\cdot)$ is linear and by definition, we have

$$\frac{\partial\left[\mathbf{z}\right]_{i'}}{\partial\left[\mathbf{R}\right]_{ij}} = \begin{cases} 1, & \text{for } i' = (j-1)N + i \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

for each $i' = 0, \ldots, N^2 = 1$. Hence the Jacobian matrix $\mathbf{J}_{\text{vec}} = \mathbf{I}$, where $\mathbf{I}$ is a $N^2 \times N^2$ identity matrix having a unity determinant.

It follows from Proposition 1 that a well-defined inverse for VEC $(\cdot)$ exists, which we denote as MAT $[\cdot]$. It's easy to observe that this operator simply reverses the vectorization operator by reshaping a N^2 vector back to its $N \times N$ matrix form. The operator $\mathbf{P} : \mathbb{C}^{N^2} \to \mathbb{C}^{\tilde{N}}$, on the other hand, complicates matters because in general it is not invertible, as we are mapping a N^2 vector down to a lower dimensional space $\mathbb{C}^{\tilde{N}}$. In other words, $\mathbf{P}$ has a null space $N(\mathbf{P})$ associated with it, and thus an ambiguity exists when attempting to recover a vector $\mathbf{z}$ from the data $\tilde{\mathbf{z}} = \mathbf{Pz}$. However, we can get around this complication by restricting the domain of the $\mathbf{P}$ operator. More specifically, for NAP, we effectively are applying the VEC $(\cdot)$ operator with respect to the space of Hermitian positive semi-definite and Toeplitz matrices $\mathbb{M}^N$. Furthermore, in practice, if we assume that underlying array in question is fixed, then the redundant elements in the covariance matrices are always known *a priori*. Thus, the covariance data in this case belongs to a sub-space $\mathbb{V}_{\mathbb{D}_l} \subset \mathbb{M}^N$ consisting of matrices $\mathbb{M}^N$ whose individual elements follow the redundancies dictated by $\mathbb{D}_l$. With this in mind, let us denote the space $\mathcal{X}$ as the range of of VEC $(\cdot)$ when its domain is restricted to $\mathbb{V}_{\mathbb{D}_l}$. Thus, the permutation matrix $\mathbf{P}$ now can be seen as a mapping between $\mathcal{X}$ and its range under such a restriction that we can denote as $\mathcal{Y} \subset \mathbb{C}^{\tilde{N}}$. Under this restriction, $\mathbf{P}$ still has a null space associated with it, but now $N(\mathbf{P})$ can be characterized explicitly. If we let the set $\mathbb{S}_r \subset \{1, 2, \ldots, N^2\}$ be the set of vector indices corresponding to the redundant virtual elements of the coarray data $\mathbf{z}$ (for a fixed array structure), then the null space of $\mathbf{P}$ becomes

$$N(\mathbf{P}) = \{\mathbf{z} \in \mathcal{X} : z_i = 0, \forall i \notin \mathbb{S}_r\}. \tag{41}$$

Thus, for any vector $\mathbf{z} \in N(\mathbf{P})$, we can represent it as

$$\mathbf{z} = \sum_{i \in \mathbb{S}_r} a_i \hat{\mathbf{e}}_i, \tag{42}$$

where $\hat{\mathbf{e}}_i$ is the standard basis vector consisting of zeros except for a one at the i^{th} position, and $a_i \in \mathbb{C}$ is some weighting coefficient. With this in mind, we now can proceed to define a regularized a regularized inverse of the $\mathbf{P}$ operator by leveraging this characterization. For a linear operator to be invertible, we need to establish a right inverse as well as a left inverse. A right inverse of $\mathbf{P}$ is established readily by noting the following:

Proposition 2. *The permutation matrix* $\mathbf{P} : \mathbb{C}^{N^2} \rightarrow \mathbb{C}^{\tilde{N}}$ *is many-to-one and onto.*

Proof. It is clear that the rows of $\mathbf{P}$ only consists of the standard basis vectors $\hat{\mathbf{e}}_i$ and thus $\mathbf{P}$ has full-row rank and whose range spans $\mathbb{C}^{\tilde{N}}$.

Thus, we are assured of a right inverse $\mathbf{P}^{\#} : \mathbb{C}^{\tilde{N}} \rightarrow \mathbb{C}^{N^2}$, such that $\mathbf{P}\mathbf{P}^{\#} = \mathbf{I}_{\tilde{N}}$. A common and useful right-inverse is the minimum-norm solution, which in this case is shown to be

$$\mathbf{P}^{\#} = \mathbf{P}^T \left(\mathbf{P}\mathbf{P}^T\right)^{-1} = \mathbf{P}^T, \tag{43}$$

which follows from the orthonormality of the rows of $\mathbf{P}$. It is not hard to see that this choice of right inverse places zeros for the indices corresponding to the elements that did not get sampled by the forward operator $\mathbf{P}$. The minimum norm right inverse $\mathbf{P}^T$ then would place zeros in exactly these positions. Under the restriction of $\mathbf{P}$ to the domain $\mathcal{X}$, a left inverse then can be established (and thus recovering the redundant elements) by adding a fixed $N^2 \times \tilde{N}$ permutation matrix $\mathbf{P}_r$ to the minimum norm solution $\mathbf{P}^T$, which places the repeated elements back to their original locations. More specifically, the matrix $\mathbf{P}_r$ contains a complete basis for $N(\mathbf{P})$. One can show that such a basis of the null space forms an orthogonal compliment to $R(\mathbf{P}^T)$, and thus we have $\mathcal{X} = R(\mathbf{P}^T) \bigoplus N(\mathbf{P})$. As such, we can define the operator $\mathbf{P}^{\#} = \mathbf{P}^T + \mathbf{P}_r$, which serves as a right inverse on the entire space $\mathbb{C}^{\tilde{N}\times 1} \rightarrow \mathbb{C}^{N^2 \times 1}$ and also as a left inverse for $\mathcal{Y} \rightarrow \mathcal{X}$. Using these arguments, we can make a formal conclusion:

Proposition 3. *Let* $\mathbf{P}$ *be a* $\tilde{N} \times N^2$ *permutation matrix restricted to the domain* $\mathcal{X}$ *with range* $\mathcal{Y} \subset \mathbb{C}^{\tilde{N}}$. *There exists a left inverse* $\mathbf{P}^{\#} : \mathcal{Y} \rightarrow \mathcal{X}$ *such that for every* $\mathbf{z} \in \mathcal{X}$, *we have* $\mathbf{P}^{\#}\mathbf{P}\mathbf{z} = \mathbf{z}$.

Consider a concrete example of this as a continuation of Example 1:

Example 2. *For the array* $\mathbb{S} = \{0, 1, 2, 5\}$ *with the permutation matrix* $\mathbf{P}$ *defined in Example 1, we can define a left inverse from* $\mathcal{Y}$ *to* $\mathcal{X}$ *as*

$$\mathbf{P}^{\#} = \mathbf{P}^T + \mathbf{P}_r \tag{44}$$

with

$$\mathbf{P}_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{45}$$

3.3.3 The Noisy NAP VULA Models

Note that even though the physical array adheres to a noisy linear model, theoretically, the NAP VULA model here is noiseless because the covariance matrix is deterministic. Again, in practice, we do not have perfect knowledge of $\mathbf{R}_x$ but its estimate, $\mathbf{S}_x$, the sample covariance matrix. Therefore, there is an error matrix $\mathbf{N}_x$ satisfying

$$\mathbf{S}_x = \mathbf{R}_x + \mathbf{N}_x. \tag{46}$$

This error term is incorporated into the NAP model by introducing the quantity $\boldsymbol{\nu} = \text{VEC}\left(\mathbf{N}_x\right)$ and adding it as an error term to Eq. (37)

$$\mathbf{z} = \mathbf{A}_{\mathbb{D}_l}\mathbf{p} + \sigma_0^2\mathbf{1} + \boldsymbol{\nu}. \tag{47}$$

Furthermore, we can define $\tilde{\boldsymbol{\nu}} = \mathbf{P}\text{VEC}\left(\mathbf{N}_x\right)$ to serve as a noise term for Eq. (39),

$$\tilde{\mathbf{z}} = \mathbf{A}_{\mathbb{D}_c}\mathbf{p} + \sigma_0^2\mathbf{1}_c + \tilde{\boldsymbol{\nu}}. \tag{48}$$

We can make Eq. (48) a bit more succinct by augmenting the virtual manifold as $\mathbf{A}_v = \left[\mathbf{1}_c, \mathbf{A}_{D_c}\right]$,

$$\tilde{\mathbf{z}} = \mathbf{A}_v\mathbf{p} + \tilde{\boldsymbol{\nu}}. \tag{49}$$

Note in Eq. (49) the source signal is now defined as $\mathbf{p} = [\sigma_0^2, \ldots, \sigma_M^2]^T$. In both the redundant and nonredundant models, Eqs. (47) and (49), the virtual array data is now a random variable where the source of noise is strictly due to finite time sample support.

4. NONLINEAR ARRAY PROCESSING

In this chapter, we examine techniques, performance, and limitations of spatial processing with respect to the nonlinear model. We take a deep dive into the fundamental questions that we discussed in Chapter 1, which include establishing upper bounds on the number of sources that can be estimated through this framework. We also give some analysis into the expected time sample support needed to leverage the DOF of the sparse array effectively. We then proceed to discuss how to adapt some of the existing DOA estimation and adaptive filtering algorithms we discussed in Chapter 2 to the nonlinear model. We conclude the chapter with some simulations as a proof of concept of the discussed techniques.

4.1 DOA Estimation via the NAP Model

We begin our analysis by considering the DOA estimation problem. One of the first observations to note for the nonlinear model (39) is that the associated VULA is $\tilde{N}$ elements long. From Fig. 6, we see that this is twice as long as the VULA associated with the augmentation approach as well as the physical array, itself. This observation motivated the claim in [11] that NAP potentially could estimate up to $\tilde{N}-1$ sources as opposed to the conventional limitation of $(\tilde{N}-1)/2$ through augmentation. Additionally, the longer virtual aperture implies better resolution performance, which, for example, we can see in Fig. 7.

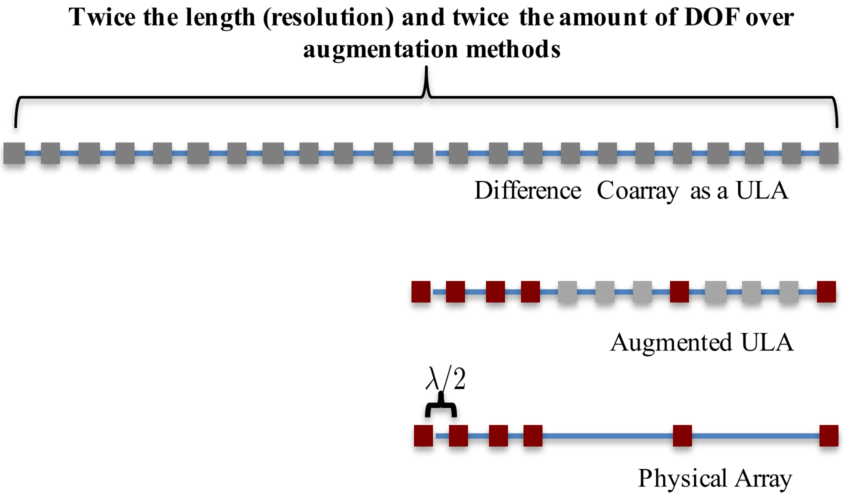

Twice the length (resolution) and twice the amount of DOF over augmentation methods
Difference Coarray as a ULA
Augmented ULA
$\lambda/2$
Physical Array

Fig. 6—Demonstrating the VULA generated through augmentation and NAP. The NAP VULA is twice as long as an augmented ULA.

Now let us consider establishing bounds on the maximum number of sources that can be estimated via NAP. Since in the nonlinear model the given data $\mathbf{R}_x$ or $\mathbf{S}_x$ constitutes a single virtual snapshot of $\tilde{\mathbf{z}}$, as discussed in Chapter 2, DOA estimation algorithms such as beamforming-based or CS techniques are the most straightforward to apply because they are suitable for SMV data without major modification. However, as we also have discussed, SMV methods are able to estimate only up to $(\tilde{N}-1)/2$ sources. In other words, CS-based or beamforming techniques would not be able to resolve any more sources than an augmentation approach. On the other hand, in the linear model, we know that correlation-based DOA estimation algorithms are able to estimate twice the number of sources that such direct methods could on the presumption we could

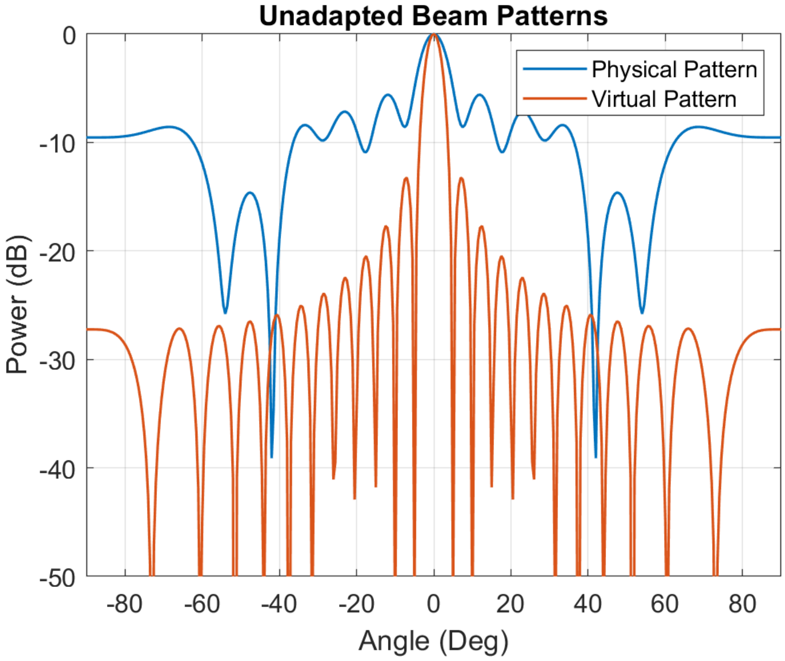


Fig. 7—Virtual beam pattern for a 6-element nested array. The virtual pattern exhibits a narrower main lobe, which is due to its larger aperture.

estimate the data covariance matrix. So the question we ask here is whether this would be the case with the nonlinear model if we somehow were able to somehow estimate $\mathbf{R}_z$. One possibility of doing this, of course, is to collect multiple snapshots of $\tilde{\mathbf{z}}$, which would imply an observation time equivalent to an entire dwell. There are many issues with attempting to use an entire dwell to perform DOA estimation with NAP. Firstly, for $\mathbf{R}_z$ to resemble the covariance structure seen in the linear model, this approach would require the source amplitudes to be quasi-stationary throughout the dwell, meaning a WSS stationary assumption would need to hold during a single CPI with some statistical fluctuation across the CPIs. Moreover, the DOAs, themselves, would need to be stationary/constant during the entire dwell, which, as we mentioned before, is a poor assumption to make for most fielded radar applications. A more practical approach is to consider correlation-based techniques with a restriction to SMV data. One major roadblock for the SMV approach is that $\tilde{\mathbf{z}}\tilde{\mathbf{z}}^H$ only provides a rank-one estimate of $\mathbf{R}_z$ and thus is rank-deficient for DOA estimation. Pal et al [11] attempted to get around the rank-deficiency issue through a spatial smoothing scheme. In this approach, an estimate $\hat{\mathbf{R}}_z$ is computed through spatial averaging across $(\tilde{N}+1)/2$ overlapping sub-arrays of the VULA (see Fig. 8),

$$\hat{\mathbf{R}}_z = \frac{2}{\tilde{N}+1} \sum_{i=1}^{(\tilde{N}-1)/2} \mathbf{z}_i \mathbf{z}_i^H ,$$

where $\mathbf{z}_i$ is the i^{th} subarray. The drawback with the spatial smoothing approach is that the NAP VULA has been decimated by half. Thus, again, we see that $(\tilde{N}-1)/2$ serves as an upper bound for the number of sources that can be estimated, and furthermore, we lose the ability to resolve sources beyond a ULA aperture of $(\tilde{N}-1)/2$ elements in length. The obvious question at this point is whether $(\tilde{N}-1)/2$ constitutes some sort of inherent limit for the under determined DOA estimation problem, or is this simply a specific

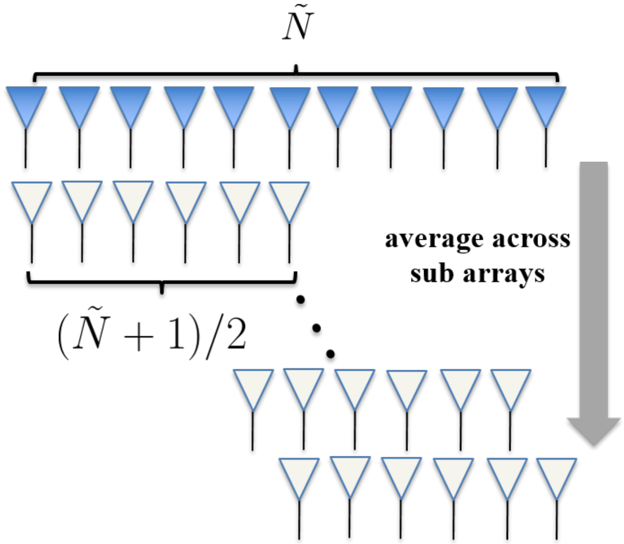


Fig. 8—The spatial smoothing scheme on the coarray

limitation of the spatial smoothing approach? The second issue that we need to address for the nonlinear model is the issue of time sample support of the physical measurements. In other words, in the virtual SMV DOA estimation approach, we still require a number of snapshots K to form a single virtual measurement. Numerical simulations seem to indicate that a relatively large number of snapshots are assumed when estimating more sources than elements via spatial smoothing and the nonlinear model. Older papers on the covariance augmentation approach [6] also indicated that estimation of sources beyond the number of elements relied on more snapshots to reduce the estimation error. Since, in the available literature to date, spatial smoothing has been the only algorithm considered for estimating $\mathbf{R}_z$ for the nonlinear model, it is not clear whether this is a specific limitation of spatial smoothing or the manifestation of a more general implication of DOA estimation with sparse arrays.

Our plan toward gaining a better understanding of the two fundamental questions we raised above for NAP DOA estimation is to develop a statistical characterization of the virtual array data using the common assumptions that usually are taken in analyzing DOA estimation performance with the linear model. More precisely, we first seek to establish the Cramer-Rao lower bound for the NAP VULA.

4.1.1 The Cramer-Rao Lower Bound for NAP

For the DOA estimation with the linear model, Stoica derived the CRLB in his early influential work of [39]. In this work, Stoica identified two paths of analysis: the conditional model, in which the sources are deterministic (their amplitudes are known a priori), and the stochastic model, in which the sources are treated as random processes. Naturally, these two perspectives lead to two different expressions for the CRLB. Here, we will focus on the stochastic version which holds under the Gaussian and stationary assumptions 1–3. The FIM for this case $\mathbf{J}_{\text{linear}}(\alpha)$ can be shown to be (interested readers may consult the derivation presented in Appendix C)

$$[\mathbf{J}_{\text{linear}}]_{ij}(\alpha) = K\text{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\alpha_j}\right]. \tag{50}$$

We seek to compare this expression with a derivation of the CRLB for the stochastic DOA estimation problem assuming measurements via the non-redundant virtual model. Our first task is to identify the underlying PDF of the virtual data (with the same assumptions we have made for the linear model). More specifically, each snapshot $\tilde{\mathbf{z}}$ constitutes an observation of the random vector $\tilde{z}$ that is drawn from a PDF that needs to be determined. To simplify this determination, we want to relate the random variable $\tilde{z}$ back to the random variable $\boldsymbol{S}_X$ for which we can derive its distribution more easily. The first step in doing so is realizing that the distribution of $\boldsymbol{S}_X$ coincides with that of the redundant random variable z. First, we note the following result, which follows from Proposition 1:

Proposition 4. *Let $\boldsymbol{S}_x$ be the random matrix representing the sample covariance matrix defined in Eq.* (15) *with underlying PDF $f_{\boldsymbol{S}_x}(\cdot)$. Then each measurement $\mathbf{z}$ in the nonlinear model Eq.* (47) *constitutes an observation of the random vector z whose PDF is given as*

$$f_z(\mathbf{z}) = f_{\boldsymbol{S}_x}(MAT\,[\mathbf{z}]).$$

Relating the PDF of $\boldsymbol{S}_x$ for the non-redundant random variable $\tilde{z}$ is a little bit trickier, however, since the Jacobian matrix of the transformation $\tilde{\mathbf{z}} = \mathbf{Pz}$ is nonsquare and hence does not have a determinant. As established in Proposition 3, however, we know that we can establish a one-to-one mapping between all vectors $\mathbf{z} \in \mathcal{X}$ and the space of non-redundant coarray measurements $\tilde{\mathbf{z}} \in \mathcal{Y}$ by means of the operators $\mathbf{P}$ and $\mathbf{P}^{\#}$. By applying the chain rule of conditional probability, we can see that the underlying probability distribution $f_z(\mathbf{z})$ is completely characterized by the non-redundant elements $\tilde{\mathbf{z}} = \mathbf{Pz}$. Consider the following simple example:

Example 3. *Consider the vector $\mathbf{z} = [z_1, z_2, z_3]^T$, where $z_3 = z_1$ for every $z_1 \in \mathbb{C}$. Using the chain rule of conditional probability, we see that*

$$\begin{aligned} f_z(z_1, z_2, z_3) &= f_z(z_1, z_2, z_1) \\ &= f_z(z_1 \mid z_2, z_1) f_z(z_2 \mid z_1) f_z(z_1). \end{aligned}$$

By definition, however, $f_z(z_1 \mid z_2, z_1) = 1$, since z_1 has already occurred. Thus, we see that

$$\begin{aligned} &f_z(z_1 \mid z_2, z_1) f_z(z_2 \mid z_1) f_z(z_1) \\ &= f_z(z_2 \mid z_1) f_z(z_1) = f_z(z_1, z_2). \end{aligned}$$

Therefore, we conclude that

$$f_z(z_1, z_2, z_3) = f_z(z_1, z_2).$$

In this manner, we can generalize this argument for any vector $\mathbf{z} \in \mathcal{X}$ using a straightforward calculation of the chain rule to make the claim that the PDF $f_z(\mathbf{z})$ over the sample space $\mathcal{X}$ is only dependent on the non-redundant portion of the vector. Therefore, we can conclude the following:

Proposition 5. *Let $f_{\tilde{z}}(\mathbf{z})$ denote the PDF of the random variable $\tilde{z}$. There exists a nonsquare permutation operator $\mathbf{P}^{\#} : \mathbb{C}^{\tilde{N}} \to \mathcal{X}$ such that for every $\tilde{\mathbf{z}} \in \mathbb{C}^{\tilde{N}}$, we have*

$$f_{\tilde{z}}(\tilde{\mathbf{z}}) = f_z(\mathbf{P}^{\#}\tilde{\mathbf{z}}).$$

The implication of Proposition 5 and Proposition 4 is that we only need to determine an appropriate PDF of the sample covariance matrix, since we can always map back and forth between the two perspectives with no distortion of the Euclidean space between the non-redundant variables. Therefore, we proceed with specifying the PDF of the sample covariance matrix. As before, we assume uncorrelated Gaussian sources and assume further that the number of snapshots K is fixed with $K > N$. We define the random matrix

$$\mathbf{W} = \sum_{k=1}^{K} \mathbf{x}_k \mathbf{x}_k^H . \tag{51}$$

Since the source amplitudes are assumed to be Gaussian, we see from Eq. (51) that the elements of $\boldsymbol{W}$ are a sum of squared Gaussian random variables. It can be shown that $\boldsymbol{W} \sim \mathcal{CW}\left(\mathbf{R}_x, K\right)$, which is a complex Wishart distribution with K degrees of freedom [6]. Again, since $K > N$, the $\boldsymbol{W}$ is a positive-definite random matrix for which its PDF takes the explicit form

$$f_{\boldsymbol{W}}\left(\mathbf{W} \mid \mathbf{R}_x, K\right) = \frac{|\mathbf{W}|^{K-N}}{I(\mathbf{R}_x)} \exp\left(-\mathrm{tr}\left[\mathbf{R}_x^{-1}\mathbf{W}\right]\right), \tag{52}$$

where $I(\cdot)$ is defined as

$$I(\mathbf{R}_x) = \pi^{N(N-1)/2}\, |\mathbf{R}_x|^K \prod_{n=1}^{N} \Gamma(K - n + 1).$$

The random matrix $\boldsymbol{S}_x$ is a linear transformation of $\boldsymbol{W}$ with a Jacobian of $1/K$; hence it is distributed as

$$\boldsymbol{S}_x \sim \mathcal{CW}\left(\frac{1}{K}\mathbf{R}_x, K\right). \tag{53}$$

With the PDF of $\boldsymbol{S}_X$ established, we now can compute its log-likelihood function:

$$L_{\mathrm{nonlinear}}(\boldsymbol{\alpha} \mid \mathbf{S}_x) = (K - N)\log|\mathbf{S}_x| - \log I\left(\frac{1}{K}\mathbf{R}_x\right) - K\mathrm{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right]. \tag{54}$$

The leftmost term in Eq. (54) vanishes as far as derivatives with respect to any of the parameters are concerned, hence we can reduce Eq. (54) to the simpler expression

$$L_{\mathrm{nonlinear}}(\boldsymbol{\alpha} \mid \mathbf{S}_x) = -\log I\left(\frac{1}{K}\mathbf{R}_x\right) - K\mathrm{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right]. \tag{55}$$

Now, proceeding to compute the gradient of the nonlinear likelihood function, we observe below that we have two terms to consider:

$$\frac{\partial L_{\mathrm{nonlinear}}}{\partial \alpha_i} = -\frac{\partial \log I\left(\frac{1}{K}\mathbf{R}_x\right)}{\partial \alpha_i} - K\frac{\partial \mathrm{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right]}{\partial \alpha_i}. \tag{56}$$

The second term on the right-hand side is a familiar quantity seen in the linear model and can be computed in a straightforward manner:

$$-K\frac{\partial \text{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right]}{\partial\alpha_i} = K\text{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1}\frac{\partial\mathbf{R}_x}{\partial\alpha_i}\right]. \tag{57}$$

The first term, however, needs a more intricate treatment; first note that

$$I\left(\mathbf{R}_x/K\right) = \left|\mathbf{R}_x/K\right|^K \pi^{N(N-1)/2}\prod_{n=1}^{N}\Gamma(K-n+1)$$

and using the determinant property $|c\mathbf{A}| = c^N\,|\mathbf{A}|$, we can simplify this to

$$\begin{aligned} &K^{-NK}\,|\mathbf{R}_x|^K\,\pi^{N(N-1)/2}\prod_{n=1}^{N}\Gamma(K-n+1)\\ &= K^{-NK}I(\mathbf{R}_x). \end{aligned} \tag{58}$$

Now, carrying out the gradient with respect to the first term on the right-hand side of Eq. (56),

$$\begin{aligned} &\frac{\partial\left[-\log I\left(\mathbf{R}_x/K\right)\right]}{\partial\alpha_i} = -\frac{1}{I\left(\mathbf{R}_x/K\right)}\frac{\partial I\left(\mathbf{R}_x/K\right)}{\partial\alpha_i}\\ &= -\frac{1}{I\left(\mathbf{R}_x\right)}\frac{\partial I\left(\mathbf{R}_x\right)}{\partial\alpha_i}\\ &= -\frac{K\left|\mathbf{R}_x\right|^{K-1}\pi^{N(N-1)/2}\prod_{n=1}^{N}\Gamma(K-n+1)}{I\left(\mathbf{R}_x\right)}\frac{\partial\left|\mathbf{R}_x\right|}{\alpha_i}\\ &= -\frac{K}{\left|\mathbf{R}_x\right|}\frac{\partial\left|\mathbf{R}_x\right|}{\partial\alpha_i}\\ &= -K\text{tr}\left[\mathbf{R}_x^{-1}\frac{\partial\mathbf{R}_x}{\partial\alpha_i}\right], \end{aligned} \tag{59}$$

which we see is also a familiar quantity from the likelihood function for the linear model. In fact, when putting Eqs. (59) and (57) back into Eq. (56), we see that the gradients for the nonlinear and linear models agree,

$$\frac{\partial L_{\text{nonlinear}}}{\partial\alpha_i} = \frac{\partial L_{\text{linear}}}{\partial\alpha_i}.$$

It then follows that the respective Hessians of the two models also agree, thus they yield the same FIM and ultimately the same CRLB. This is not a surprising conclusion; the nonlinear model is derived from the physical model without the introduction of any new information, and as such, *we cannot expect to attain additional DOF simply by changing our perspective of the estimation problem.*

The analysis above implies that we can establish an upper bound on the number of sources that can be estimated by reverting back to the FIM for the conventional linear model established in Eq. (50). In other words, we need to establish the conditions for which this FIM becomes singular as a function of the number of sources impinging on the array. Unfortunately, it is difficult to get perspective on this directly from Eq. (50); however, the work of Stocia [40] revealed a factorization of $\mathbf{J}(\boldsymbol{\alpha})$ through which the conditions for when it becomes singular can be seen readily. In fact, this work was furthered by Liu and Vaidyanathan [41], who established the connection of FIM singularity to the difference coarray. We proceed to summarize these findings. Using the fact that $\mathrm{tr}\,[\mathbf{ABCD}] = \mathrm{VEC}\left(\mathbf{B}^H\right)^H \left(\mathbf{A}^T \otimes \mathbf{C}\right) \mathrm{VEC}\left(\mathbf{D}\right)$ (see Proposition 8, Appendix A), and $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$, for nonsingular $\mathbf{A}$ and $\mathbf{B}$, we can rewrite the FIM as

$$\mathbf{J}(\boldsymbol{\alpha}) = K\mathrm{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\alpha_j}\right] \tag{60}$$

$$= K\left[\left(\mathbf{R}_x^T \otimes \mathbf{R}_x\right)^{-\frac{1}{2}}\frac{\partial \mathbf{z}}{\partial \alpha_i}\right]\left[\left(\mathbf{R}_x^T \otimes \mathbf{R}_x\right)^{-\frac{1}{2}}\frac{\partial \mathbf{z}}{\partial \alpha_j}\right]. \tag{61}$$

From here, we can factorize $\mathbf{J}(\boldsymbol{\alpha})$ so that the partials with respect to the angular and power parameters are separated out:

$$\mathbf{J}(\boldsymbol{\alpha}) = K\begin{bmatrix}\mathbf{G}^H \\ \mathbf{H}^H\end{bmatrix}\begin{bmatrix}\mathbf{G} & \mathbf{H}\end{bmatrix},$$

where

$$\mathbf{G} = \left(\mathbf{R}_x^T \otimes \mathbf{R}_x\right)^{-\frac{1}{2}}\left[\frac{\partial \mathbf{z}}{\partial \theta_1}, \dots, \frac{\partial \mathbf{z}}{\partial \theta_M}\right]$$

and

$$\mathbf{H} = \left(\mathbf{R}_x^T \otimes \mathbf{R}_x\right)^{-\frac{1}{2}}\left[\frac{\partial \mathbf{z}}{\partial \sigma_0^2}, \dots, \frac{\partial \mathbf{z}}{\partial \sigma_M^2}\right].$$

Expanding the expression, we have the block matrix

$$J(\boldsymbol{\alpha}) = K\begin{bmatrix}\mathbf{G}^H\mathbf{G} & \mathbf{G}^H\mathbf{H} \\ \mathbf{H}^H\mathbf{G} & \mathbf{H}^H\mathbf{H}\end{bmatrix}, \tag{62}$$

which clearly shows that $\mathbf{J}(\boldsymbol{\alpha})$ is positive semi-definite. We can conclude from Proposition 7 in the appendix that $\mathbf{J}(\boldsymbol{\alpha})$ is nonsingular if and only if $K\mathbf{H}^H\mathbf{H}$ and the Schur compliment of $\mathbf{J}$ which is $\mathbf{G}^H\mathbf{P}_{\mathbf{H}}^{\perp}\mathbf{G}$, are both nonsingular. The results can be established directly in terms of the number of sources (see [41] for detailed proofs):

Lemma 1. *The matrix* $\mathbf{H}^H\mathbf{H}$ *is positive definite if and only if* $\mathbf{A}_v$ *has full column rank, i.e.,*

$$\mathit{rank}(\mathbf{A}_v) = M + 1.$$

To establish the second condition, i.e., that $\mathbf{G}^H\mathbf{P}_{\mathbf{H}}^{\perp}\mathbf{G}$ is positive definite, it helps to define the augmented coarray manifold (ACM).

Definition 4. *The ACM is the $\tilde{N} \times (M+1)$ matrix defined as*

$$\tilde{\mathbf{A}}_c = \left[diag \left[D_c \right], \mathbf{A}_c \right].$$

Lemma 2. *The matrix $\mathbf{G}^H \mathbf{P}_{\mathbf{H}}^{\perp} \mathbf{G}$ is positive definite if and only if*

$$rank(\tilde{\mathbf{A}}_c) = 2M + 1.$$

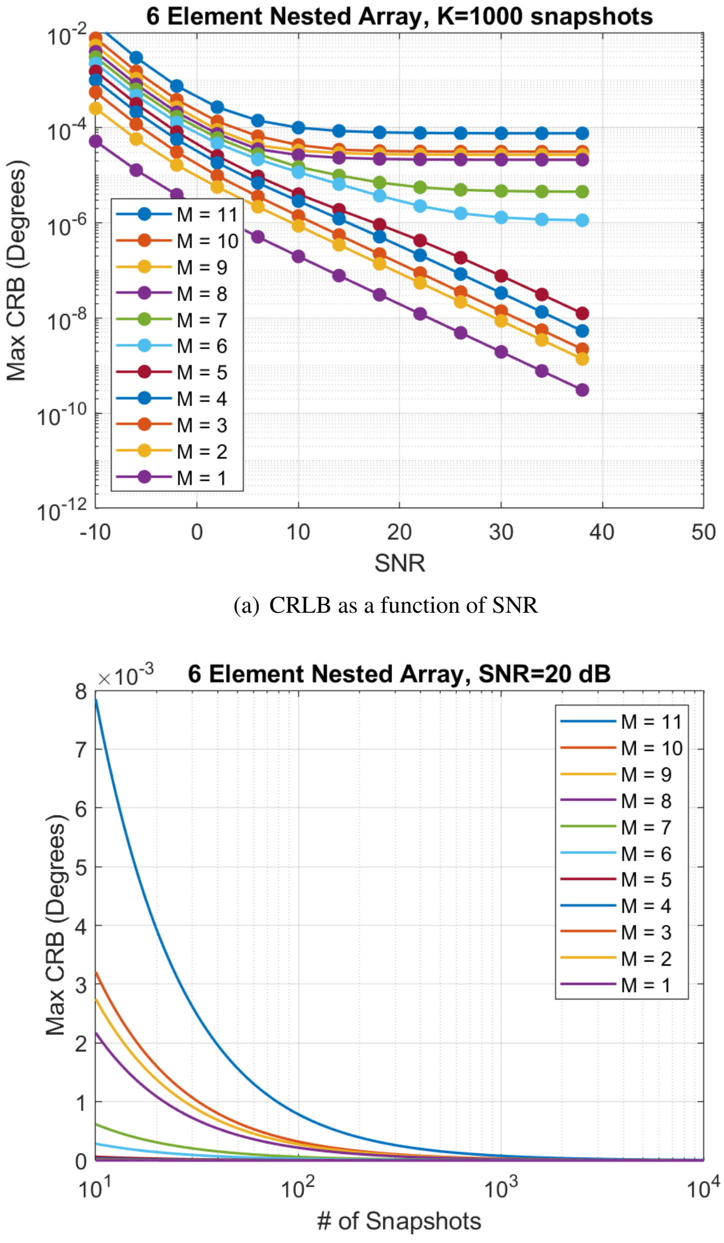
These two lemmas provide the basis to conclude the main result:

Theorem 1. *If $M > (\tilde{N} - 1)/2$, then for any choice of M DOA angles, we have that the FIM $\mathbf{J}(\alpha)$ is singular.*

Theorem 1 answers one important question but at the same time leaves the door open to another fundamental question. We can safely conclude that NAP does not yield any more DOF than conventional augmentation can, regardless of which DOA estimator is applied with respect to the non-linear model. However, nothing that we have discussed yields any limitation to the resolution question due to the seemingly longer aperture of the nonlinear model, which motivates an exploration of alternative DOA estimation approaches other than spatial smoothing that can exploit this.

4.1.2 Dependence of the Virtual Covariance on the Number of Snapshots

As we discussed in the introduction of this report, the issue of finite time sample support presents a challenge for estimating more sources than the number of array elements N. Even though we have shown that for certain sparse arrays NAP allows for more DOF than elements, the amount of time sample support available plays a crucial for the ability to exploit this. To see this, we can appeal to the CRLB characterization presented above, as again, the CRLBs for the linear and nonlinear models coincide. As a simple example, we consider a 6-element nested array in which a varying number of plane wave sources are incident on the array with fixed DOAs. In this case the number of virtual elements is $\tilde{N} = 23$ which implies that the maximum number of sources that can be identified with the array is 11. Figure 9 shows a plot of the maximum CRLB as a function of SNR for a fixed number of snapshots as well as a plot of the maximum CRLB as a function of the number of snapshots for a fixed SNR level. In the SNR sweep plot, Fig. 9(a), we can observe two distinct regions: the overdetermined regime ($M < N$) and the underdetermined regime ($M \geq N$). In the underdetermined regime, we see that simply increasing SNR does very little to lower the stagnating error bound, which is in stark contrast to the overdetermined case, in which the maximum CRLB bound readily decreases with increasing SNR. However, in Fig. 9(b), we observe that in the underdetermined region, the estimation error converges to zero as we take the number of snapshots to infinity. This simple example shows the importance of snapshots for the underdetermined DOA estimation problem, which is independent of whether NAP is used for DOA estimation. This simple example demonstrates that we cannot expect to treat the nonlinear model completely as a physical ULA where the traditional RMB rule of thumb is to collect approximately twice the number of elements, worth of snapshots.



(a) CRLB as a function of SNR

(b) CRLB as a function of snapshots

Fig. 9—Plots of the CRLB for a 6-element nested array for a varying number of sources

We can get a more concrete sense of the role of finite sample support by explicitly characterizing the covariance matrix, $\mathbf{R}_z$, that corresponds to the noisy virtual model in Eq. (49). Under the assumption of Gaussian sources, a straightforward calculation shows that

$$\begin{aligned}\mathbf{R}_z &= E\left[\tilde{\mathbf{z}}\tilde{\mathbf{z}}^H\right] \\ &= \mathbf{A}_v\mathbf{p}\mathbf{p}^H\mathbf{A}_v^H + E\left[\tilde{\mathbf{v}}\mathbf{p}^H\mathbf{A}_v^H\right] + E\left[\mathbf{A}_v\mathbf{p}\tilde{\mathbf{v}}^H\right] + E\left[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^H\right] \\ &= \mathbf{A}_v\mathbf{p}\mathbf{p}^H\mathbf{A}_v^H + E\left[\tilde{\mathbf{v}}\right]\mathbf{p}^H\mathbf{A}_v^H + \mathbf{A}_v\mathbf{p}E\left[\tilde{\mathbf{v}}\right]^H + E\left[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^H\right],\end{aligned} \tag{63}$$

where we make note that the underlying (power) sources are no longer random in this view, thus, the expectation operator is taken only with respect to the virtual noise vector $\tilde{\boldsymbol{\nu}}$. Under our previously stated assumptions, here the virtual noise vector $\mathbf{R}_\nu$ is solely attributable to the fact that we have finite sample data. It is of interest to characterize the first two moments of the virtual noise $E\left[\tilde{\boldsymbol{\nu}}\right]$ and $\mathbf{R}_\nu = E\left[\tilde{\boldsymbol{\nu}}\tilde{\boldsymbol{\nu}}^H\right]$. Similar to our CRLB analysis, we again note that it is equivalent to work directly with the redundant matrix form $\tilde{\boldsymbol{\nu}} = \mathbf{P}\mathrm{VEC}\left(\mathbf{N}\right)$. This means we have a mapping for the virtual spatial lags to the matrix indices of the covariance data as $i' \mapsto (i, j)$ and $j' \mapsto (k, l)$. From this mapping we can compute the covariance terms explicitly:

$$E\left[\left[\mathbf{N}_x\right]_{ij}\left[\mathbf{N}_x\right]^*_{kl}\right] = E\left[\left[\mathbf{S}_x\right]_{ij}\left[\mathbf{S}_x\right]^*_{kl}\right] - E\left[\left[\mathbf{S}_x\right]_{ij}\right]E\left[\left[\mathbf{S}_x\right]^*_{kl}\right].$$

Based on our prior assumptions, the sample covariance matrix obeys a complex Wishart distribution; straightforward calculations lead to the obvious conclusion:

$$E\left[\mathbf{S}_x\right] = \mathbf{R}_x,$$

which means that

$$E\left[\mathbf{N}_x\right] = \mathbf{0},$$

from which, by using the mapping between the covariance data and the virtual array, we conclude that $E\left[\tilde{\boldsymbol{\nu}}\right] = \mathbf{0}$. Using this result in Eq. (63), we see that

$$\mathbf{R}_z = \mathbf{A}_\nu\mathbf{p}\mathbf{p}^H\mathbf{A}_\nu^H + \mathbf{R}_\nu. \tag{64}$$

The straightforward, albeit tedious, calculations presented in [42] lead to an expression for the second moment of $\mathbf{S}_x$:

$$E\left[\left[\mathbf{S}_x\right]_{ij}\left[\mathbf{S}_x\right]^*_{kl}\right] = \left[\mathbf{R}_x\right]_{ij}\left[\mathbf{R}_x\right]_{kl} + \frac{1}{K}\left[\mathbf{R}_x\right]_{lj}\left[\mathbf{R}_x\right]_{ik}, \text{ for } i, j, k, l = 1 \ldots N, \tag{65}$$

implying that the covariance noise term can be expressed as

$$E\left[\left[\mathbf{N}_x\right]_{ij}\left[\mathbf{N}_x\right]^*_{kl}\right] = \frac{1}{K}\left[\mathbf{R}_x\right]_{lj}\left[\mathbf{R}_x\right]_{ik}, \text{ for } i, j, k, l = 1 \ldots N. \tag{66}$$

Now, again using the mapping $(i', j') \mapsto ((i, j), (k, l))$, we compute the diagonal and off-diagonal terms as

$$E\left[\left[\tilde{\boldsymbol{\nu}}\right]_{i'}\left[\tilde{\boldsymbol{\nu}}\right]^*_{j'}\right] = \frac{1}{K}\begin{cases}\left(\sum_{m=0}^{M}\sigma_m^2\right)^2, \text{for } i' = j' \\ \\ \sum_{m=1}^{M}\sum_{m'=1}^{M}\sigma_m^2\sigma_{m'}^2 e^{i\frac{2\pi}{\lambda}\left(d_{lj}\sin(\theta_m) + d_{ik}\sin(\theta_{m'})\right)}, \ i' \neq j',\end{cases} \tag{67}$$

where $d_{ik}, d_{jl} \in \mathbb{D}_l$ are the appropriate coarray elements.

Looking at the structure of $\mathbf{R}_z$ and $\mathbf{R}_\nu$ in Eqs. (64) and (67), we can make a few observations: First, as expected, we see that $\mathbf{R}_\nu$ converges to zero as the number of snapshots $K \to \infty$ at a rate of $\mathcal{O}\left(\frac{1}{K}\right)$.

Second, we see that diagonal terms are a direct sum of $\binom{M+2}{2}$ terms (self terms + cross terms), which grows as $\mathcal{O}\left(M^2\right)$. Third, unlike the case of the physical array, the noise covariance matrix is not white and actually depends explicitly on the DOAs of the sources. As a simple example, we can use the variance expression in Eq. (67) to calculate the virtual SNR for the case of $M = 11$ sources impinging on a 6-element nested array all with unity power and where we use a per-element SNR of 20 dB to measure each snapshot. In Fig. 10, we show a plot of the average virtual SNR as seen by each virtual channel in the coarray. We note that in this example, we would require almost 1,000 snapshots to achieve a virtual SNR equivalent to that of the physical channel's 20 dB SNR.

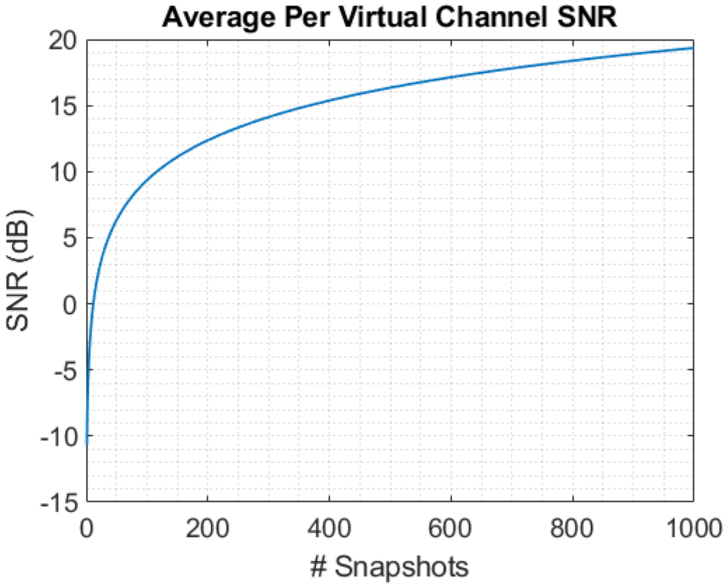


Fig. 10—Calculation of the average virtual channel SNR as a function of snapshots for the case of 11 sources measured at a 20 dB per-physical-channel SNR

4.2 Spatial Processing Algorithms for NAP

We now consider adopting the spatial-processing techniques that we have covered in Chapter 2 for the linear model toward the NAP virtual model. This includes numerical algorithms for DOA estimation as well as the spatial filter problem. For all numerical simulations we used a 6-element nested array, which corresponds to a 23-element virtual ULA.

4.2.1 DOA Estimation

DOA estimation using the NAP model can be applied using any of the single-snapshot-capable algorithms we previously discussed in Chapter 2. We briefly run through these and point out differences or parameters associated with each approach that will need to be adapted for the NAP model. Although a Fourier approach,

$$\hat{\mathbf{p}}(\boldsymbol{\theta}) = \mathbf{A}_c^H \tilde{\mathbf{z}},$$

is the most straightforward, as it requires almost no modification from the physical array version, resolution and sidelobe concerns are still an issue. Another SMV method we discussed in Chapter 2 is BPDN, which again is a superresolution approach that is free of sidelobes. A useful modification of BPDN for use with NAP is to add a positivity constraint, as this imposes additional regularization, which explicitly enforces the fact that source powers are real and positive:

$$\min_{\mathbf{p}\in\mathbb{R}^M} \|\mathbf{p}\|_1 \text{ subject to:} \tag{68}$$

$$\|\mathbf{A}_c\mathbf{p} - \tilde{\mathbf{z}}\|_2 < \epsilon_\nu \tag{69}$$

$$p_i \geq 0, \text{ for i=1,...,M.} \tag{70}$$

Here, the parameter $\epsilon_\nu = \sqrt{E\left[\|\boldsymbol{\nu}\|^2\right]}$, which, with the aid of Eq. (67), we can estimate as

$$\epsilon_\nu = \sqrt{\frac{\tilde{N}}{K}\sum_{m=0}^{M}\sigma_m^2}.$$

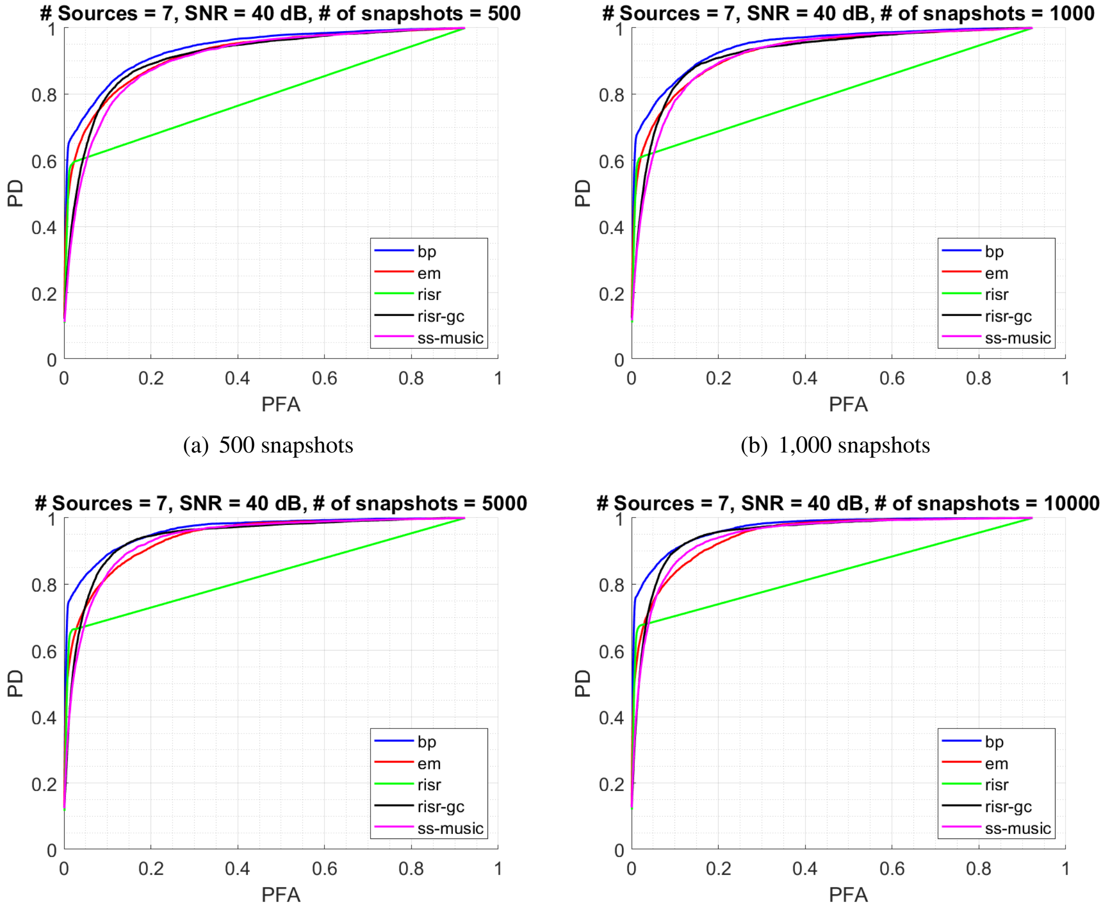
The iterative correlation-based approaches RISR, RISR-GC, and EM are all single-snapshot-capable methods and could also be applied to the NAP virtual array. Again, all of these methods assume a structured covariance model for which our characterization of $\mathbf{R}_z$ in Eq. (63) could be used in place. Examination of Eq. (63), however, reveals that both the virtual source and the virtual noise covariance matrices are nondiagonal. As we are only interested in estimating the diagonal terms of the virtual source covariance matrix, we seek to enforce the following virtual covariance structure:

$$\mathbf{R}_z = \mathbf{A}_c\boldsymbol{\Sigma}_c\mathbf{A}_c^H + \mathbf{R}_\nu,$$

with $\boldsymbol{\Sigma}_c = \operatorname{diag}\left[\sigma_0^4, \ldots, \sigma_M^4\right]$ and $\mathbf{R}_\nu$ is the virtual noise covariance matrix. Although we calculated both the diagonal and off-diagonal terms of the virtual noise covariance in Eq. (67), it is clear that the off-diagonal terms depend on knowledge of the source DOAs, which we do not know a priori. Therefore, for DOA estimation, we only consider the variance terms, which, as shown, are simply the total power received at each element squared and would be known prior to any DOA estimation.

Numerical simulations were conducted to quantitatively evaluate the different DOA estimation algorithms with NAP. Although the mean-square error (MSE) between the true angles and the estimated ones is a common metric to evaluate DOA estimation performance, we argue that it is not adequate to capture the behavior of false or missed sources that is prevalent with underdetermined DOA estimation. This is because calculating the MSE relies on a peak detector as well as an alignment of the estimated angles with the ground truth grid. Furthermore, it also assumes that the spectrum of an estimation algorithm will always yield M peaks. What we observed in our numerical experiments, however, is that missed detections, false peaks, and angular drift often can occur with of these methods being applied to the virtual array. As such, we will characterize the performance of the various DOA estimation algorithms through their respective receiver operating curves (ROCs), which is a plot of the probability of detection P_d against the probability of false alarm P_{fa} for various peak thresholds. The ROCs were generated from different sets of Monte Carlo simulations using the discussed estimation algorithms under different simulation parameter permutations, including SNR, the number of sources, snapshots, and power characteristics. During each trial run, the source DOAs were chosen randomly according to a uniform distribution along the entire FOV using a grid with 1° spacings. For the off-grid trials, a perturbation factor was added to the true DOAs according to a normal distribution with half a degree of variance. The power levels of the respective sources also were chosen randomly according to a Swerling 1 power fluctuation model; i.e., the power levels were drawn from a Chi-square distribution with two DOF and with no fluctuation from snapshot to snapshot. After estimating the spectrum using the various NAP algorithms discussed, we thresholded the resulting spectrums across their respective dynamic ranges using linearly increasing thresholds to compute detection and false-alarm characteristics. To account for angular drift in the detection process, we declared any spectral energy meeting the threshold characteristic that was within a 1° tolerance of the true DOA a true detection.

The first set of experiments considered only on-grid sources. Figure 11 shows some results corresponding to the case of seven plane wave sources measured using a 40 dB per element SNR. From here, we clearly see that BPDN as well as RISR-GC exhibit the best performance characteristics exhibiting the best detection performance over all false alarm probabilities. This was especially the case within the lower-snapshot regimes. The ROC plot for RISR shows rather poor performance of the method, which was expected, as the technique seemed to be highly prone to generating false peaks. Spatial smoothing appeared to be on par



(a) 500 snapshots

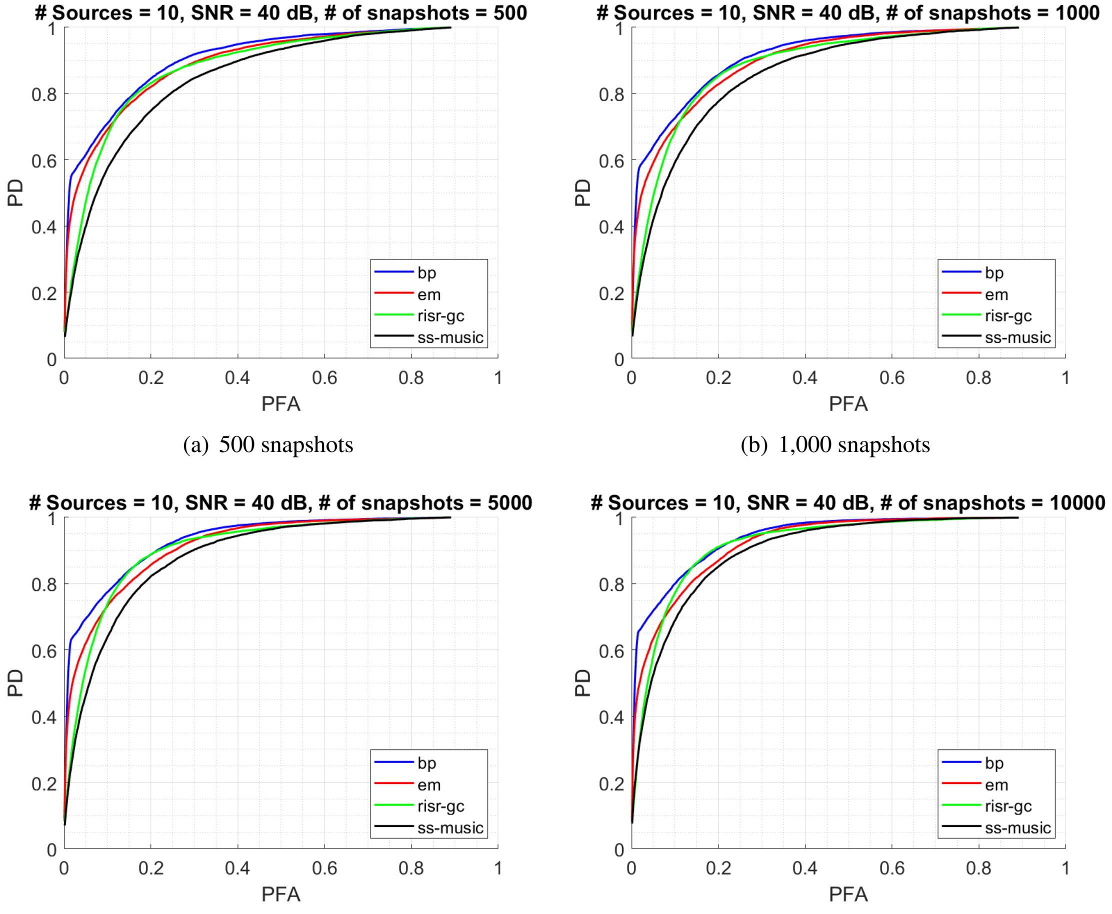
(b) 1,000 snapshots

(c) 5,000 snapshots

(d) 10,000 snapshots

Fig. 11—ROC curves for the case of 7 sources and 40 dB SNR

with EM when the number of sources is low; however, spatial smoothing demonstrates a clear increase in performance as the number of snapshots increases. At least in this case, spatial smoothing exhibited the most sensitivity to time sample support. Figure 12 shows the results of another set of similar experiments in which ten plane wave sources were used in the estimation experiments. Note that we dropped RISR from these results because the appearance of false sources dramatically increased with an increase in the number of sources, which resulted in very poor performance. Here, again, BPDN demonstrates the most favorable performance, while RISR-GC and EM exhibited similar performance for lower numbers of snapshots. However, differences between RISR-GC and EM become more notable as the number of snapshots increases. Spatial smoothing has visibly worse characteristics here than in the previous trial.



(a) 500 snapshots

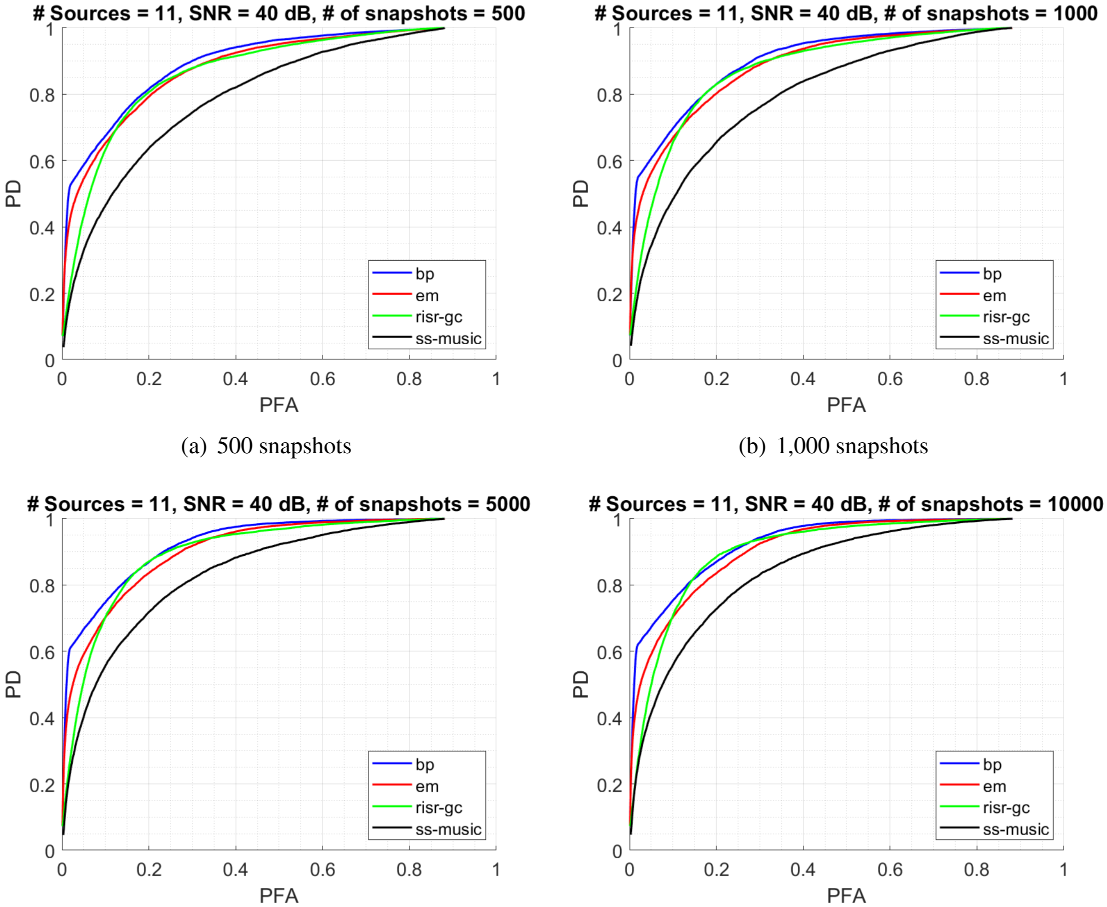
(b) 1,000 snapshots

(c) 5,000 snapshots

(d) 10,000 snapshots

Fig. 12—ROC curves for the case of 10 sources and 40 dB SNR

A similar story occurs in Fig 13, which is the case where the array is operating at the maximum limit in terms of the number of sources. Again, we see a considerable gap between the performance of spatial smoothing MUSIC and the rest of the other algorithms. Even as the number of snapshots becomes relatively large, spatial smoothing is unable to close the gap. Although we did conduct other experiments using lower SNR levels, we note that similar conclusions between the various algorithms can be drawn as presented here. Again, as shown in our analysis, in the underdetermined region, NAP performance is really dominated by



(a) 500 snapshots

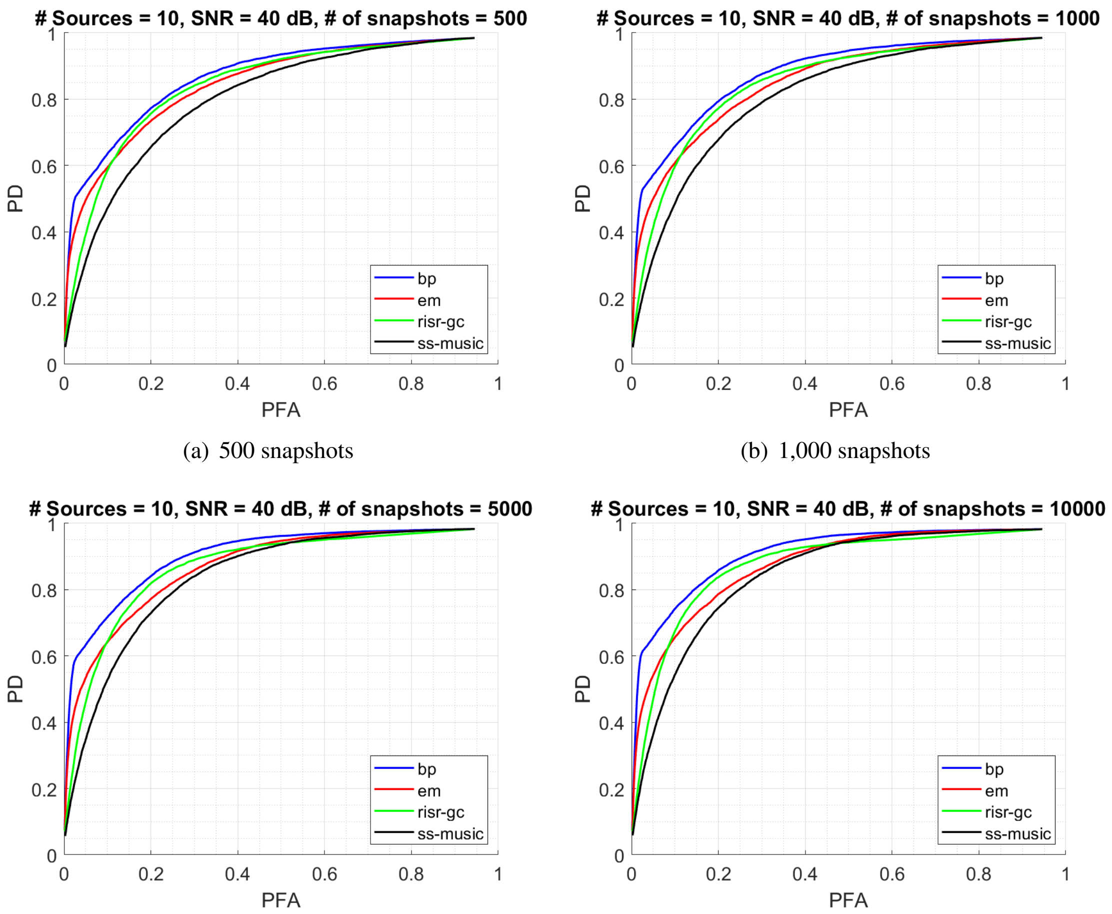
(b) 1,000 snapshots

(c) 5,000 snapshots

(d) 10,000 snapshots

Fig. 13—ROC curves for the case of 11 sources and 40 dB SNR

time sample support rather than SNR. Figure 14 presents Monte-Carlo simulations in which ten plane wave sources using off-grid DOAs were considered. Surprisingly, BPDN still exhibits the best performance and again, we see spatial smoothing had the least-favorable performance. Our next set of numerical experiments was focused on the question of resolution. As discussed in Chapter 2, RISR, RISR-GC, and EM can be seen as spectrum estimators that effectively yield an estimate of the data covariance $\mathbf{R}_x$ using a structured covariance model. Subsequently, MUSIC then can be applied with respect to this reconstruction. We can apply this same approach for each of these algorithms toward estimation of $\mathbf{R}_z$. In this manner, we can compare resolution performance of these spectrum estimators against spatial smoothing.



(a) 500 snapshots

(b) 1,000 snapshots

(c) 5,000 snapshots

(d) 10,000 snapshots

Fig. 14—ROC curves for the case of 10 off-grid sources whose response was measured at 40 dB SNR

Figure 15 shows a single simulation in which two sources of equal power separated by 1° are estimated using MUSIC through spectrums generated by spatial smoothing, RISR-GC, and EM. Although for all three methods we do observe some angular drift, it is clear that the spectrums of RISR-GC and EM yield sharper estimates of the two peaks than the spectrum of spatial smoothing. In an attempt to conduct a fair quantitative analysis, we consider computing the probability of resolution [43] using the MUSIC spectrums corresponding to each of the three methods. More specifically, let $p\left(\theta \middle| \hat{\mathbf{R}}_z\right)$ be the MUSIC spectrum yielded for a given estimate of the NAP covariance matrix. The corresponding null spectrum then is given as

$$n(\theta) = \frac{1}{p\left(\theta \middle| \hat{\mathbf{R}}_z\right)}.$$

Then, for any two sources with corresponding DOA angles θ_1 and θ_2, we have the metric

$$\gamma(\theta_1, \theta_2) = n(\theta_m) - \frac{1}{2}\left(n(\theta_1) + n(\theta_2)\right), \tag{71}$$

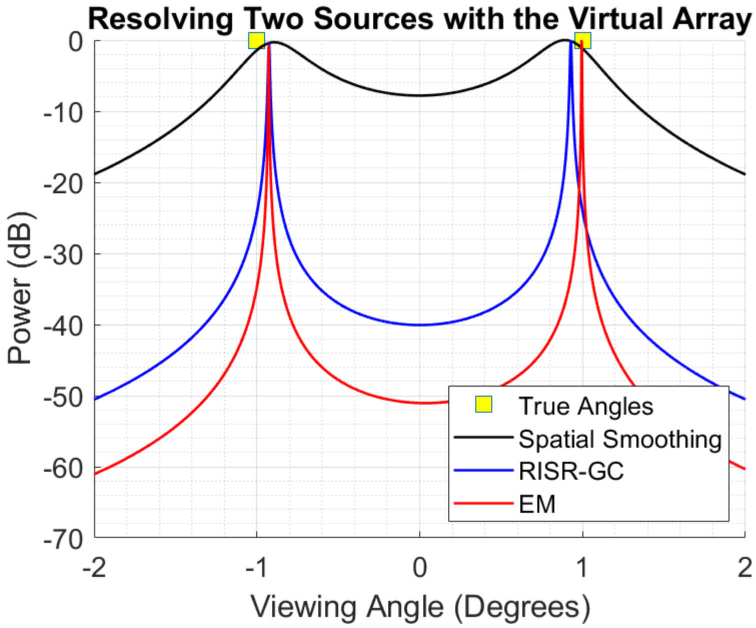


Fig. 15—Comparison of the MUSIC spectrum as estimated through spatial smoothing and iterative estimation techniques for two sources separated by 1°

where $\theta_m = (\theta_1 + \theta_2)/2$ is the midpoint angle between the two DOAs. We say that the two sources are resolved whenever $\gamma(\theta_1, \theta_2) > 0$. The probability of resolving the two sources at two fixed DOAs θ_1 and θ_2 is then

$$P_{\text{res}} = \Pr\left[\gamma > 0\right].$$

We attempted to estimate probability of resolution for all three estimation methods using Monte Carlo simulations for two closely spaced DOAs. To remove the effect of angular drift in our comparison, a peak detector was employed during each run to determine the angular positions of the two peaks. The center angle θ_m then is computed using the detected angles rather than the true center angle. If only one peak was obtained, then we would assign γ a value of zero. We used 2,000 snapshots to perform the estimations of $\mathbf{R}_z$ for all the runs. Figure 16 shows the results of the comparison between spatial smoothing, RISR-GC, and EM in terms of their respective probabilities of resolution. It is clear that both iterative methods exhibit better resolution performance than does spatial smoothing. Although these results could be attributed to the fact that both iterative methods make use of the full length of the coarray, a more rigorous study should be undertaken to confirm this.

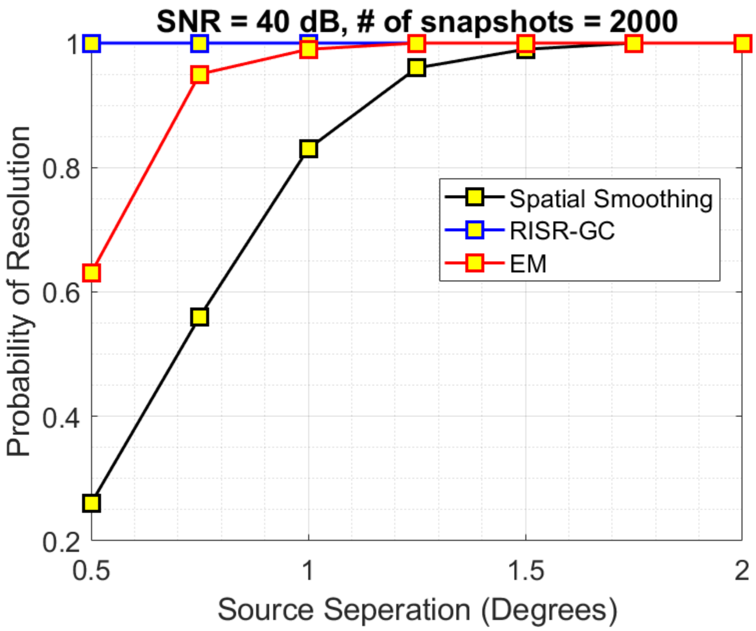


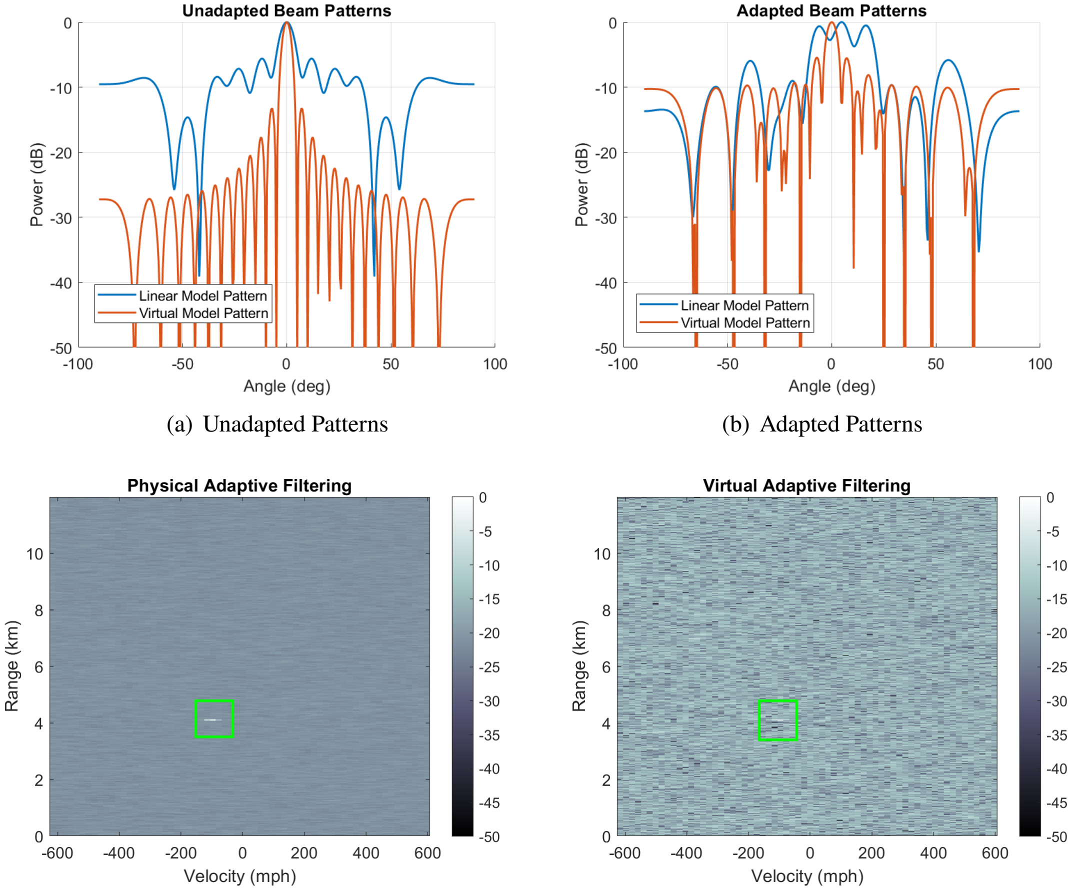
Fig. 16—An estimation of the probability of resolution comparing iterative spectrum estimators RISR-GC and EM to spatial smoothing

4.2.2 Adaptive Beamforming

At first glance, adaptive beamforming with the NAP virtual array appears straightforward, since beamforming weights such as Eqs. (10) and (12), or any other set of weights designed for a ULA with the same length as $\mathbf{z}$, could be applied directly. However, in reality, the issue of time sample support (both physical and virtual) and the fact that NAP relies on incoherent integration puts limits on its applicability toward radar. It is not hard to see that techniques like MVDR become problematic for NAP beamforming, since not only does a noise-plus-interference virtual covariance matrix $\mathbf{R}_z$ need to be estimated from a single virtual snapshot, but also, as we have shown in our discussion above, interference in the virtual array largely can result from signal cross-terms that have not decayed sufficiently. Thus, the expectation is that adaptive spatial filters like MVDR, which compute their nulls based on covariance data, will end up expending the increased DOF of the virtual array towards suppressing cross-term energy rather than the physical interference sources. On the other hand, we can employ beamformers where nulls are specified explicitly (such as Eq. (10)) to ensure mitigation of the interference directions, themselves, rather than the cross-terms. However, this comes at the risk of cross-term energy coming in through the sidelobes of the virtual array. So a natural question here is whether the increased DOF offered by NAP really does result in an overall increase in SINR, given that we should expect cross-term energy to be present for any quasi-reasonable radar dwell time.

We conducted some preliminary simulations to assess how such a clairvoyant virtual beamforming approach would work within the context of a typical monostatic radar system, and whether any gain in SINR over conventional linear beamforming could be observed. Since NAP relies on incoherent integration, snapshot data would need to be obtained only after all coherent integration stages are completed. One way to implement this is to consider array snapshots over multiple CPIs within a dwell. Our goal, then, is to implement NAP using this approach and then to compare the resulting SINR with what could be achieved using using linear array processing. In both cases, we will assume clairvoyant knowledge of the interference directions and design a beamformer using Eq. (11) for the linear array processing case and Eq. (10) for the NAP case. To ensure a fair comparison, for the same dwell, we first process each CPI along the range and Doppler dimensions to yield a data cube over the physical array channels, range, and Doppler dimensions. For linear processing, beamforming is applied within the same CPI block, which yields a single range-Doppler image for each CPI block. The final image for the linear processing case is then incoherently integrated across each CPI block within the dwell. For NAP, we take each channel-range-Doppler map and compute a sample covariance matrix for each range-Doppler cell with snapshots taken as samples across each CPI block. This results in a data cube over virtual channel, range, and Doppler. Virtual beamforming then is applied across the virtual channel dimension to obtain a single range-Doppler spectrum. Our simulations were conducted assuming a target of interest located at an approximate range of 4 kilometers, azimuth angle 0°, and a constant radial velocity of 100 mph. The target is interrogated using 64 pulses of a 10 MHz LFM chirp using a variable number of CPI blocks. We assume the target's parameters remain constant for the entire dwell. The received signal then is corrupted with a variable number of stationary Gaussian noise jammers M (for M greater than the number of array elements), all with equal power relative to the RCS of the target. Each physical channel also is corrupted with white Gaussian noise using a 30 dB SNR. Figure 17 shows one example of a case where eight jammers with DOA angles $\theta = [-65°, -47°, -32°, -15°, 25°, 35°, 48°, 68°]^T$ flood the 6-element physical array, each with a power strength of 50 dB relative to the RCS of the target.

In Fig. 17(b), we see the adapted beam patterns of the nested array and resulting virtual ULA. We observe that the nested array's beam pattern does a poor job at maintaining gain at the look direction while it tries to mitigate the flood of interference, while the virtual array's beam pattern clearly leverages the increased DOF to place deep nulls at the desired locations while maintaining gain in the desired look direction. In Fig. 17(c), however, the resulting range-Doppler map of the NAP approach shows a much higher noise floor than the resulting heat map of the linear approach shown in Fig. 17(b). We can attribute this result to the effect of limited time sample support where the cross terms of the various jammers induce secondary interference at various DOA angles. By increasing the radar dwell time (the number of CPI blocks) we can obtain more snapshots for virtual beamforming. Figure 18 plots the gain in SINR as a function of the number of CPI blocks used to estimate the virtual array data $\mathbf{z}$. As expected, we see that SINR gain does indeed increase as the number of CPI blocks increase; however, it is only when using approximately 512 CPI blocks that we finally achieve a positive gain. This simple example demonstrates the impracticality of NAP, as it seems an incredibly long time dwell time still only manages a small amount of SINR gain.



(a) Unadapted Patterns

(b) Adapted Patterns

(c) Linear Processing

(d) NAP

Fig. 17—Adapted patterns and resulting range-Doppler maps using 64 CPIS for the case of eight jammers at about 50 dB above the RCS of the target

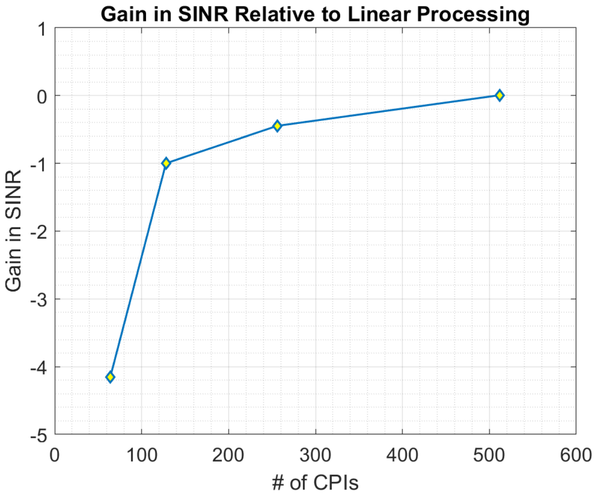


Fig. 18—Comparison between linear and nonlinear processing as a function of dwell time. We see that the relative gain in SINR for NAP becomes positive around 512 CPIs.

5. TESTBED AND EXPERIMENTAL SETUP

A significant component of this project involved constructing a testbed to allow the generation and capture of signals in a controlled environment as a means to augment the simulations and to provide some "real-world" grounding. For budgetary reasons, we repurposed as much existing hardware and test equipment as possible, with a mixture of custom designs and new test equipment making up the balance. The end result had certain specific limitations but was more than sufficient to generate the data needed here. In this chapter, we document the design, implementation, and operation of the testbed.

An overview of the main functional components of the testbed developed for the NAP project is shown in Fig. 19. The testbed consists of two primary functional units: the scene generator (left side of the figure) and the receive array (right). The former is used to create test waveforms with programmable spatial and temporal dependence, while the latter receives the waveforms and stores the data for later processing. These units are described in greater detail below, followed by an overview of the software used to integrate and control the testbed and procedures used for testing.

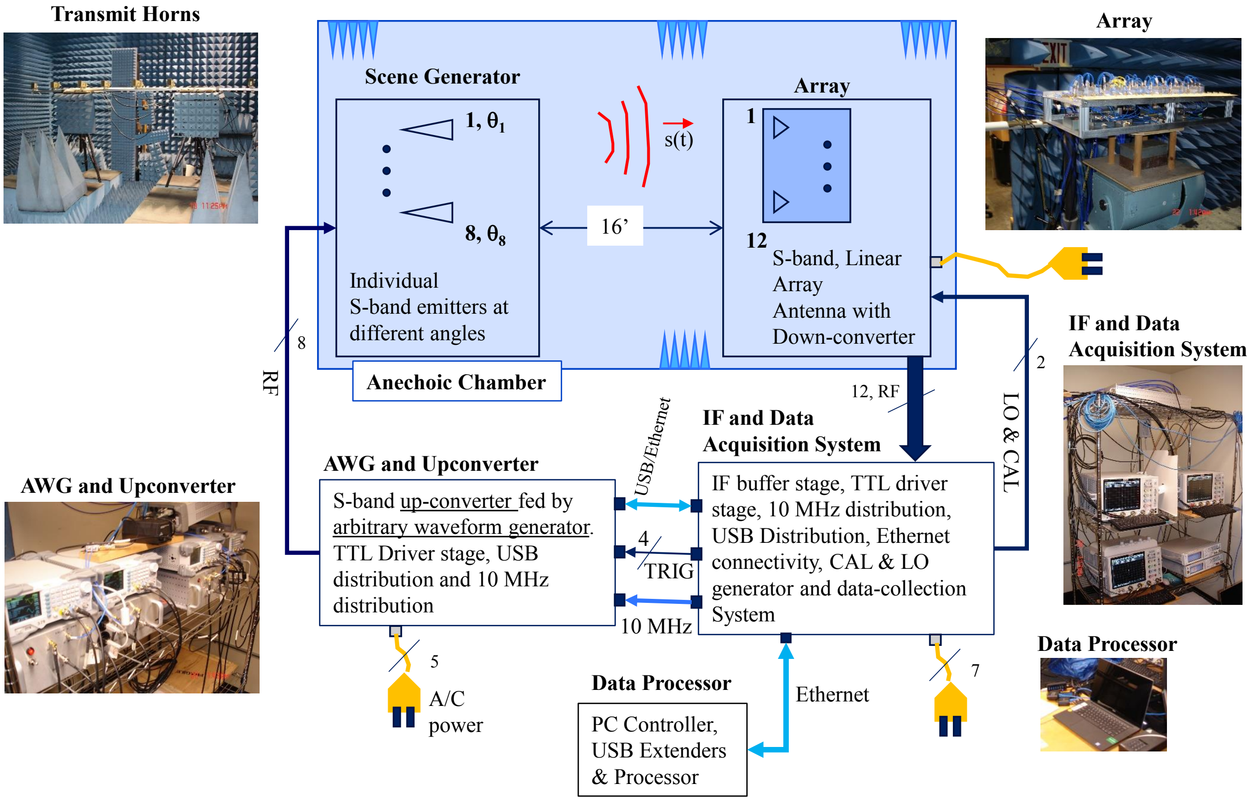

Transmit Horns
Array
Scene Generator
Array
1, θ1
8, θ8
s(t)
1
12
16'
Individual S-band emitters at different angles
S-band, Linear Array Antenna with Down-converter
Anechoic Chamber
8
RF
12, RF
2
LO & CAL
USB/Ethernet
IF and Data Acquisition System
IF and Data Acquisition System
AWG and Upconverter
AWG and Upconverter
S-band up-converter fed by arbitrary waveform generator. TTL Driver stage, USB distribution and 10 MHz distribution
IF buffer stage, TTL driver stage, 10 MHz distribution, USB Distribution, Ethernet connectivity, CAL & LO generator and data-collection System
4
TRIG
10 MHz
5
A/C power
7
Ethernet
Data Processor
Data Processor
PC Controller, USB Extenders & Processor

Fig. 19—Top-level overview of NAP experimental system

5.1 Scene Generator

The purpose of the scene generator, as the name implies, is to generate and transmit electromagnetic fields corresponding to a variety of scenarios. It can generate signals from up to eight independent sources, each of which can be either continuous white noise or a pulse train of identical pulses. The white noise sources are independent and uncorrelated with each other. The pulses can be programmed arbitrarily (within the hardware limitations) and thus can be made to be correlated or uncorrelated with each other as desired. In addition, the pulse trains can be delayed independently and frequency-modulated in order to simulate range and velocity of a radar target. With this setup, we thus can model radar returns, radar multipath, and both noise- and DRFM-based jammer returns, as well as other pulsed and CW-type interferences.

The scene generator consists of three primary subsystems, as shown from top to bottom in Fig. 20. The first is baseband waveform generation, which consists of four Rigol DG1032Z dual-channel arbitrary waveform generators (AWGs). The second is the custom upconverter, which translates the eight AWG output to our RF of 3.4 GHz while applying any Doppler offsets and amplifying the signals. Finally, the signals are transmitted from eight S-band horn antennas, which can be either tripod-mounted or mounted to an 80/20 rail for greater positioning precision. We describe each in turn.

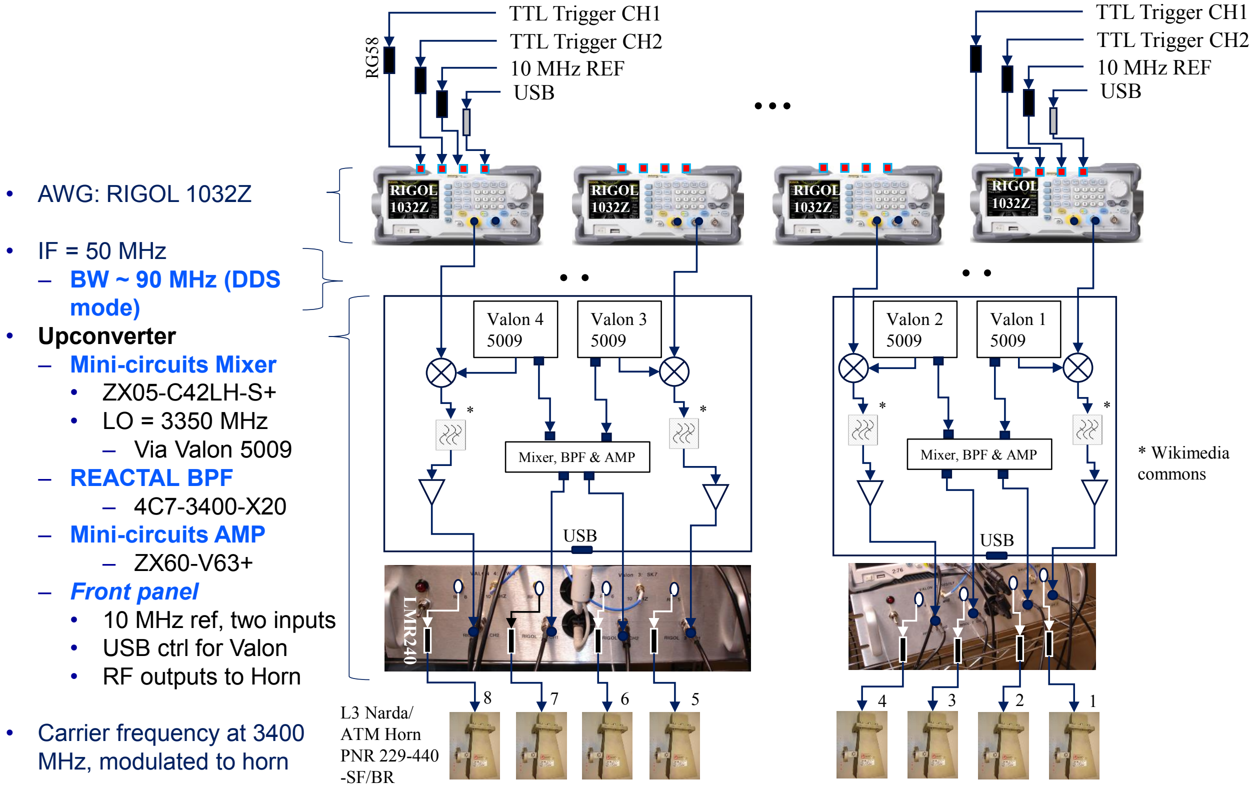

- AWG: RIGOL 1032Z
- IF = 50 MHz
  - **BW ~ 90 MHz (DDS mode)**
- **Upconverter**
  - **Mini-circuits Mixer**
    - ZX05-C42LH-S+
    - LO = 3350 MHz
      - Via Valon 5009
  - **REACTAL BPF**
    - 4C7-3400-X20
  - **Mini-circuits AMP**
    - ZX60-V63+
  - ***Front panel***
    - 10 MHz ref, two inputs
    - USB ctrl for Valon
    - RF outputs to Horn
- Carrier frequency at 3400 MHz, modulated to horn

Fig. 20—The major components of the scene generator include (from top) Rigol DG1032Z arbitrary waveform generators, eight-channel custom upconverter, and S-band horn antennas with mounts

5.1.1 *Rigol DG1032Z Arbitrary Waveform Generator*

The signals for the eight sources were generated using four Rigol DG1032Z dual-channel AWGs. Each channel can be operated independently, providing eight independent signals, and the Rigols can be controlled remotely via USB or ethernet to automate waveform loading and configuration. The Rigol is fairly inexpensive, with the primary associated drawback that its documentation can be vague and somewhat misleading as to the unit's true capabilities. The basic capabilities of each AWG is summarized in Table 1, with some commentary on the modes following.

Table 1—Summary of AWG Modes and Capabilities

Parameter	SiFi AWG Mode	DDS AWG Mode	Noise Mode
Sample rate	≤60 MSa/s	200 MSa/s	200 MSa/s
Output BW (one-sided)	≤30 MHz	≈90 MHz	≈90 MHz
Memory depth	8 MiSa	8 KiSa	∞
DAC Resolution	14 bit	14 bit	14 bit

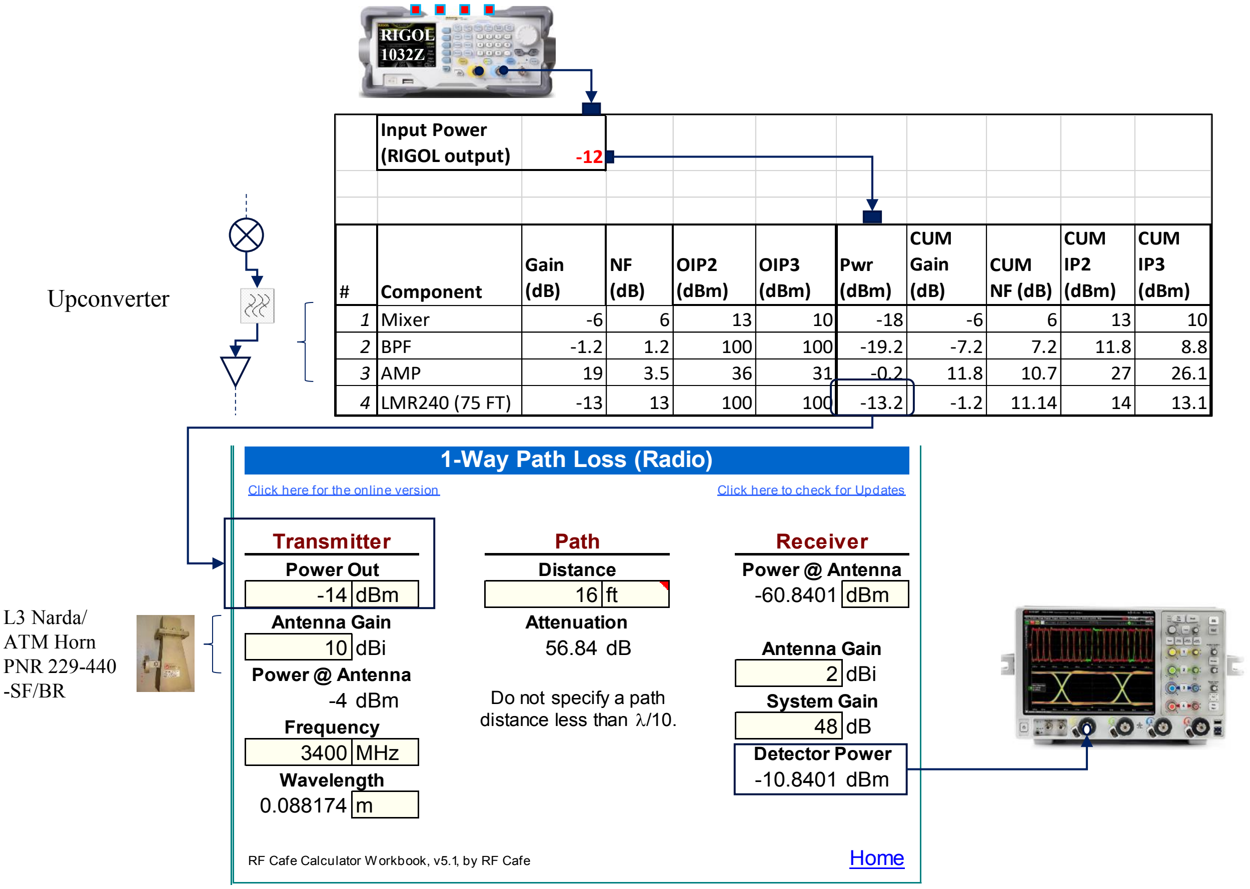
The Rigol has two arbitrary-waveform modes, named “SiFi” and “DDS” in the manual. In SiFi mode, the full 8 MiSa memory depth is available for waveforms with an adjustable sample rate; however, the sample rate is limited to ≤60 MSa/s. This mode thus supports less than 30 MHz of bandwidth, although the second-Nyquist region does not appear to be filtered beyond the sinc rolloff due to the DAC and thus could be used with appropriate bandpass filtering. The DDS mode uses the maximum DAC sample rate of 200 MSa/s, which is fixed, and has output filtering that rolls off around 90 MHz. In this mode, however, only a fixed 8,192-sample waveform segment is allowed; this is quite unclear in the documentation and significantly limits the usefulness of the unit as an AWG. Given our frequency plan, this is the AWG mode we wound up using despite the limitations; it produces radar pulses up to 40.96 µs wide at a fixed PRF of 24.4 kHz, both of which are reasonable values for a medium-PRF radar waveform. The AWG waveforms in either mode can be repeated any number of times and triggered with an adjustable delay.

In addition to the arbitrary-waveform modes, the Rigol also has a noise mode, which produces free-running white noise controlled by a gate signal. Although not well specified in the manual, analysis of the output indicates that the noise is generated digitally and played out of the DAC at 200 MSa/s, with analog filtering limiting the output to around 90 MHz, similar to the DDS mode.

The waveforms need to be generated at an IF prior to upconversion in order to effect quadrature modulation with a single channel and a single LO. Given the 200 MSa/s sampling rate, the natural choice is $f_s/4 = 50$ MHz. For pulsed waveforms, this first upconversion is applied directly to the AWG data. For noise waveforms, no such modulation is needed (or possible), as the upconverter filtering will simply select a band of noise around 50 MHz.

5.1.2 Upconverter

The eight-channel upconverter for the scene generator is a relatively straightforward custom design, with the schematic shown in the middle of Fig. 20 and the cascade analysis shown in Fig. 21. The 50 MHz IF input from the Rigol AWGs is mixed with a 3350 MHz local oscillator to generate the final 3400 MHz RF output. The mixer output is bandpass filtered and then amplified. The mixer and amplifier are stock Minicircuits components, but the sharp filtering requirements necessitated custom filters (Reactal 4C7-3400-X20), with the frequency response shown in Fig. 22. The filters have a flat passband over 20 MHz, with an approximate 3 dB bandwidth of 30 MHz. Stopband suppression of the image at 3300 MHz is better than 70 dB.



Input Power (RIGOL output)	-12

#	Component	Gain (dB)	NF (dB)	OIP2 (dBm)	OIP3 (dBm)	Pwr (dBm)	CUM Gain (dB)	CUM NF (dB)	CUM IP2 (dBm)	CUM IP3 (dBm)
1	Mixer	-6	6	13	10	-18	-6	6	13	10
2	BPF	-1.2	1.2	100	100	-19.2	-7.2	7.2	11.8	8.8
3	AMP	19	3.5	36	31	-0.2	11.8	10.7	27	26.1
4	LMR240 (75 FT)	-13	13	100	100	-13.2	-1.2	11.14	14	13.1

1-Way Path Loss (Radio)

Click here for the online version

Click here to check for Updates

Transmitter

Power Out: -14 dBm

Antenna Gain: 10 dBi

Power @ Antenna: -4 dBm

Frequency: 3400 MHz

Wavelength: 0.088174 m

Path

Distance: 16 ft

Attenuation: 56.84 dB

Do not specify a path distance less than λ/10.

Receiver

Power @ Antenna: -60.8401 dBm

Antenna Gain: 2 dBi

System Gain: 48 dB

Detector Power: -10.8401 dBm

RF Cafe Calculator Workbook, v5.1, by RF Cafe

Home

Fig. 21—The top table shows a cascade analysis of the upconverter, from Rigol output signal to horn antenna. The bottom window traces power levels through free space to the array antenna and receiver.

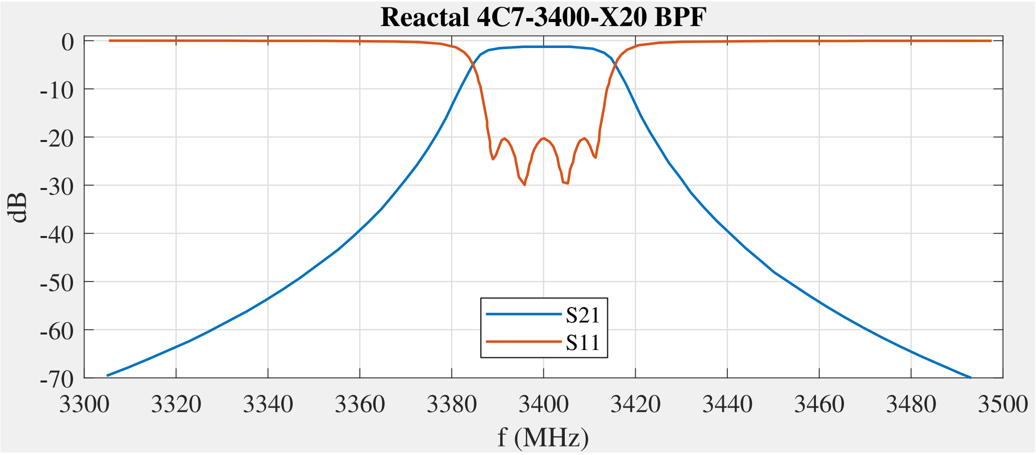


Fig. 22—Frequency response of Reactal bandpass filter

5.1.3 Valon 5009 Dual RF Frequency Synthesizer

The LOs for each pair of upconverter channels were supplied by a Valon 5009 dual RF frequency synthesizer, which provides a compact and economical source for a pair of LOs with both programmable frequency and power level. The Valon can be programmed over a serial/USB connection. Its relevant parameters are given in Table 2. The available frequency steps depend on both the frequency output range and the reference frequency; Table 2 shows the values for RF above 3 GHz and with both an external 10 MHz reference (as used here) as well as the internal 20 MHz reference in the Valon. The documented steps for 10 MHz and 20 MHz are 4,884 Hz and 9,768 Hz, respectively, and these were verified experimentally. However, undocumented steps of 5,000 Hz and 1,000 Hz also were confirmed. Since $4,884\text{Hz} = 4\,\text{MHz}/819$, the frequency steps cycle every 4 MHz for the 10 MHz reference. Conveniently, this meant that at our LO frequency of 3.35 GHz, the available offsets are $\{\ldots, -7328, -5000, -2444, 0, 2442, 5000, 7326, \ldots\}$ Hz, providing better than the advertised resolution. These offsets were used to impose virtual Doppler shifts on radar targets.

Table 2—Summary of the Programmable Parameters of the Valon 5009 Frequency Synthesizer

Parameter	10 MHz Reference	20 MHz Reference
Frequency range	23.5 MHz to 6 GHz	23.5 MHz to 6 GHz
Frequency step (RF above 3 GHz)	4884 Hz and 5000 Hz	9768 Hz and 10 000 Hz
Power range	−15 dBm to 15 dBm	−15 dBm to 15 dBm
Power step	0.5 dBm	0.5 dBm
Phase Noise @ 3 GHz + 1 kHz	−92 dBc/Hz	−92 dBc/Hz
Phase Noise @ 3 GHz + 1 MHz	−130 dBc/Hz	−130 dBc/Hz
Non-harmonic spurs	< -60 dBc	< -60 dBc

5.1.4 Transmit Horns

The transmit horns used in this project are L3 Narda/ATM S-band horn antennas, model 229-440, as shown in Fig. 23. These have a gain of 10 dB and beamwidths greater than 50° in both the E- and H-planes, thus requiring no precision in aiming them. They are mounted easily to provide either vertical or horizontal polarization; here, the latter was used to match the array.

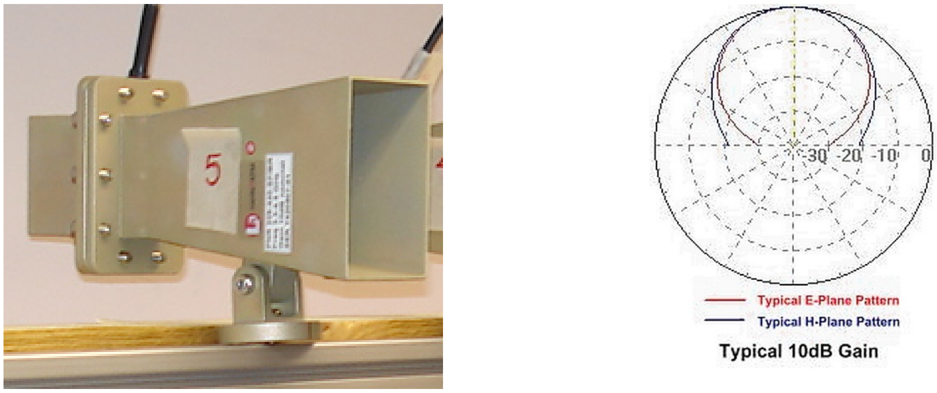


Fig. 23—The S-band horn antenna used for the far-field sources, and its typical principal-plane patterns

5.2 Array and Receiver

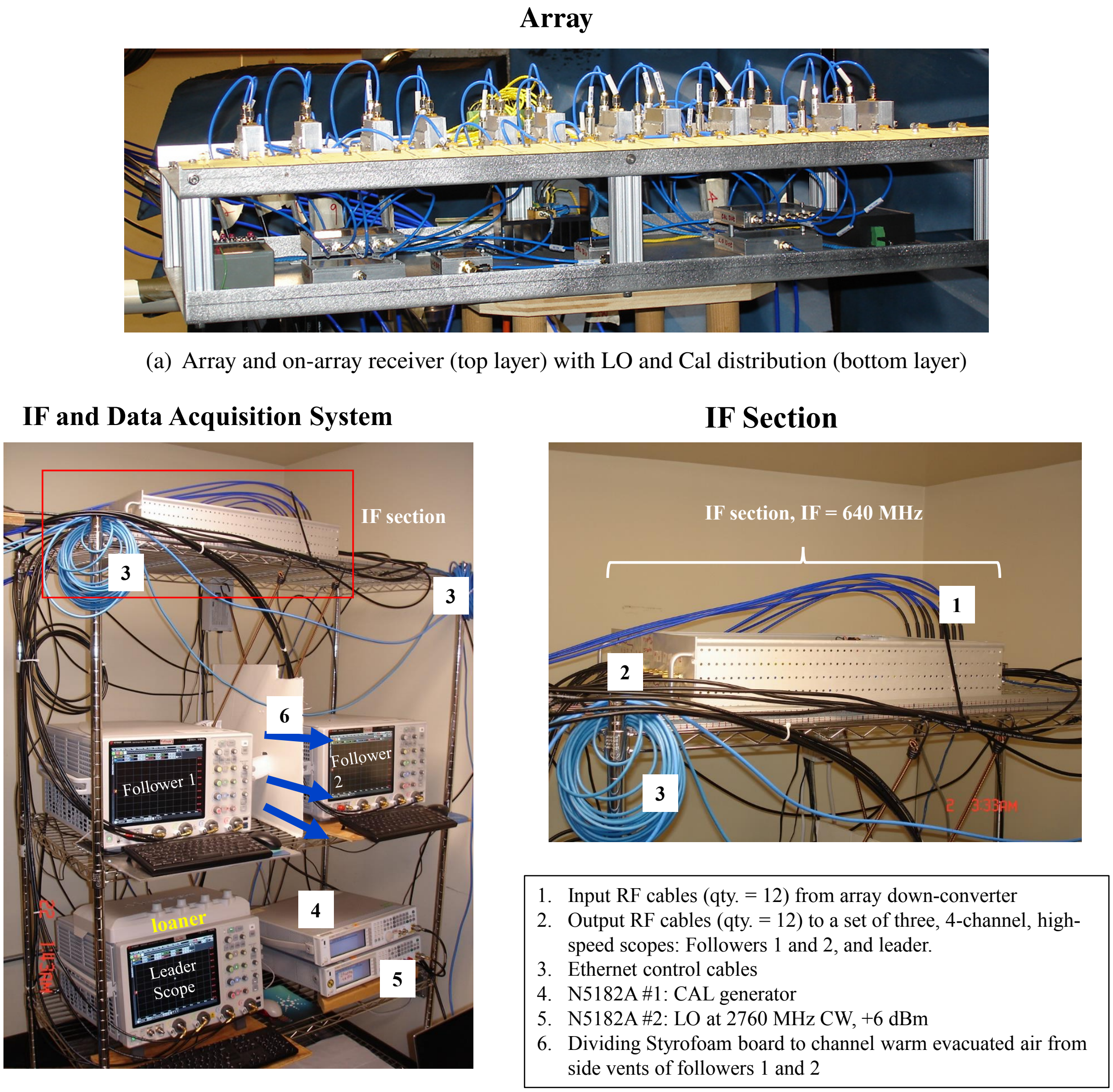
The array, array receiver, and associated hardware form the second major hardware component of the testbed. To demonstrate the sparse processing techniques, we required simultaneous capture of sufficient elements to represent useful sparse configurations. The number of channels captured from the array was dictated largely by the cost of implementing the corresponding receivers. Ultimately, 12 channels were implemented, as this allowed both a six-element nested configuration and the 12-element ULA of the same spatial extent to be captured simultaneously.

The array receiver is distributed between an on-array receiver section, an off-array IF receiver section, and a 12-channel IF digital receiver in the form of three digital sampling scopes. In addition, two vector signal generators provide the 10 MHz reference, LO, trigger, and calibration waveforms, which then are distributed across the system. Figure 24 provides a pictoral overview of the major components, which are discussed individually below.

5.2.1 Array and On-Array Receiver

The array elements used are simple printed dipoles, as shown in Fig. 25. The 18-element board was repurposed from a previous project. As used here, the outer three elements on each end were terminated to make the active element patterns more uniform. The central 12 elements were used for capture and beamforming. A future user easily could reconfigure the 12 active channels as desired.

Immediately behind the array elements lies the 12-channel on-array receiver. The purpose of the on-array receiver is to provide filtering, amplification, and downconversion from the 3,400 MHz RF to the 640 MHz IF prior to the cable run to the IF receiver section. This greatly reduces cable losses for longer runs and allows the use of inexpensive cables. This design initially was intended for an outdoor experiment, with the off-array components housed in a trailer, although we ultimately chose to perform the tests indoors, instead. Figure 26 shows the schematic for the on-array receiver at lower right. Each channel consists of a calibration injection coupler, a highpass/lowpass pair of filters, a LNA, and a mixer to convert from the 3400 MHz RF down to the 640 MHz IF. The left side of Fig. 26 shows the distribution of the 2,760 MHz local oscillator (LO) as well as the calibration waveform injected into each coupler. We will defer discussion of these items to Section 5.2.4. The cascade analysis for both the on-array and IF receivers is shown in Fig. 27, with the input power level assumed coming from the cascade analysis of Fig. 21.



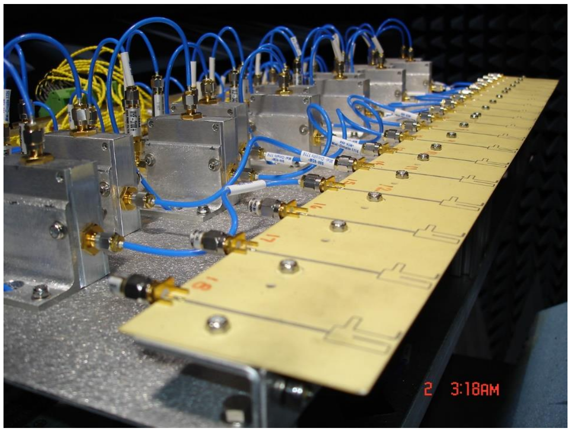
(a) Array and on-array receiver (top layer) with LO and Cal distribution (bottom layer)

1. Input RF cables (qty. = 12) from array down-converter
2. Output RF cables (qty. = 12) to a set of three, 4-channel, high-speed scopes: Followers 1 and 2, and leader.
3. Ethernet control cables
4. N5182A #1: CAL generator
5. N5182A #2: LO at 2760 MHz CW, +6 dBm
6. Dividing Styrofoam board to channel warm evacuated air from side vents of followers 1 and 2

(b) Off-array receiver components: IF receiver section, digital sampling oscilloscopes, LO generator, Cal generator

Fig. 24—Array and receiver overview

S-band, Linear Array Antenna
Code 5317 (Dorsey & Pickles)



- 18 dipole elements
  - Micro-strip
  - Multilayer PCB
- Element spacing = 4.6cm, ~λ/2 @ 3258 MHz
- Carrier frequency = 3400 MHz
- Elemental gain ~ 2 dBi
- Bandwidth ~ 1 GHz
- Physical length = 83.3 cm
- Width = 7.1 cm
- Twelve center elements used
- Three elements on either end of array are terminated into 50 ohms
- Power handling ~ 2 to 3 W

Fig. 25—Array elements and key parameters

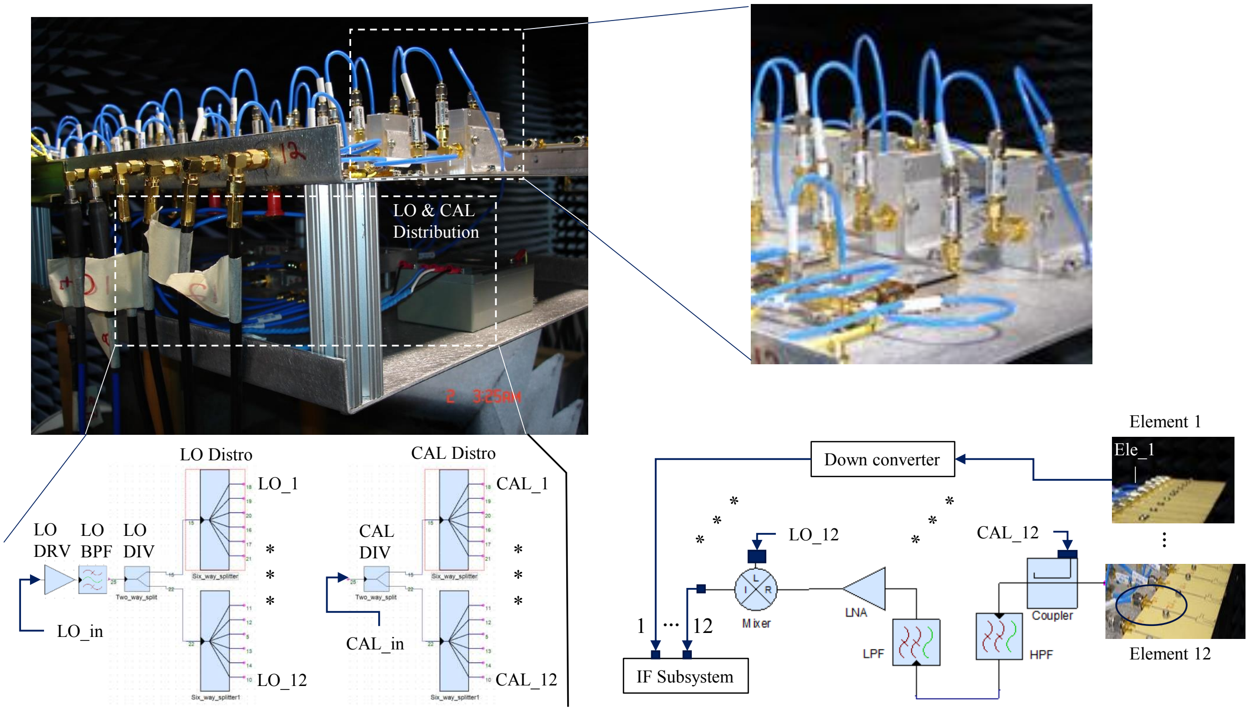
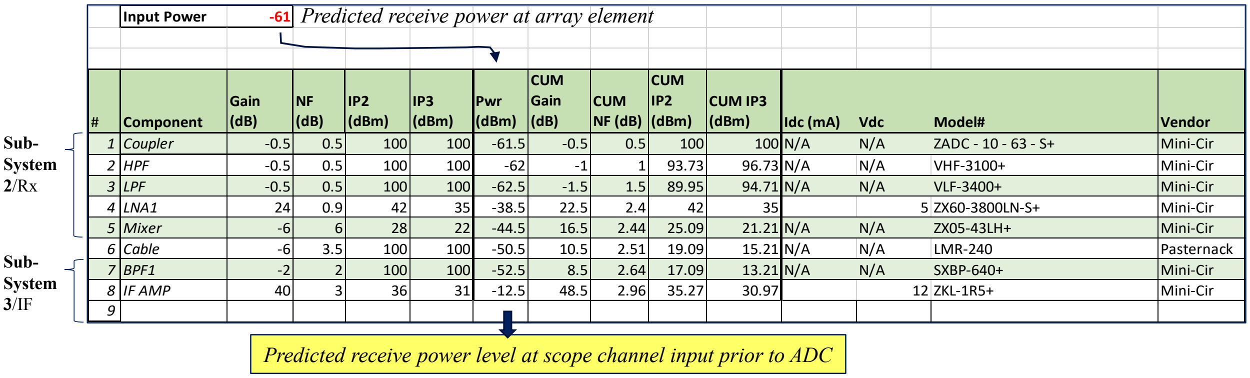


Fig. 26—Details of on-array receiver with LO and Cal distribution



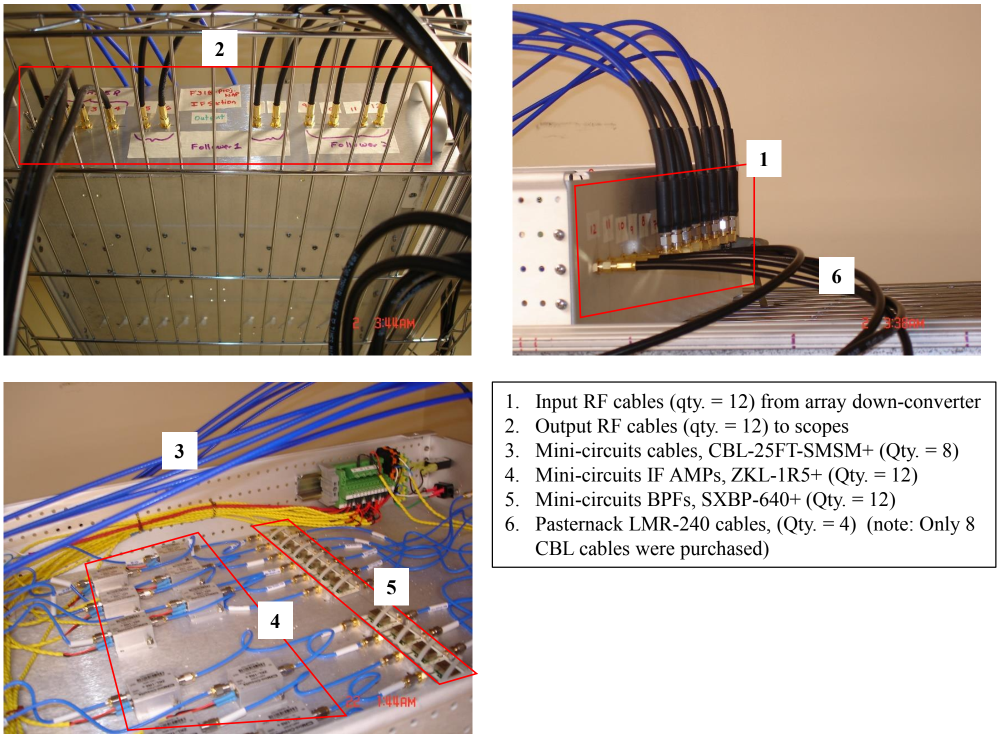
	Input Power	-61	*Predicted receive power at array element*										
#	Component	Gain (dB)	NF (dB)	IP2 (dBm)	IP3 (dBm)	Pwr (dBm)	CUM Gain (dB)	CUM NF (dB)	CUM IP2 (dBm)	CUM IP3 (dBm)	Idc (mA)	Vdc	Model#	Vendor
1	*Coupler*	-0.5	0.5	100	100	-61.5	-0.5	0.5	100	100	N/A	N/A	ZADC - 10 - 63 - S+	Mini-Cir
2	*HPF*	-0.5	0.5	100	100	-62	-1	1	93.73	96.73	N/A	N/A	VHF-3100+	Mini-Cir
3	*LPF*	-0.5	0.5	100	100	-62.5	-1.5	1.5	89.95	94.71	N/A	N/A	VLF-3400+	Mini-Cir
4	*LNA1*	24	0.9	42	35	-38.5	22.5	2.4	42	35		5	ZX60-3800LN-S+	Mini-Cir
5	*Mixer*	-6	6	28	22	-44.5	16.5	2.44	25.09	21.21	N/A	N/A	ZX05-43LH+	Mini-Cir
6	*Cable*	-6	3.5	100	100	-50.5	10.5	2.51	19.09	15.21	N/A	N/A	LMR-240	Pasternack
7	*BPF1*	-2	2	100	100	-52.5	8.5	2.64	17.09	13.21	N/A	N/A	SXBP-640+	Mini-Cir
8	*IF AMP*	40	3	36	31	-12.5	48.5	2.96	35.27	30.97		12	ZKL-1R5+	Mini-Cir
9														

Sub-System 2/Rx
Sub-System 3/IF
Predicted receive power level at scope channel input prior to ADC

Fig. 27—Receiver cascade analysis

5.2.2 IF Receiver Section

The IF receiver section provides additional signal conditioning (filtering and amplification) prior to the digital receivers (scopes). Implementing these functions off the array conserves space and, as mentioned previously, would allow for widely separating the on-array and off-array components, if needed. Details of the 12 channels into and out of the IF receiver box are shown in Fig. 28 along with the internal circuits. Because of the differing geometry between the original proposed outdoor tests and the actual chamber tests,



1. Input RF cables (qty. = 12) from array down-converter
2. Output RF cables (qty. = 12) to scopes
3. Mini-circuits cables, CBL-25FT-SMSM+ (Qty. = 8)
4. Mini-circuits IF AMPs, ZKL-1R5+ (Qty. = 12)
5. Mini-circuits BPFs, SXBP-640+ (Qty. = 12)
6. Pasternack LMR-240 cables, (Qty. = 4) (note: Only 8 CBL cables were purchased)

Fig. 28—IF receiver section detail

we repurposed eight of the long RF cables originally intended to span between the array and IF section to drive the far-field horns and replaced them with shorter cables. Calibration served to equalize out any differences between the two cable types.

5.2.3 Agilent/Keysight Infinium DSO/DSA

Since a dedicated 12-channel digital receiver was not readily available for a reasonable cost, we used three available 4-channel Keysight (formerly Agilent) Infinium digital oscilloscopes as digital receivers. The models used were V-series models 334A and 204A, but any of the V-series and some of the S-series will give comparable results and use a compatible command set. Some relevant parameters are given in Table 3.

Table 3—Agilent/Keysight Infinium Parameters and Modes

Parameter	Real-Time (standard) mode	High-Resolution mode
Sample rate	≤80 GSa/s	≤40 GSa/s
BW	20 GHz (204A), 33 GHz (334A)	Varies (lower)
Memory depth	512 MiSa per channel	256 MiSa per channel
ADC Resolution	8 bit	up to 12 bit

The scopes have two capture modes of interest here: Real-time mode is the standard mode, which returns ADC samples at the chosen rate without temporal filtering, while high-resolution mode averages blocks of adjacent samples and outputs one sample per block. In real-time mode, the ADC can be run at a variety of rates, while in high-resolution mode, the ADC is always run full-speed internally, with the output sample rate a fraction of the full rate. The more samples that are averaged, the higher the effective resolution and the lower the output rate. The simple averaging filter also causes a significant reduction in effective bandwidth; thus, while real-time mode at 2 GSa/s is sufficient to capture our receive IF of 640 MHz, in high-resolution mode an output sample rate of 5 GHz is required, as the filter bandwidth is much less than the Nyquist frequency. This drives up the number of samples required to be captured in high-resolution mode, so the mode is most useful when capturing for relatively short durations.

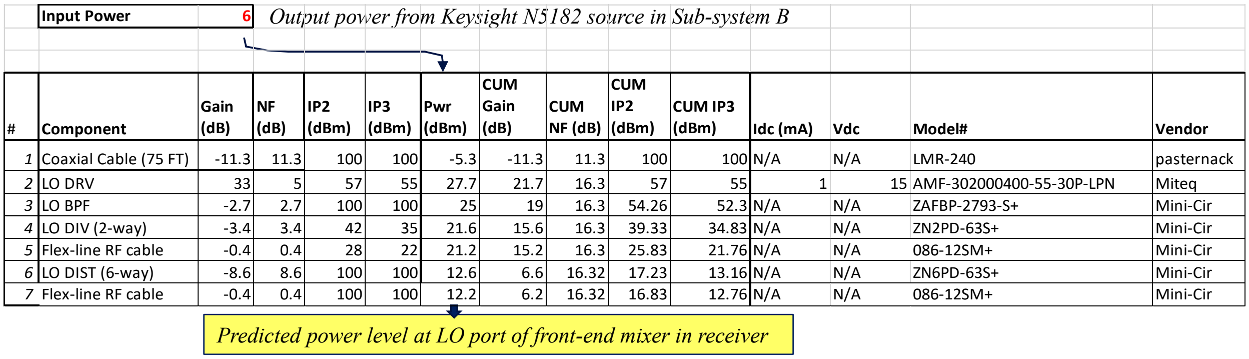
5.2.4 Agilent N5182A VSG w/ AWG

The Agilent N5182A is a vector signal generator with an integrated AWG providing baseband I and Q signals to the modulator. Two units were used in the experimental setup; the first provided the 2,760 MHz LO for the array receivers, while the second one was used to produce a loopback calibration waveform as well as to generate the master trigger for the system. Its relevant parameters are summarized in Table 4.

The LO from the first VSG is sent by cable to the on-array receiver, where it is amplified, filtered, and split 12 ways. The distribution network is shown at lower left in Fig. 26, while the cascade analysis is given in Fig. 29. The second VSG was used in its AWG mode to generate a linear-FM calibration pulse with 40 MHz bandwidth centered at 3,400 MHz. This signal was routed to a cal distribution network on the array as shown at lower middle left in Fig. 26. The 12 outputs then were routed to the couplers behind each active element. This loopback calibration setup provided a common reference signal on each channel to equalize differences in the receiver responses, between the elements and the scopes. While it is not able to calibrate any differences in the element responses themselves, the uniformity of a printed array and the three dummy elements per side should minimize variation. One binary marker output was available from the VSG, and

Table 4—Agilent N5182A Vector Signal Generator Parameters

Parameter	Values
Center Frequency range	100 kHz to 6 GHz
Sample rate	125 MSa/s
Output BW (one-sided)	50 MHz
Memory depth	8 MiSa
DAC Resolution	16 bit
Phase Noise @ 3.5 GHz + 1 kHz	−105 dBc/Hz
Phase Noise @ 3.5 GHz + 1 MHz	−130 dBc/Hz



Input Power 6 *Output power from Keysight N5182 source in Sub-system B*

#	Component	Gain (dB)	NF (dB)	IP2 (dBm)	IP3 (dBm)	Pwr (dBm)	CUM Gain (dB)	CUM NF (dB)	CUM IP2 (dBm)	CUM IP3 (dBm)	Idc (mA)	Vdc	Model#	Vendor
1	Coaxial Cable (75 FT)	-11.3	11.3	100	100	-5.3	-11.3	11.3	100	100	N/A	N/A	LMR-240	pasternack
2	LO DRV	33	5	57	55	27.7	21.7	16.3	57	55	1	15	AMF-302000400-55-30P-LPN	Miteq
3	LO BPF	-2.7	2.7	100	100	25	19	16.3	54.26	52.3	N/A	N/A	ZAFBP-2793-S+	Mini-Cir
4	LO DIV (2-way)	-3.4	3.4	42	35	21.6	15.6	16.3	39.33	34.83	N/A	N/A	ZN2PD-63S+	Mini-Cir
5	Flex-line RF cable	-0.4	0.4	28	22	21.2	15.2	16.3	25.83	21.76	N/A	N/A	086-12SM+	Mini-Cir
6	LO DIST (6-way)	-8.6	8.6	100	100	12.6	6.6	16.32	17.23	13.16	N/A	N/A	ZN6PD-63S+	Mini-Cir
7	Flex-line RF cable	-0.4	0.4	100	100	12.2	6.2	16.32	16.83	12.76	N/A	N/A	086-12SM+	Mini-Cir

Predicted power level at LO port of front-end mixer in receiver

Fig. 29—LO distribution cascade analysis

was used as the master trigger for the system. The trigger was distributed to the scopes as well as to the eight Rigol channels in the scene generator, as shown in Fig. 30. In order to handle the large fanout, multiple Pulse Research Labs TTL line drivers were used.

The triggering and calibration scheme that was used is illustrated in Fig. 31. The marker output from the calibration VSG is shown at top, while the bottom curve shows a typical capture at one scope channel. The marker is held high prior to a capture. When a trigger is initiated via software at $t = 0$, the marker goes low and the falling edge triggers any Rigol channels that are configured in DDS mode and the oscilloscopes. Rigol channels configured as noise sources are controlled by a gate level, rather than by trigger edges, and are turned off at this time. After a delay of 100 µs, the VSG AWG plays out the 200 µs loopback calibration pulse, and then after another delay of 100 µs, the marker returns high. This enables the Rigol noise sources, which will play out until the next capture cycle, and marks the start of the main data capture. The Rigol DDS sources have a programmable delay parameter, which is set by default to 400 µs so that the first pulse is aligned with the noise sources. This delay can be modified to simulate target range, or to provide a second effective calibration pulse. One of the far-field DDS sources can be set to have one PRI-less delay, thus playing out the first pulse in the dead time between the loopback calibration waveform and enabling of the other sources. This provides the capability to transmit a single far-field calibration pulse. Unlike the loopback cal, this pulse does include the response of the array elements, possibly eliminating a source of mismatch. However, it potentially adds new sources of channel mismatch due to differences in propagation to each element, and it adds the frequency response of the upconverter and horn antenna (to be fair, the loopback cal includes the VSG AWG response and coupler mismatches as well). Ultimately, the loopback cal was used throughout the experiments, while the far-field cal was used primarily for sanity checking.

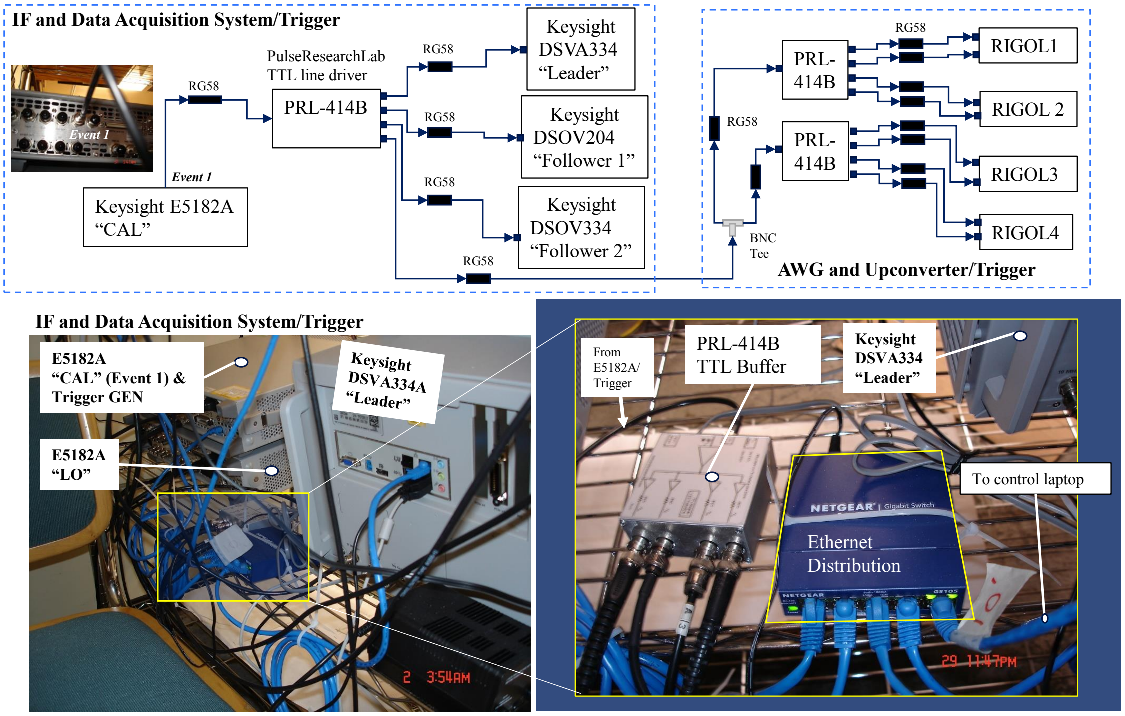

IF and Data Acquisition System/Trigger
Event 1
RG58
Event 1
Keysight E5182A "CAL"
PulseResearchLab TTL line driver
PRL-414B
RG58
RG58
RG58
RG58
Keysight DSVA334 "Leader"
Keysight DSOV204 "Follower 1"
Keysight DSOV334 "Follower 2"
RG58
RG58
PRL-414B
PRL-414B
RIGOL1
RIGOL 2
RIGOL3
RIGOL4
BNC Tee
AWG and Upconverter/Trigger
IF and Data Acquisition System/Trigger
E5182A "CAL" (Event 1) & Trigger GEN
Keysight DSVA334A "Leader"
E5182A "LO"
2 3:54AM
From E5182A/ Trigger
PRL-414B TTL Buffer
Keysight DSVA334 "Leader"
To control laptop
Ethernet Distribution
NETGEAR
29 11:47PM

Fig. 30—Trigger distribution

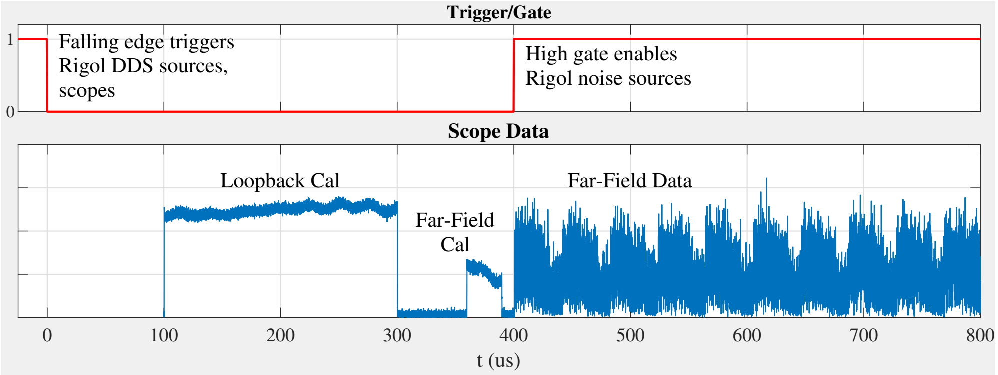


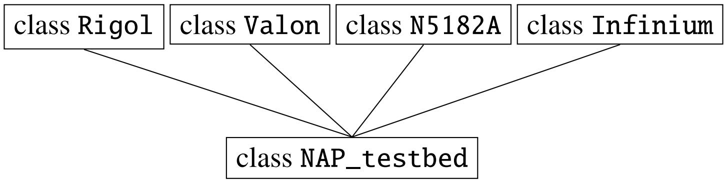
Fig. 31—Trigger and calibration timing

5.3 Software

One of the requirements when assembling the various pieces of equipment for these experiments was that everything would be remotely programmable, preferably through a simple text interface such as SCPI. As we planned to perform many captures, we needed a way to automate the process. The integration was done using Matlab and its Instrument Control Toolbox. The software, itself, will not be fully documented here, but an overview of the code is given below.

5.3.1 Class Hierarchy

For each of the major pieces of equipment, a Matlab class was written to encapsulate the functionality needed for the experiments. The authors have found that this approach has several benefits: It allows for a reasonably uniform interface to equipment with potentially very different underlying command sets, it serves as a compact self-documenting distillation of the (often quite lengthy) programming guide for each piece of equipment, and it simplifies subsequent uses of the equipment. A class `NAP_testbed` then was written to integrate the individual classes and to contain all of the experiment-specific code. The class hierarchy is therefore:



The headers (property definitions) for each of the classes are typeset in Appendix E for readers curious as to the available settings and parameters. The full code package is available from the authors upon request.

5.3.2 Configuration Files

To encapsulate all the settings for the eight far-field sources, a configuration file format was defined. This allows easy parameter changes and leaves behind documentation of the settings used for a given capture. The config files use a standard line-based comma-separated value (CSV) format with eight fields:

1. Channel number: (1–8)
2. Waveform type: (dds,noise,off)
3. Arbitrary waveform file (dds mode only): local .mat file
4. Apparent target range (dds mode only): range in meters
5. Apparent target velocity (dds mode only): Doppler offset in Hz
6. Power level at AWG: power in dBm
7. Number of pulses Kp (dds only): positive integer (no cal), negative integer (w/ cal), 0 (cal only)
8. Position of source horn antenna along rail: signed offset in cm

Many of the settings are specific to the DDS AWG mode, as the noise mode does not provide any way to delay or pulse the output and applying a Doppler shift to the noise will have no significant effect. Any settings that don't apply to a particular mode are ignored, as are any lines beginning with "#". Nonzero range values are converted to an equivalent delay in the corresponding Rigol DDS channel, while nonzero Doppler values are converted to an equivalent offset in the corresponding Valon LO. The absolute value of the seventh field gives the number of DDS pulses that are played out after the marker gate goes high (turning on any noise sources). If the number is negative, an extra calibration pulse is enabled, which fires one PRI prior to the gate. If the number is zero, then only a calibration pulse is sent, after which the source remains off. The eighth field records the position of the corresponding horn antenna along the 80/20 rail in centimeters. This has no effect on the waveforms, but is merely a bookkeeping convenience. An example config file is shown below; it enables two pulsed (DDS) waveforms, two noise sources, and one far-field cal pulse:

```
#Ch    Type                Arb file          Rng (m)    Dop (Hz)   Pow (dBm)    Kp    Pos (cm)
  1,    dds,   chirp_20MHz_40us_200MSps.mat,    5000,      10000,      -45.0,   100,       80
  2,    dds,   chirp_40MHz_40us_200MSps.mat,       0,          0,      -15.0,     0,       40
  3,  noise,                               ,       0,          0,      -35.0,     1,       20
  4,    off,                               ,       0,          0,      -15.0,     1,        0
  5,  noise,                               ,       0,          0,      -25.0,     1,      -20
  6,    off,                               ,       0,          0,      -15.0,     1,      -40
  7,    off,                               ,       0,          0,      -15.0,     1,      -80
  8,    dds,   chirp_20MHz_40us_200MSps.mat,   52500,          0,      -60.0,   100,     -160
```

5.3.3 A Minimum Working Example

With the equipment-control code tucked into the class library and use of the configuration files, a script to perform a capture can be very compact. All that remains is to initialize the hardware, to choose a few run-time parameters, to load the configuration file, to set up the scopes and calibration, and to trigger and retrieve the data. A minimum working example is given below:

```
% MWE.m: A minimum working example for using the NAP_testbed class

%% Units
m=1; cm=1e-2*m; in=2.54*cm; ft=12*in;
s=1; ms=1e-3*s; us=1e-6*s; ns=1e-9*s;
Hz=1/s; kHz=1e3*Hz; MHz=1e6*Hz; GHz=1e9*Hz;

%% instantiate class
nap=NAP_testbed;
nap.open; % connect to and config equipment

%% set up cal, uses current parameters
nap.upload_Cal_Wfm;
nap.trigger; % dummy trigger here to set up the marker state correctly
pause(1);

%% scope mode
%nap.sampleMode='RTIM'; nap.fs_Capture=2*GHz; % for non-hires sampling
nap.sampleMode='HRES'; nap.fs_Capture=5*GHz; % for hi-res sampling

%% array parameters
nap.FFDist=196.5*in; % distance from array to baseline of far-field antennas

%% load config file and apply
nap.load_FF_config('MWE.conf');
nap.config_sources; % send parameters to sig gens

%% setup scopes and cal amplitude to match
nap.T_capture = duration(nap) + 100*us;
CaldB_offset=12; % magic number, empirical
scope_mult=1.5*[1 1 0.8]; % account for different cable loss to scopes
nap.Vpp_capture=scope_mult*max(0.15,nap.Vpp_estimate); % need to put a floor on the scope level
nap.VSG.Amplitude=20*log10(nap.Vpp_capture(1))-CaldB_offset; % scale cal pulse with scope
nap.config_DSO;
pause(1); % avoid lockups
```

```

%% trigger scopes
nap.trigger;
pause(3); % need this to prevent VISA/scope lockups

%% get data and save to file
y=nap.save_data('MWE');
```

5.4 Anechoic Chamber Setup and Test Procedures

The tests for the NAP project were carried out in the small anechoic chamber in room 16A3 of building A59 at the main NRL campus in Washington, D.C. A schematic of the system layout is shown in Fig. 32, where the left half of the figure is the chamber, and the right side is the adjacent equipment room (antechamber). Figure 33 shows photos of the setup. The array was mounted on a positioner at the front of the chamber and faced the rear of the chamber. An aluminum 80/20 rail was mounted horizontally along the back of the chamber, with the scene-generator horn antennas mounted so that they could be slid along the rail. This allowed for relatively precise positioning of the sources. The rail was located about 5 m (≈16 ft) from the array at its center and allowed linear offsets from array normal between −1.8 m and 2.1 m. This corresponds to angles between −20° and 23°. In later tests, the horizontal spread was increased by adding two tripods outside the ends of the rails and somewhat closer to the front of the chamber, yielding additional source angles of ≈ ±31°. Other than the rail, horns, and array assembly, the rest of the system was kept on racks in the antechamber.

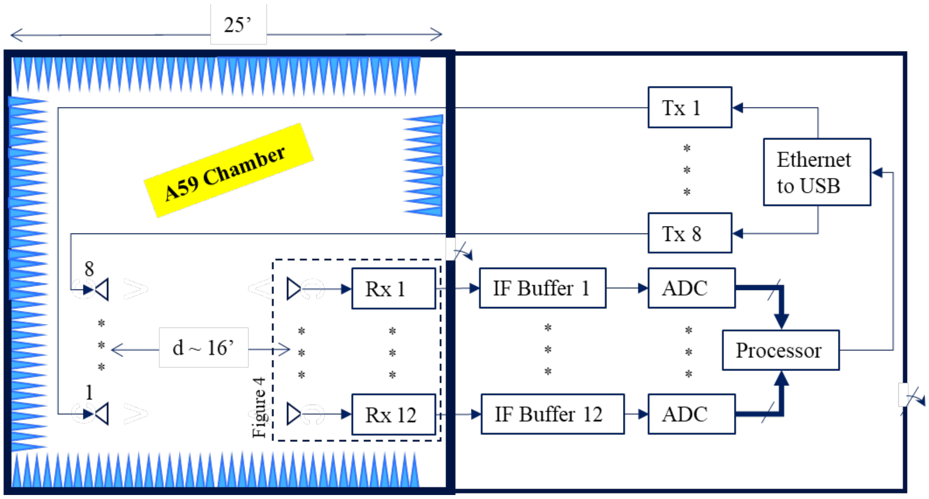


Fig. 32—High-level schematic of the experimental setup. Eight source horns were located at the back wall of the chamber, the array was mounted to the positioner at the front of the chamber, and the rest of the setup was located in the antechamber.

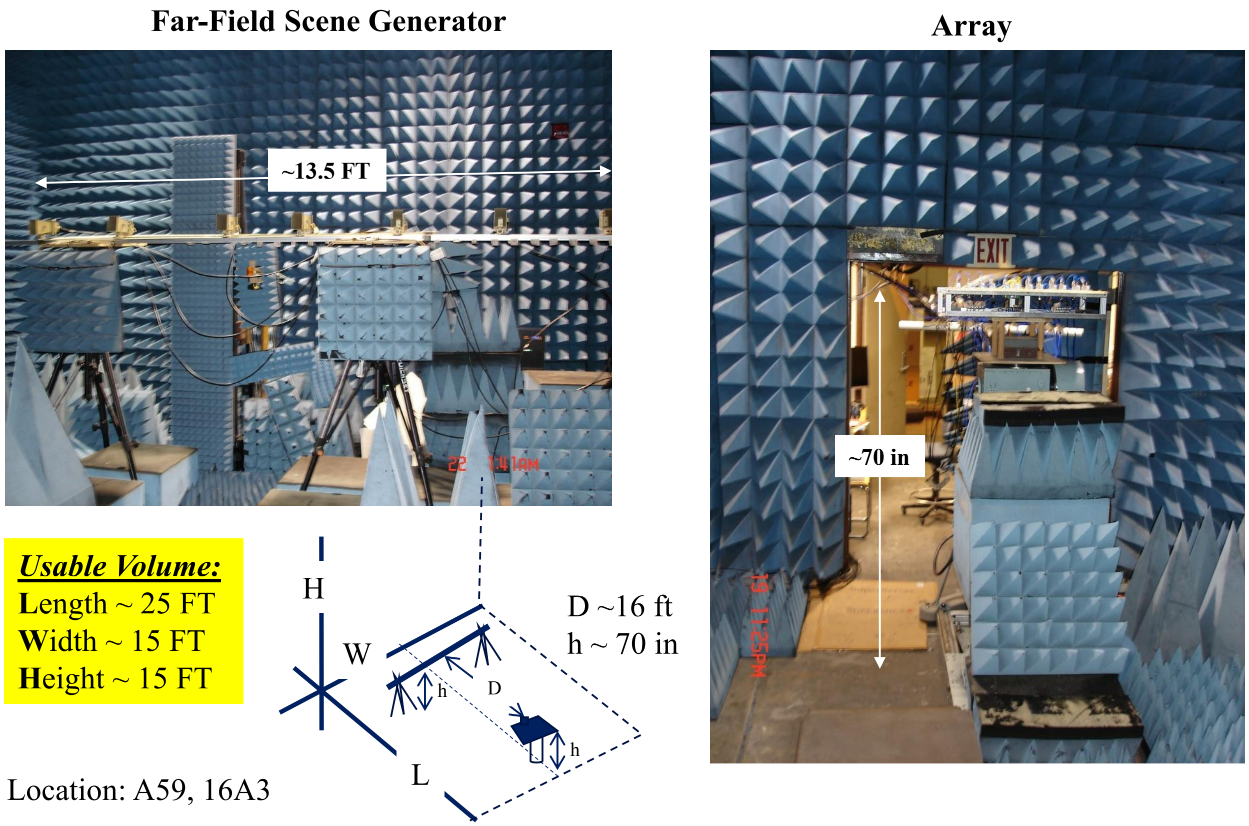


Fig. 33—Photos of the rail at the rear of the chamber with the horns mounted (left) and of the positioner with the array mounted (right)

A number of different tests were performed with the system between spring and fall 2018. These included:

Element pattern sweeps: The antenna is physically rotated by the positioner in 1° increments. At each step, a single, common probe pulse is sent out of all eight far-field sources with sufficient delay stagger to separate them in time.

Incoherent resolution tests: Two noise sources are activated at various separations and relative power levels.

Coherent resolution tests: Same as previous, but two sources are DDS sources transmitting the same waveform to simulate multipath.

Spectral Estimation Tests: Various numbers of noise sources at various positions and power levels.

Incoherent interference tests: One DDC source representing a radar return, and various numbers of interference noise sources with various power levels.

Coherent interference tests: Same as previous, except that some of the interference sources are copies of the radar signal to represent repeater jamming.

5.5 Element-Pattern and Steering-Vector Characterization

The element-pattern sweeps mentioned in the preceding section were performed in order to assess how the experimental setup behaved relative to an ideal linear array. Many potential sources of error exist, including nonuniformity in the array elements and their individual electromagnetic environment, multipath in the chamber, and geometrical considerations. The following sections provide an analysis of the expected and measured element patterns in the array.

5.5.1 Geometry Considerations

One known source of deviation between our array and an ideal array is due to the geometry; the $\approx 5\,\mathrm{m}$ distance from array center to the center of the rail with the source horns mounted on it does not meet the usual rule of thumb for the far-field:

$$2\frac{D^2}{\lambda} = 2\frac{(12d)^2}{\lambda} = 6.9\,\mathrm{m}.$$

Thus, our sources are technically in the Fresnel region of the array rather than the Fraunhofer (far-field) region. This means that the wavefronts arriving at the array are not quite planar; there is a residual range-dependent quadratic term. In this region, we assume that the amplitude at each element is constant but account for phase due to the actual distance from each element to the source position. Thus, the elements of the steering vector to a source located at range r and angle θ from the array origin are

$$[\mathbf{v}]_k(r,\theta) = e^{-j\frac{2\pi}{\lambda}(D_k(r,\theta)-r)}, \tag{72}$$

where D_k is the distance from the source to the kth element and where we normalize by removing the central distance r. For the element at coordinate $(x, y) = (x_k, 0)$, this yields

$$\begin{aligned} D_k(r,\theta) &= \left\| r\begin{pmatrix}\sin(\theta)\\ \cos(\theta)\end{pmatrix} - \begin{pmatrix}x_k\\ 0\end{pmatrix}\right\| \\ &= r\sqrt{1 - \frac{2\sin(\theta)x_k}{r} + \frac{x_k^2}{r^2}} \\ &\approx r\left(1 - \frac{\sin(\theta)x_k}{r} + \frac{x_k^2}{2r^2}\right) \\ &= r - \sin(\theta)x_k + \frac{x_k^2}{2r}, \end{aligned} \tag{73}$$

where we take the first three terms of the Taylor series of the square root in the third line. Substituting Eq. (73) into Eq. (72) yields the Fresnel steering vector

$$[\mathbf{v}]_k(r,\theta) = e^{j\frac{2\pi}{\lambda}\left(\sin(\theta)x_k - \frac{x_k^2}{2r}\right)}, \tag{74}$$

which, for large r, converges to the standard far-field steering vector

$$[\mathbf{v}]_k(r,\theta) = e^{j\frac{2\pi}{\lambda}\sin(\theta)x_k}. \tag{75}$$

For the 12 elements here, the far-field and Fresnel steering vectors at boresight are compared in Fig. 34. Since the distance from a boresight source to the outer elements is larger than to the center elements, they have a proportionally larger phase delay. The bottom plot shows the first-order phase difference between elements; since the "extra" Fresnel phase is quadratic, the difference is linear.

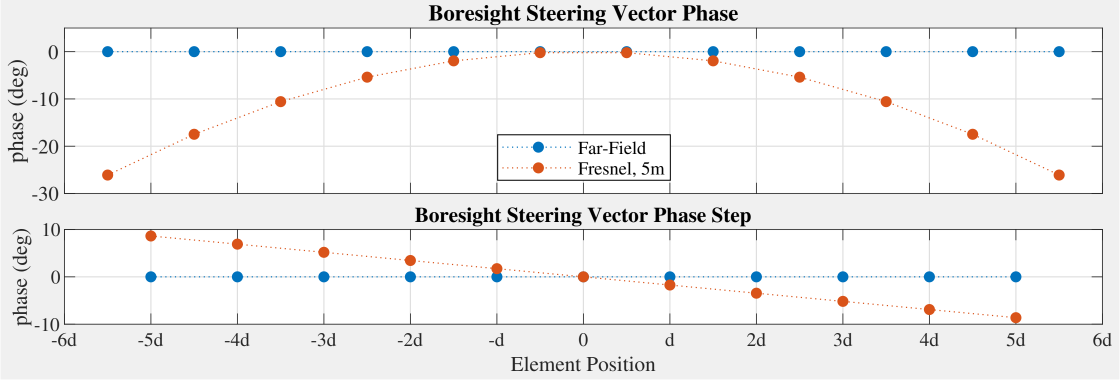


Fig. 34—Comparison of the phases and the interelement phase steps of the far-field and Fresnel boresight steering vectors

5.5.2 *Measured Element Patterns*

With the array mounted on a motorized positioner, it was a straightforward task to rotate it through 180° of azimuth in 1° steps to measure the embedded array element patterns. The scene generator was configured to transmit one 40 µs pulse out of each of the eight sources, staggered in time so there was no overlap. The sources were spread out over ±31° of azimuth, yielding eight independent captures at each angle. The pulses were equalized using the loopback calibration, compressed, and delay-compensated, and were angle-aligned to yield a $12 \times 8 \times 181$ data cube giving estimate of the complex response at each element from each source at each angle.

The measured magnitude responses of the element patterns are shown in Fig. 35. Each plot contains eight traces, one for each source; they are generally consistent other than overall scaling, which is primarily due to different power levels coming from the sources. The element patterns are, for our purposes, reasonably behaved over ±30° or so, with some exceptions. There are obvious differences between the elements, with some amount of mirror symmetry across the center of the array. The phase response of the elements is dominated by a linear slope that is proportional to $\sin(\theta)$, which tends to obscure the errors. Therefore, we instead plot the element-to-element phase differences for the measured responses, which removes the slope. The top plot of Fig. 36 shows the result using source #5 from angles in the range ±45°. In an ideal far-field array, all the lines would be horizontal with heights proportional to $\sin(\theta)$; here, the trend is roughly horizontal but we clearly have various errors. If we further remove the ideal vertical offset from each curve, what remains is just the phase errors, shown in the bottom plot of Fig. 36. Computing the linear trend (green line) shows that it almost exactly matches the expected slope (dashed, red line) due to the Fresnel-region geometry.

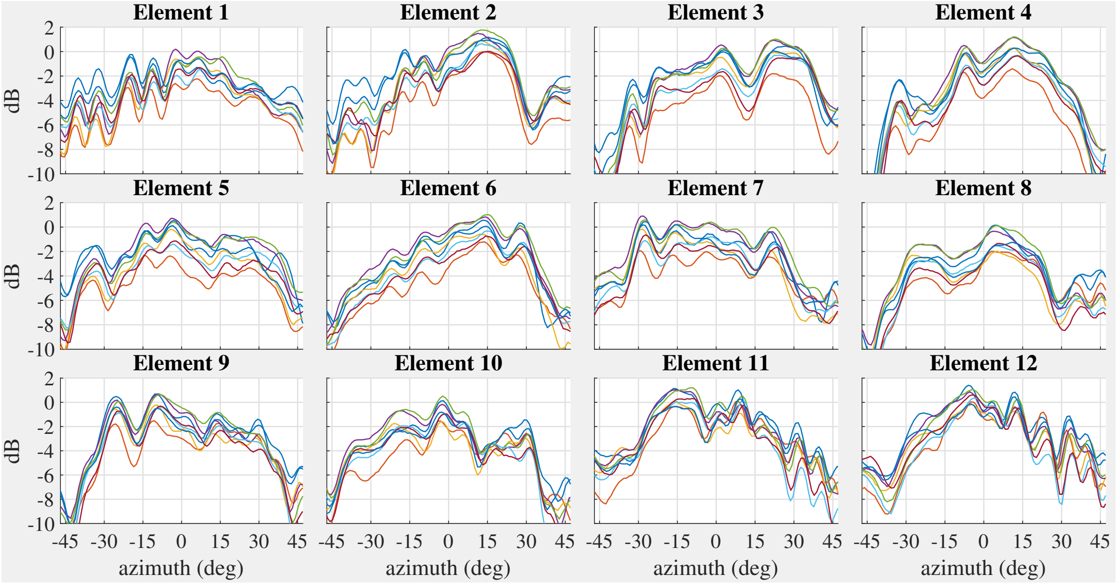


Fig. 35—The magnitude responses of the element patterns for all eight sources

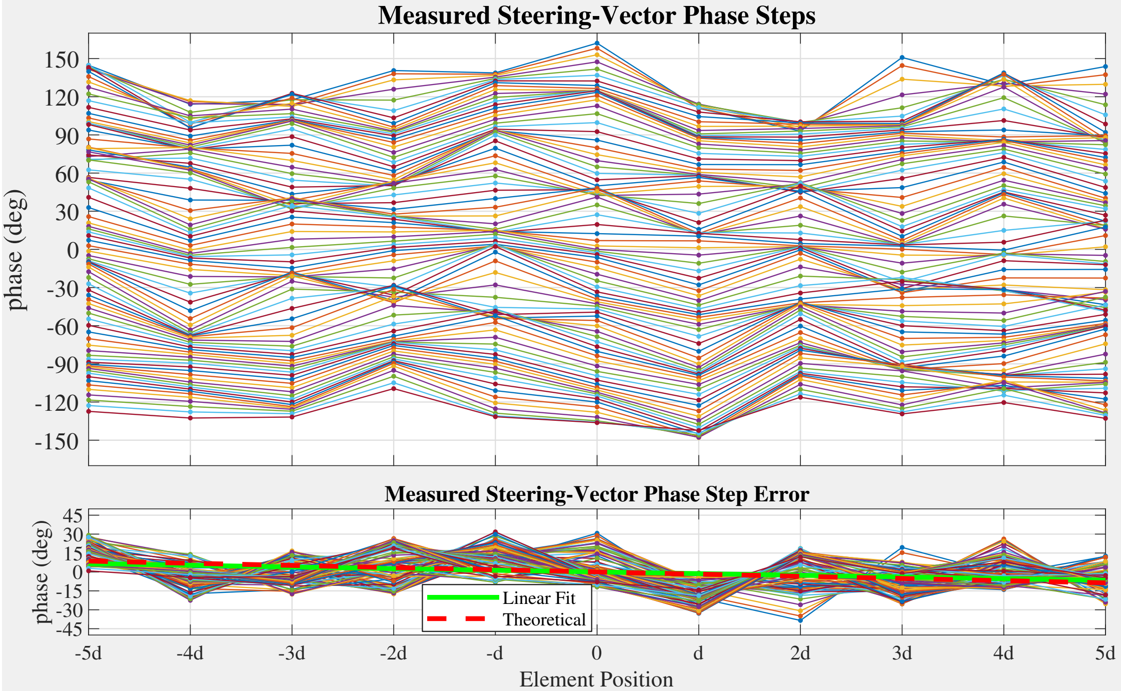


Fig. 36—The measured steering vector phases corresponding to source #5. At top are the measured phase increments between elements; in an ideal far-field array, these would be uniformly spaced horizontal lines. At bottom, the ideal values (means) have been removed to show only the phase-step errors. The green line shows the best-fit linear trend, while the dashed, red line indicates that this is an almost perfect match to the expected Fresnel-zone phase errors.

The preceding analysis suggests that there may be a benefit to using either the ideal Fresnel steering vectors or the measured array steering vectors instead of the standard far-field steering vectors when processing captured data. To investigate this for simple linear beamforming, we compare various cases in Fig. 37. The first plot is for reference and shows the response of an ideal array to a far-field source, where the processing is linear beamforming using ideal far-field steering vectors. We see the classic sinc structure cross each axis cut. The subsequent three plots show the result of beamforming actual measured data from the array using source #5. The second plot from the left uses far-field steering vectors, the third plot uses Fresnel steering vectors, and the fourth plot uses steering vectors measured using source #4. All three are quite similar, although the far-field steering vector case shows a degraded (filled-in) first null while the Fresnel vectors largely preserve the null. Interestingly, while the measured-vector case also preserves the null for small angles, at higher angles, it appears to degrade also. This likely represents phase and amplitude errors that are independent from measurement to measurement. Interestingly, this advantage for linear processing did not appear to carry over to the nonlinear algorithms. Using either the Fresnel or the measured steering vectors tended to make the algorithms less stable in terms of identifying source locations relative to the far-field vectors. This is likely because the far-field vectors provide a more uniform basis for the space of signals and noise even if they are not quite the best fit for the signals of interest.

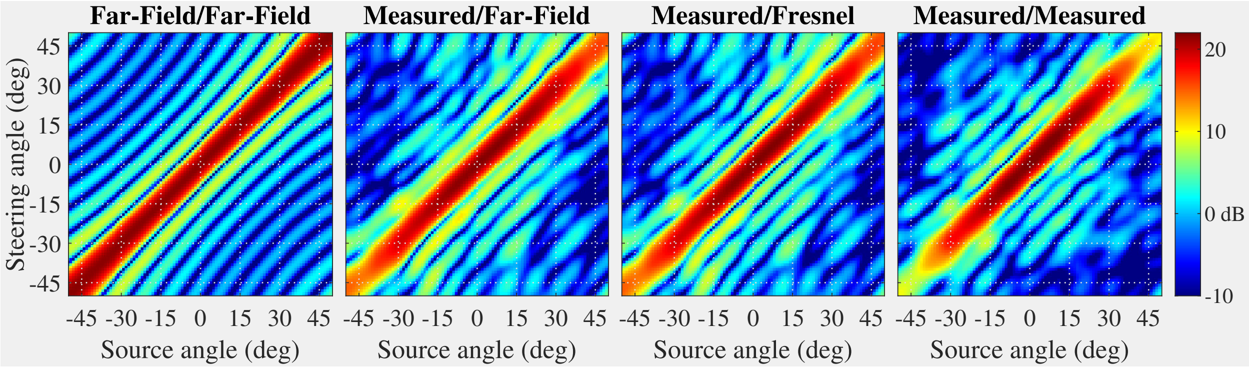


Fig. 37—A comparison of the beamforming response to different sets of steering vectors. At left is the reference response of an ideal array to a far-field source, beamformed using ideal far-field steering vectors. The subsequent plots show the measured response of the NAP array to Fresnel-zone source #5, beamformed using far-field, Fresnel, and measured (from source #4) steering vectors, respectively.

Finally, as a sanity check, the data from all eight sources was processed with ideal far-field steering vectors to verify roughly that their position was registered properly. The results are shown in Fig. 38, with the white, dashed line indicating identical values on both axes.

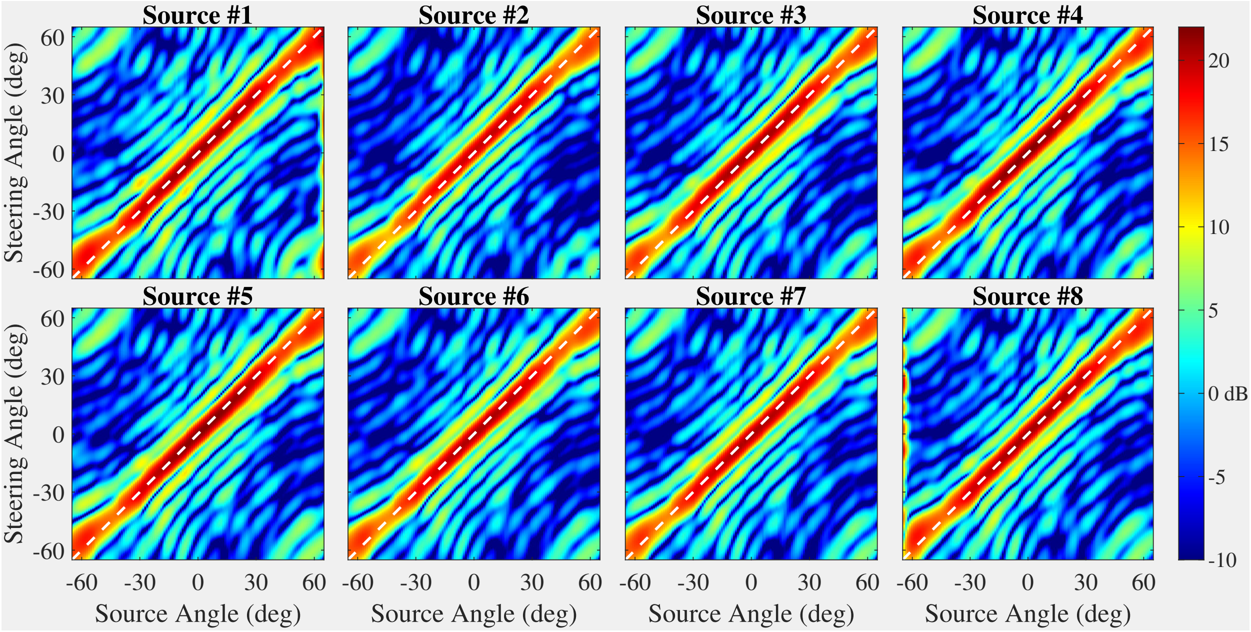


Fig. 38—A comparison of the apparent far-field angle of each source (via beamforming with ideal steering vectors) with the actual physical angle as the array was swept. The white, dashed line through the origin has unity slope.

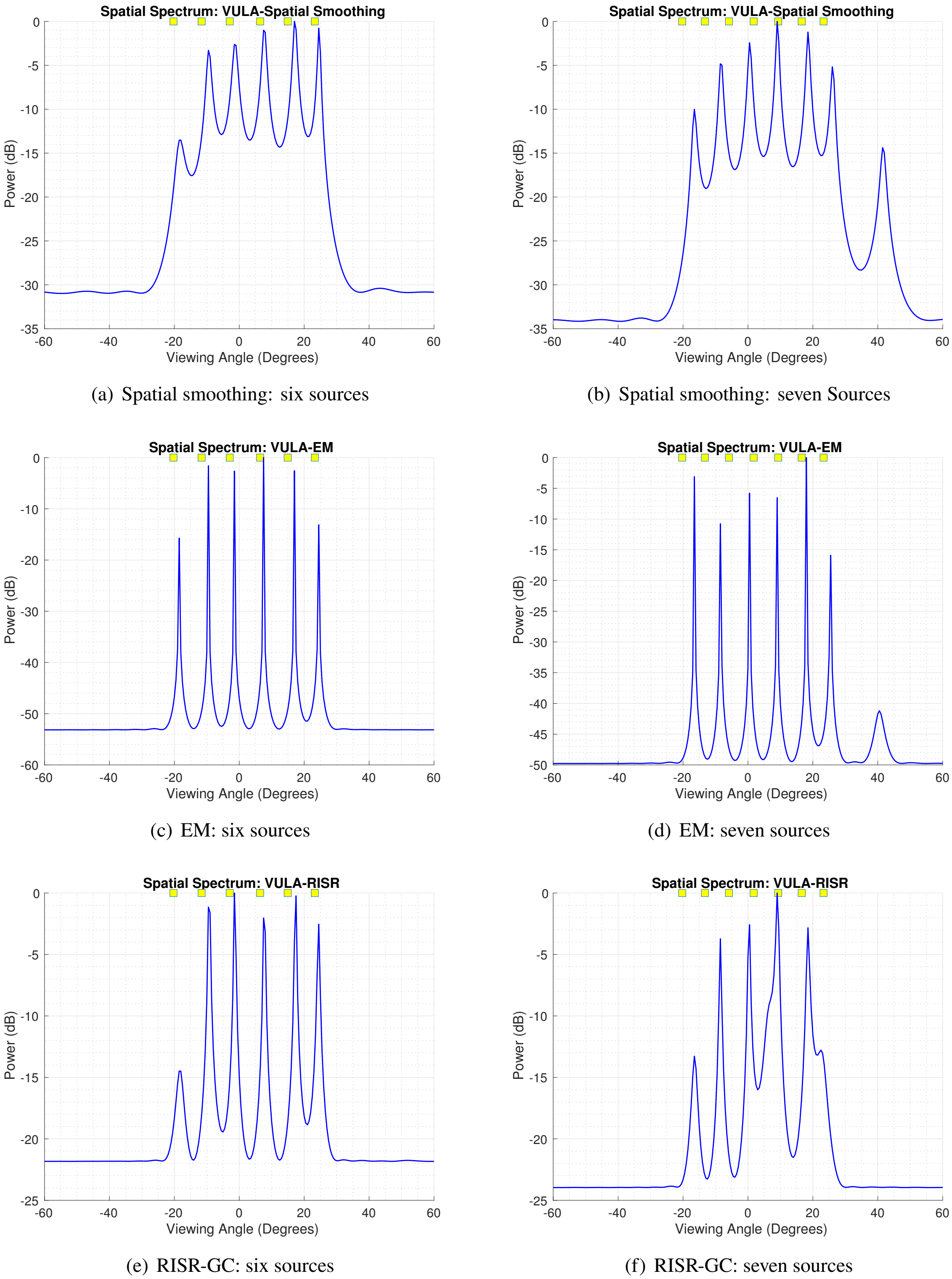
5.6 NAP Results

Here, we show some samples of the data we used to evaluate the efficacy of NAP with the measured data. Like the simulations presented in the previous chapter, we considered the DOA estimation problem and the adaptive beamforming problem, respectively. All the spatial processing shown below was performed using only the six channels that corresponded to a nested-array configuration.

5.6.1 DOA Estimation

For these experiments, we set the relevant transmitters to emit a Gaussian noise waveform. In the cases shown here, the noise sources were emitted with approximately uniform power levels. We used $K = 5,000$ (fast time) snapshots for all the cases shown here.

Figure 39 shows results for the case of $M = 6$ transmitters located at DOA angles $[23.3°, 14.9°, 6.4°, -2.9°, -11.5°, -20.2°]^T$, and for the case of $M = 7$ transmitters with DOA angles $[23.3°, 16.5°, 9.3°, 1.7°, -5.8°, -13.2°, -20.2°]^T$ all relative to boresight. For the $M = 6$, case, we do note that all three methods (spatial smoothing, EM, and RISR-GC) experienced some angular drift (even more so than observed in simulation) relative to their true angular locations. We also note that for runs where $M < N$, this was not the case for any of the three methods. We can see that EM exhibited the sharpest peaks relative to the other two methods. For the $M = 7$ case, both spatial smoothing and EM exhibit a false source in roughly the same angular location. RISR-GC did not, but did fail to identify all seven peaks. The inability of the methods to identify all seven sources was not specific to the NAP approach, as we note that this was the case even DOA estimation was performed with all 12 channels using conventional methods. This could be attributed to the angular spacing of the sources, which could not be increased due to the limited field of view that the chamber could accommodate. We also considered tests to compare the spatial resolution of the techniques. For this,



(a) Spatial smoothing: six sources

(b) Spatial smoothing: seven Sources

(c) EM: six sources

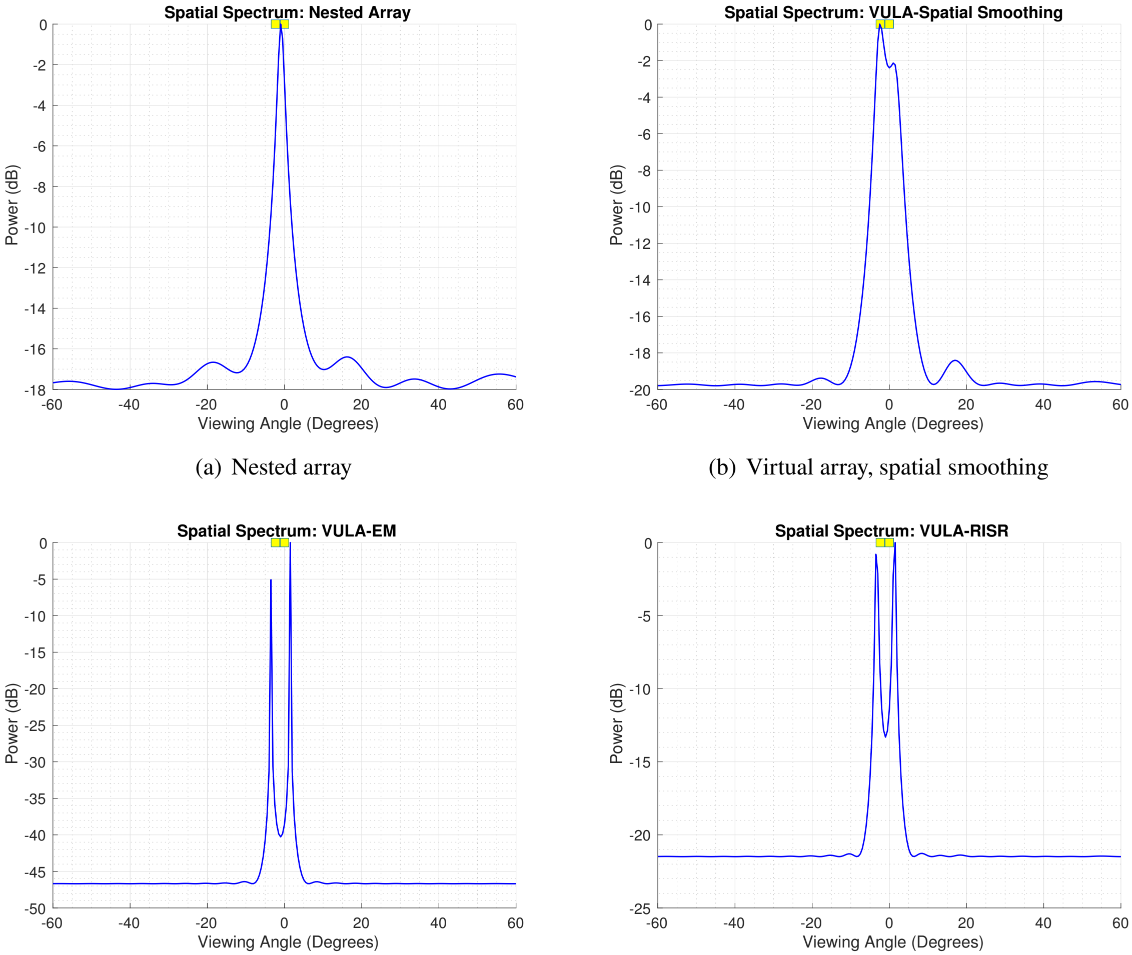
(d) EM: seven sources

(e) RISR-GC: six sources

(f) RISR-GC: seven sources

Fig. 39—Comparison of the different methods for coarray-based DOA estimation showing spatial smoothing, EM, and RISR for the cases of six and seven Gaussian noise sources

we used two noise sources emitting the same power and placed within close angular proximity, roughly 2° of separation. Figure 40 shows the results for linear MUSIC (conventional MUSIC using the linear model), SS, EM, and RISR. We can clearly see that EM and RISR are able to leverage the longer virtual array as opposed to linear MUSIC and spatial smoothing MUSIC.



(a) Nested array

(b) Virtual array, spatial smoothing

(c) Virtual array, EM

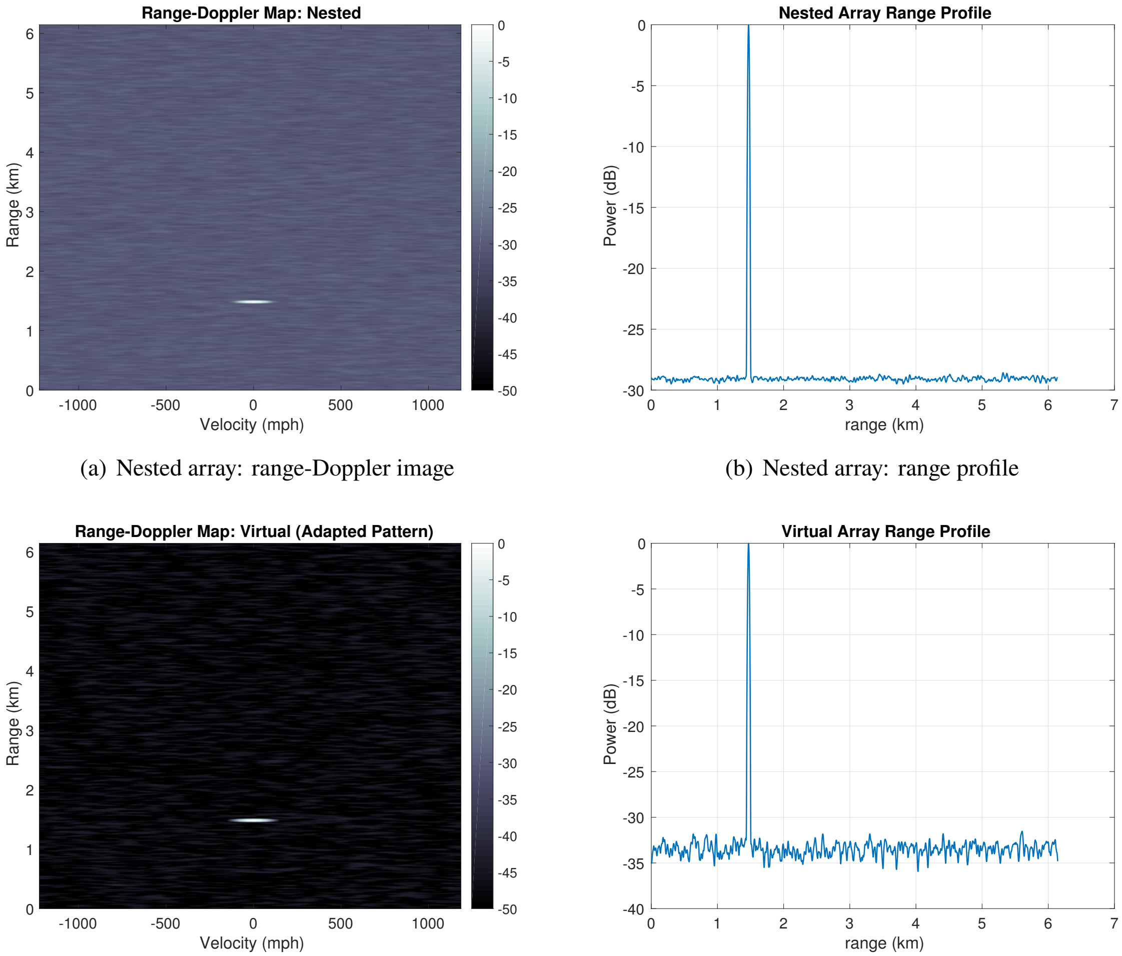
(d) Virtual array, RISR-GC

Fig. 40—DOA estimation comparison of resolution performance. We demonstrate empirically that iterative methods such as RISR and EM applied to the nonlinear model can take advantage of the seemingly longer aperture.

5.6.2 Adaptive Beamforming

In similar fashion to our simulation-based experiments, we sought to assess whether NAP could improve SINR when more noise jammers than elements are used to flood the array. For these tests, we set one of the transmitting horns to radiate a coherent waveform corresponding to a fixed range-Doppler cell while setting a variable number of other horns to radiate Gaussian noise. In all test cases the target horn was fixed at −20.2° relative to boresight and 2,000 pulses of a 40 μs 10 MHz LFM chirp was radiated. This coherent source was set to emulate a target at a range of about 1.47 km and zero velocity. Like before, we consider beamforming under the assumption of clairvoyant knowledge of the DOAs of all incoming sources using the minimum-norm beamformer. Also as before, we split the overall dwell time of 2,000 pulses into roughly 41 CPIs consisting of 48 pulses, where each CPI was used to coherently process range-Doppler maps for each of the six receive channels.

Figure 41 shows an example comparison between linear processing and NAP for the case of seven Gaussian noise jammers spaced at $[23.3°, 17.6°, 11.5°, 5.2°, -1.76°, -8.16°, -14.3°]$. In this case, the chirp was transmitted at roughly the same power as each of the noise sources, and as a result, overall SINR is favorable even for linear processing and can be attributed mainly to coherent gain. Nevertheless, we still observe that NAP had roughly a 4.3 dB gain over the linear processing approach. In the next experiment, we



(a) Nested array: range-Doppler image

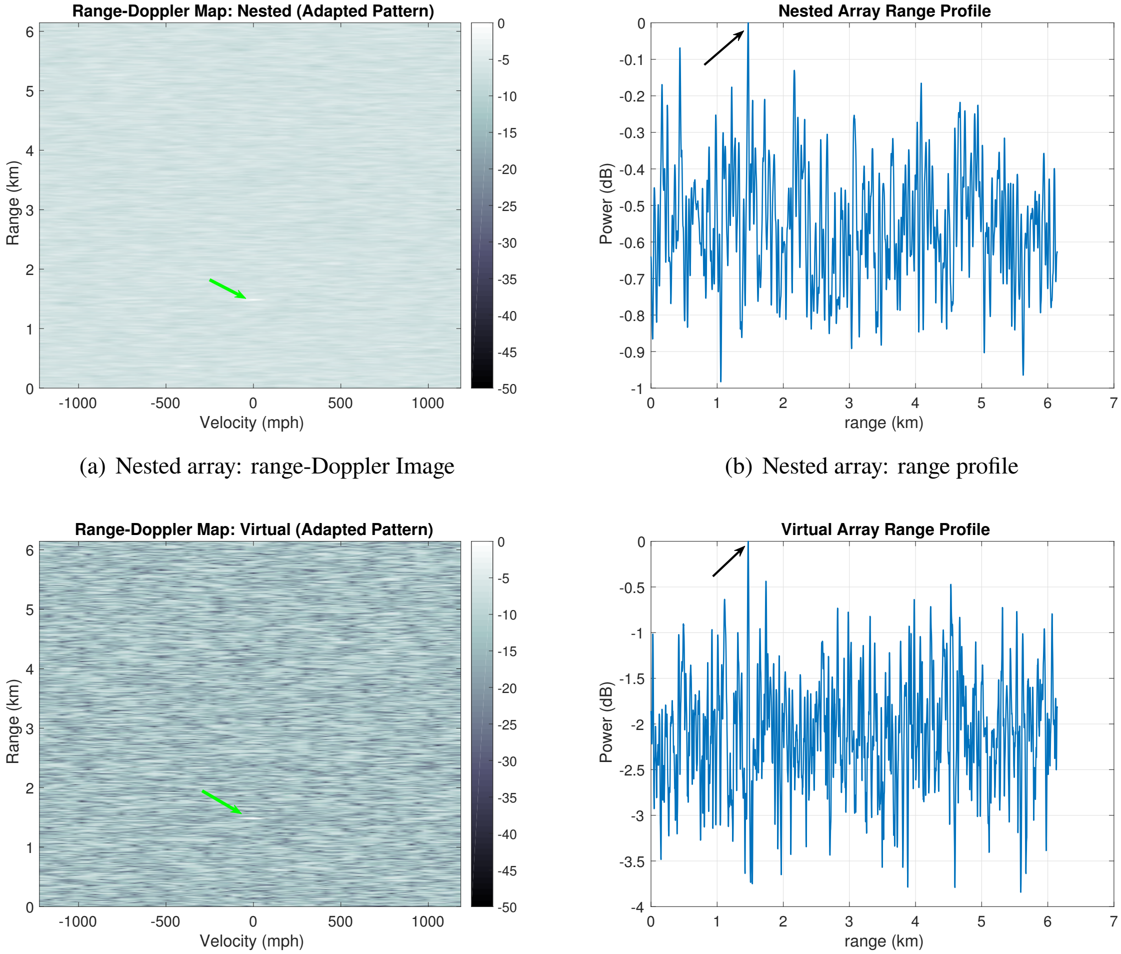
(b) Nested array: range profile

(c) Virtual array: range-Doppler image

(d) Virtual array: range profile

Fig. 41—Adaptive beamforming comparison between nonlinear and linear models. In these experiments, the target was roughly the same power as each of the jammers. NAP had a clear gain over linear processing in this case, although most of the SINR gain can be attributed to coherent gain.

wanted to see if NAP could provide SINR gain for a case in which coherent gain did not result in favorable SINR. In this case, we had five noise jammers with DOA angles $[23.3°, 11.5°, 5.2°, -1.76°, -8.16°]^T$, except now the coherent source was roughly 50 dB down in amplitude relative to each of the jammer sources. Figure 42 shows the results, where we can observe about a 1.5 dB SINR gain for NAP over the linear model.



(a) Nested array: range-Doppler Image

(b) Nested array: range profile

(c) Virtual array: range-Doppler Image

(d) Virtual array: range profile

Fig. 42—Another comparison between linear and nonlinear adaptive beamforming. Here, the target was roughly 50 dB gain down relative to each of the jammers. We can observe that NAP achieves about a 1.5 dB gain advantage over conventional processing.

6. FINAL SUMMARY AND CONCLUSIONS

In summary, the purpose of this study was to investigate a NAP framework that is based on treating the difference coarray of a sparse array as a virtual ULA and to assess its applicability in radar. Our study was supported by a number numerical simulations to assess array-processing algorithms for DOA estimation and beamforming. Furthermore, an experimental testbed was designed and built for the purpose of evaluating array-processing algorithms within an emulated radar framework. We also note that testbed, as well as the various data that was captured, is not specific to NAP, and as such could be used for future studies concerning full/sparse array research.

The following conclusions, which were, for the most part, consistent with numerical simulations and experiments, summarize various aspects of our study:

- A CRLB analysis was conducted under the usual Gaussian assumptions and showed that no new information is gained through the NAP perspective. As such, earlier conclusions regarding the maximum number of DOF that can be exploited for DOA estimation that have been known in the literature for some time now still hold for NAP: The maximum number of sources that can be estimated is approximately half of the difference coarray. Furthermore, as prior studies [6] have stated, time sample support becomes a critical issue for sparse augmentable arrays when the number of sources exceeds the number of physical elements. This is still the case with the NAP framework.
- We offered a concrete characterization of the virtual covariance matrix for NAP using a Gaussian/Wishart analysis. This was important for being to able to explicitly characterize the virtual SNR, as well as to show differences between the virtual covariance matrix structure and physical data covariance matrix structure.
- We presented the application of a number of alternatives to spatial smoothing for DOA estimation with the virtual array. Specifically, iterative algorithms such as RISR, RISR-GC, and EM do not rely on decimating the virtual array in half, as is the case with a spatial smoothing approach. Although this does not imply the ability to exploit the full DOF of the virtual array for DOA estimation, it may have some benefits toward the resolution performance of the estimators, especially when viewed as covariance matrix estimators for the MUSIC algorithm. This seemed to be the case in the DOA estimation simulations and experiments that were conducted.
- We showed that adaptive beamforming with the virtual array was not a practical approach mainly due to the need for limiting the long dwell times needed by NAP to reduce cross-term energy. Although our simulations and measured experiments show that it is possible to achieve SINR gains with NAP, the amount of time needed to dwell on the scene was not within reasonable time frames.

From these key summarized points, the overall conclusion is that NAP could have some benefit primarily as a passive receive system for which spectrum estimation is the primary objective rather than a complete replacement of array processing for monostatic radar systems.

REFERENCES

1. M. Ryle, A. Hewish, and J. Shakeshaft, "The synthesis of large radio telescopes by the use of radio interferometers," *IRE Transactions on Antennas and Propagation* **7**(5), 120–124 (Dec. 1959).

2. A. Moffet, "Minimum-redundancy linear arrays," *IEEE Transactions on Antennas and Propagation* **16**(2), 172–175 (Mar. 1968).

3. R. Hoctor and S. Kassam, "The unifying role of the coarray in aperture synthesis for coherent and incoherent imaging," *Proceedings of the IEEE* **78**(4), 735–752 (Apr. 1990).

4. S. U. Pillai, Y. Bar-Ness, and F. Haber, "A new approach to array geometry for improved spatial spectrum estimation," *Proceedings of the IEEE* **73**(10), 1522–1524 (Oct 1985), ISSN 0018-9219, doi:10.1109/PROC.1985.13324.

5. S. Pillai and F. Haber, "Statistical analysis of a high resolution spatial spectrum estimator utilizing an augmented covariance matrix," *IEEE Transactions on Acoustics, Speech, and Signal Processing* **35**(11), 1517–1523 (Nov. 1987).

6. Y. I. Abramovich, D. A. Gray, A. Y. Gorokhov, and N. K. Spencer, "Positive-definite toeplitz completion in DOA estimation for nonuniform linear antenna arrays. i. fully augmentable arrays," *IEEE Transactions on Signal Processing* **46**(9), 2458–2471 (Sep 1998), ISSN 1053-587X, doi:10.1109/78.709534.

7. Y. I. Abramovich, N. K. Spencer, and A. Y. Gorokhov, "Positive-definite toeplitz completion in doa estimation for nonuniform linear antenna arrays. ii. partially augmentable arrays," *IEEE Transactions on Signal Processing* **47**(6), 1502–1521 (Jun 1999), ISSN 1053-587X, doi:10.1109/78.765119.

8. M. C. Dogan and J. M. Mendel, "Applications of cumulants to array processing .i. aperture extension and array calibration," *IEEE Transactions on Signal Processing* **43**(5), 1200–1216 (May 1995), ISSN 1941-0476, doi:10.1109/78.382404.

9. P. Chevalier and A. Ferreol, "On the virtual array concept for the fourth-order direction finding problem," *IEEE Transactions on Signal Processing* **47**(9), 2592–2595 (Sep. 1999), ISSN 1941-0476, doi:10.1109/78.782217.

10. P. Chevalier, L. Albera, A. Ferreol, and P. Comon, "On the virtual array concept for higher order array processing," *IEEE Transactions on Signal Processing* **53**(4), 1254–1271 (April 2005), ISSN 1941-0476, doi:10.1109/TSP.2005.843703.

11. P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Transactions on Signal Processing* **58**(8), 4167–4181 (Aug 2010), ISSN 1053-587X, doi:10.1109/TSP.2010.2049264.

12. P. Pal and P. P. Vaidyanathan, "Coprime sampling and the music algorithm," in *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pp. 289–294 (Jan 2011), doi:10.1109/DSP-SPE.2011.5739227.

13. Y. D. Zhang, M. G. Amin, and B. Himed, "Sparsity-based doa estimation using co-prime arrays," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3967–3971 (May 2013), doi:10.1109/ICASSP.2013.6638403.

14. I. Reed, J. Mallett, and L. Brennan, "Rapid Convergence Rate in Adaptive Arrays," *IEEE Transactions on Aerospace and Electronic Systems* **AES-10**(6), 853–863 (Nov. 1974).

15. W. K. Ma, T. H. Hsieh, and C. Y. Chi, "Doa estimation of quasi-stationary signals via khatri-rao subspace," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2165–2168 (April 2009), doi:10.1109/ICASSP.2009.4960046.

16. S. Fortunati, R. Grasso, F. Gini, M. S. Greco, and K. LePage, "Single-snapshot doa estimation by using compressed sensing," *EURASIP Journal on Advances in Signal Processing* **2014**(1), 120 (2014), doi:10.1186/1687-6180-2014-120, URL `https://doi.org/10.1186/1687-6180-2014-120`.

17. S. Blunt, T. Chan, and K. Gerlach, "Robust DOA estimation: The reiterative superresolution (RISR) algorithm," *IEEE Transactions on Aerospace and Electronic Systems* **47**(1), 332–346 (Jan. 2011), ISSN 0018-9251.

18. L. C. Godara and D. A. Gray, "An algorithm for adaptive augmented array beamforming," *The Journal of the Acoustical Society of America* **83**(6), 2261–2265 (June 1988).

19. C. Zhou, Y. Gu, S. He, and Z. Shi, "A robust and efficient algorithm for coprime array adaptive beamforming," *IEEE Transactions on Vehicular Technology* **67**(2), 1099–1112 (Feb 2018), doi:10.1109/TVT.2017.2704610.

20. P. Vouras, "Fully adaptive space-time processing on nested arrays," in *2015 IEEE Radar Conference (RadarCon)*, pp. 0858–0863 (May 2015), doi:10.1109/RADAR.2015.7131115.

21. I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Transactions on Aerospace and Electronic Systems* **AES-10**(6), 853–863 (Nov 1974), ISSN 0018-9251, doi:10.1109/TAES.1974.307893.

22. Y. C. Eldar and A. Nehorai, *Mean-Squared Error Beamforming for Signal Estimation: A Competitive Approach*, chapter 5, pp. 259–298 (John Wiley & Sons, Ltd, 2005), ISBN 9780471733485, doi:10.1002/0471733482.ch5, URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/0471733482.ch5`.

23. A. Paulraj, R. Roy, and T. Kailath, "A subspace rotation approach to signal parameter estimation," *Proceedings of the IEEE* **74**(7), 1044–1046 (1986).

24. A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pp. 336–339 (April 1983), doi:10.1109/ICASSP.1983.1172124.

25. S. D. Blunt, T. Chan, and K. Gerlach, "Robust doa estimation: The reiterative superresolution (risr) algorithm," *IEEE Transactions on Aerospace and Electronic Systems* **47**(1), 332–346 (2011).

26. T. Higgins, S. D. Blunt, and K. Gerlach, "Gain-constrained adaptive pulse compression via an MVDR framework," in *Proc. IEEE Radar Conference* (May 2009), doi:10.1109/RADAR.2009.4977011.

27. T. A. Barton and D. R. Fuhrmann, "Covariance estimation for multidimensional data using the em algorithm," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 203–207 vol.1 (Nov 1993), doi:10.1109/ACSSC.1993.342500.

28. D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing* **53**(8), 3010–3022 (2005).

29. G. F. Edelmann and C. F. Gaumond, "Beamforming using compressive sensing," *The Journal of the Acoustical Society of America* **130**(4), EL232–EL237 (2011), doi:10.1121/1.3632046, URL `https://doi.org/10.1121/1.3632046`.

30. Z. Ben-Haim, Y. C. Eldar, and M. Elad, "Coherence-based performance guarantees for estimating a sparse vector under random noise," *IEEE Transactions on Signal Processing* **58**(10), 5030–5043 (2010).

31. Y. Chi, A. Pezeshki, L. Scharf, and R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3930–3933 (2010).

32. G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Transactions on Information Theory* **59**(11), 7465–7490 (2013).

33. E. J. Candès and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," *Communications on Pure and Applied Mathematics* **67**(6), 906–956 (2014), doi:10.1002/cpa.21455, URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21455`.

34. H. Van-Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory* (New York: Wiley Intersci., 2002).

35. E. Vertatschitsch and S. Haykin, "Nonredundant arrays," *Proceedings of the IEEE* **74**(1), 217–217 (1986).

36. P. P. Vaidyanathan and P. Pal, "Sparse sensing with coprime arrays," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pp. 1405–1409 (2010).

37. C. Liu and P. P. Vaidyanathan, "Super nested arrays: Linear sparse arrays with reduced mutual coupling—part i: Fundamentals," *IEEE Transactions on Signal Processing* **64**(15), 3997–4012 (2016).

38. R. L. Dykstra, "An algorithm for restricted least squares regression," *Journal of the American Statistical Association* **78**(384), 837–842 (1983), ISSN 01621459, URL `http://www.jstor.org/stable/2288193`.

39. P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing* **38**(10), 1783–1795 (1990).

40. P. Stoica, E. G. Larsson, and A. B. Gershman, "The stochastic crb for array processing: a textbook derivation," *IEEE Signal Processing Letters* **8**(5), 148–150 (2001).

41. C.-L. Liu and P. Vaidyanathan, "Cramer-rao bounds for coprime and other sparse arrays, which find more sources than sensors," *Digital Signal Processing* **61**, 43 – 61 (2017), ISSN 1051-2004, doi:https://doi.org/10.1016/j.dsp.2016.04.011, URL `http://www.sciencedirect.com/science/article/pii/S1051200416300264`, Special Issue on Coprime Sampling and Arrays.

42. D. Maiwald and D. Kraus, "On moments of complex wishart and complex inverse wishart distributed matrices," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pp. 3817–3820 vol.5 (1997), doi:10.1109/ICASSP.1997.604712.

43. Q. Zhang, "Probability of resolution of the music algorithm," *IEEE Transactions on Signal Processing* **43**(4), 978–987 (1995), doi:10.1109/78.376849.

44. K. S. Miller, "On the inverse of the sum of matrices," *Mathematics Magazine* **54**(2), 67–72 (1981), ISSN 0025570X, 19300980, URL `http://www.jstor.org/stable/2690437`.

45. M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, *Introduction to compressed sensing*, p. 1–64 (Cambridge University Press, 2012), doi:10.1017/CBO9780511794308.002.

46. D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization," *Proceedings of the National Academy of Sciences of the United States of America* **100**(5), 2197–2202 (2003), ISSN 00278424, URL `http://www.jstor.org/stable/3139504`.

47. J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Transactions on Signal Processing* **54**(12), 4634–4643 (2006).

Appendix A

MATRIX NOTATION AND CALCULUS

A.1 Matrix Algebra

Let $\mathbb{C}^{m\times n}$ denote the space of $m \times n$ matrices taken over the field of complex numbers. Generally, we will use capital bold face letters to denote an $m \times n$ matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ and $[\mathbf{A}]_{ij}$ to denote the (i, j) element of the matrix $\mathbf{A}$. The notation $\mathbf{A}^T$ will be used to denote the transpose operator applied to a matrix $\mathbf{A}$. Similarly, we have $\mathbf{A}^*$ denoting the complex conjugate operator, and $\mathbf{A}^H$ denoting the conjugate transpose operator. We will use lowercase, boldfaced letters to denote vectors in some n dimensional space over the field of complex numbers, e.g., , $\mathbf{a} \in \mathbb{C}^n$. Oftentimes, we need to sum over elements of a matrix or a vector or define certain quantities elementwise. When it is advantageous to do so for these cases, we will adopt the Einstein summation convention with the hope that this will result in a more clean presentation of some of the intermediate calculations. In this convention, we will use the notation a_{ij} to denote $[\mathbf{A}]_{ij}$, where repeated indices are understood to mean a summation is performed over those indices. For example, consider the definition of matrix multiplication; Let $\mathbf{A} \in \mathbb{C}^{m\times n}$ and $\mathbf{B} \in \mathbb{C}^{n\times p}$, then the (i, j) element of the matrix $\mathbf{C} = \mathbf{AB}$ is given as

$$c_{ij} = a_{ik} b_{kj}. \tag{A1}$$

Another key operator with respect to matrix quantities is the trace operator,

Definition 5. *Let* $\mathbf{A} \in \mathbb{C}^{n\times n}$*. We define the trace of a matrix, denoted as* $tr\left[\mathbf{A}\right]$*, as*

$$tr\left[\mathbf{A}\right] = a_{ii}. \tag{A2}$$

In fact, matrix multiplication is commutative under action of the trace operator,

Proposition 6. *Let* $\mathbf{AB} \in \mathbb{C}^{n\times n}$*, then*

$$tr\left[\mathbf{AB}\right] = tr\left[\mathbf{BA}\right]. \tag{A3}$$

Proof. This is straightforward from the definition of a matrix product and the trace operator:

$$\begin{aligned} \mathrm{tr}\left[\mathbf{AB}\right] &= a_{ij} b_{ji} \\ &= b_{ji} a_{ij} \\ &= \mathrm{tr}\left[\mathbf{BA}\right]. \end{aligned}$$

We will use the notation $\mathbf{I}_n$ to refer to the identity matrix in the space $\mathbb{C}^{n\times n}$.

Definition 6. *A matrix* $\mathbf{A} \in \mathbb{C}^{n\times n}$ *is said to be invertible if there exists some matrix* $\mathbf{A}^{-1} \in \mathbb{C}^{n\times n}$ *such that*

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n. \tag{A4}$$

Furthermore, if such a matrix exists, it is unique in $\mathbb{C}^{n\times n}$.

A square matrix whose inverse exists is often referred to as nonsingular. Conversely, a non-invertible square matrix is said to be nonsingular. A fundamental result that frequently arises in array processing is the Sherman-Morrison-Woodbury identity:

Lemma 3. *Let* $\mathbf{A}$, $\mathbf{C}$, *and* $\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B}$ *be nonsingular square matrices. Then the matrix* $\mathbf{A} + \mathbf{BCD}$ *is invertible and*

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}. \tag{A5}$$

Proof. The result follows by direct multiplication of $(\mathbf{A}+\mathbf{BCD})$ with $\mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}$ to obtain the identity matrix.

A special case of Lemma 3 (see [44]) can be stated as follows:

Lemma 4. *If* $\mathbf{A}$ *and* $\mathbf{A} + \mathbf{B}$ *are invertible, and* $\mathbf{B}$ *has rank 1, then let* $g = tr\left[\mathbf{B}\mathbf{A}^{-1}\right]$. *Then* $g \neq 1$ *and*

$$(\mathbf{A} + \mathbf{B})^{-1} = \mathbf{A}^{-1} - \frac{1}{1+g}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}. \tag{A6}$$

Proof. Since $\mathbf{B}$ is a rank 1 matrix, we can represent it as the outer product of two column vectors $\mathbf{B} = \mathbf{u}\mathbf{v}^H$. Thus, by Lemma 3,

$$\begin{aligned}
(\mathbf{A} + \mathbf{B})^{-1} &= \left(\mathbf{A} + \mathbf{u}\mathbf{v}^H\right)^{-1} \\
&= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{u}\left[1 + \mathbf{v}^H\mathbf{A}^{-1}\mathbf{u}\right]^{-1}\mathbf{v}^H\mathbf{A}^{-1} \\
&= \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^H\mathbf{A}^{-1}}{1 + \mathbf{v}^H\mathbf{A}^{-1}\mathbf{u}} \\
&= \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}}{1 + \mathbf{v}^H\mathbf{A}^{-1}\mathbf{u}}.
\end{aligned}$$

A direct calculation of the scalar factor in the denominator shows that

$$\begin{aligned}
\mathbf{v}^H\mathbf{A}^{-1}\mathbf{u} &= v_j^*\left[\mathbf{A}^{-1}\right]_{jk} u_k && \text{(A7)} \\
&= \left[\mathbf{A}^{-1}\right]_{jk} u_k v_j^* && \text{(A8)} \\
&= \left[\mathbf{A}^{-1}\right]_{jk} [\mathbf{B}]_{kj} && \text{(A9)} \\
&= \left[\mathbf{A}^{-1}\mathbf{B}\right]_{jj} = \mathrm{tr}\left[\mathbf{A}^{-1}\mathbf{B}\right]. && \text{(A10)}
\end{aligned}$$

The special case of positive definite Hermitian matrices is of interest because they are always nonsingular.

Definition 7. *A matrix* $\mathbf{A}$ *is said to be Hermitian if* $\mathbf{A}^H = \mathbf{A}$.

A Hermitian matrix $\mathbf{A}$ is said to be positive-definite if and only if $\mathbf{x}^H\mathbf{A}\mathbf{x} > 0$ for all nontrivial vectors $\mathbf{x}$. An important characterization of such matrices is provided by the Schur complement:

Proposition 7. *For any Hermitian matrix* $\mathbf{F}$ *of the form*

$$\mathbf{F} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^H & \mathbf{D} \end{bmatrix} \tag{A11}$$

where $\mathbf{A}$ *and* $\mathbf{D}$ *are also Hermitian matrices, we have* $\mathbf{F}$ *is nonsingular if and only if* $\mathbf{D}$ *and the Schur compliment of* $\mathbf{D}$*,* $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^H$ *are both nonsingular.*

Proof. First note that we have the following factorization of $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} \mathbf{I} & \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^H & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^H$$

Thus, taking the determinant of $\mathbf{F}$, we see that

$$|\mathbf{F}| = |\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^H||\mathbf{D}|;$$

if both matrices $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^H$ and $\mathbf{D}$ are non-singular, it then follows that $\mathbf{F}$ is non-singular, i.e., , $|\mathbf{F}| \neq 0$.

Recall that an n-dimensional Euclidean space is a Hilbert space with inner product

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^H\mathbf{b}, \ \forall \mathbf{a}, \mathbf{b} \in \mathbb{C}^n. \tag{A12}$$

The inner product can be used to induce the Euclidean norm

$$\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^H\mathbf{a}}, \ \forall \mathbf{a} \in \mathbf{C}^n. \tag{A13}$$

In fact, the space $\mathbb{C}^{m\times n}$ can be viewed as a Hilbert space through the inner product

$$\langle \mathbf{A}, \mathbf{B} \rangle = \operatorname{tr}\left[\mathbf{A}^H\mathbf{B}\right], \ \forall \mathbf{A}, \mathbf{B} \in \mathbb{C}^{m\times n}, \tag{A14}$$

with induced norm

$$\|\mathbf{A}\|_F = \sqrt{\operatorname{tr}\left[\mathbf{A}^H\mathbf{A}\right]}, \ \forall \mathbf{a} \in \mathbf{C}^{m\times n}. \tag{A15}$$

For any arbitrary matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ we can associate with it a unique element in $\mathbb{C}^{mn}$ simply by unraveling the elements in the matrix into a vector:

$$\text{VEC}\left(\mathbf{A}\right) : \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{2n} & \dots & a_{nn} \end{bmatrix} \mapsto \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{nn} \end{bmatrix} \tag{A16}$$

Note that here we have chosen to use column-major indexing in the unraveling. One equivalently may use a row-major indexing here as long as consistency between element mappings is maintained. It is not hard to see that VEC $(\cdot)$ operator is a linear operator, and in fact defines an *isomorphic* relationship between the Hilbert spaces $\mathbb{C}^{m \times n}$ and $\mathbb{C}^{mn}$, i.e., the Euclidean distances between elements of these respective spaces are preserved under the VEC $(\cdot)$ operator. As such, one can define an inverse such that $\mathbf{a} \mapsto \mathbf{A}$ as

$$\text{MAT}\left[\mathbf{a}\right] : \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{mn} \end{bmatrix} \mapsto \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{2n} & \dots & a_{mn} \end{bmatrix}, \tag{A17}$$

with $\|\mathbf{a}\|_2 = \|\text{MAT}\left[\mathbf{a}\right]\|_F$. This isomorphic mapping between the two representations can be used to simplify the derivation of new relationships between various quantities. For example, one useful property relates the trace operator to inner products in the vector space $\mathbb{C}^{mn}$:

Proposition 8. *For any $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$, we have*

$$tr\left[\mathbf{AB}\right] = VEC\left(\mathbf{A}^H\right)^H VEC\left(\mathbf{B}\right). \tag{A18}$$

Proof. This follows from a straightforward componentwise calculation of matrix multiplication and the trace operator

$$\begin{aligned} \text{tr}\left[\mathbf{AB}\right] &= a_{ik}b_{ki} && \text{(A19)} \\ &= a_{11}b_{11} + a_{12}b_{21} + \dots a_{nn}b_{nn} && \text{(A20)} \\ &= a_{11}^T b_{11} + a_{21}^T b_{21} + \dots + a_{nn}^T b_{nn} && \text{(A21)} \\ &= \text{VEC}\left(\mathbf{A}^T\right)^T \text{VEC}\left(\mathbf{B}\right) && \text{(A22)} \\ &= \text{VEC}\left(\mathbf{A}^H\right)^H \text{VEC}\left(\mathbf{B}\right). && \text{(A23)} \end{aligned}$$

Another important relationship can be derived as follows:

Definition 8. *Let* $\mathbf{A} \in \mathbb{C}^{m \times n}$ *and* $\mathbf{B} \in \mathbb{C}^{p \times q}$*. The* $mp \times nq$ *matrix*

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{2n}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{bmatrix} \tag{A24}$$

is defined to be the Kronecker product of **A** *and* **B***.*

Proposition 9. *Let* **A** *and* **B** *be* $m \times n$*,* $n \times p$*, and* $p \times q$ *sized matrices, respectively. Then*

$$VEC\left(\mathbf{ABC}\right) = \left(\mathbf{C}^T \otimes \mathbf{A}\right) VEC\left(\mathbf{B}\right). \tag{A25}$$

Proof. Put $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_n]$ (assuming a size of $m \times n$) and $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_m]$. By definition of matrix products, we have

$$[\mathbf{ABC}]_{ij} = a_{ik} b_{kl} c_{lj}, \tag{A26}$$

therefore, the j^{th} column of the above can be expressed as

$$\begin{aligned} \mathbf{A} \sum_{k=1}^{m} \mathbf{b}_k c_{kj} &= [c_{1j}\mathbf{A}, c_{2j}\mathbf{A}, \dots, c_{mj}\mathbf{A}] \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix} \\ &= \underbrace{\left([c_{1j}, c_{2j}, \dots, c_{mj}] \otimes \mathbf{A}\right)}_{\mathbf{c}_j^T \otimes \mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}}_{\text{VEC}(B)}. \end{aligned}$$

By stacking all the remaining columns, the result follows.

A.2 Calculus

We first consider some important properties concerning differentiation of matrix-valued quantities with respect to a real-valued scalar θ. For such matrices that depend implicitly on θ, we have $\mathbf{A} = \mathbf{A}(\theta)$, and the derivative of **A** with respect to θ is then taken componentwise:

$$\frac{\partial \mathbf{A}}{\partial \theta} = \begin{bmatrix} \frac{\partial a_{11}}{\partial \theta} & \frac{\partial a_{12}}{\partial \theta} & \dots & \frac{\partial a_{1n}}{\partial \theta} \\ \frac{\partial a_{21}}{\partial \theta} & \frac{\partial a_{22}}{\partial \theta} & \dots & \frac{\partial a_{2n}}{\partial \theta} \\ \vdots & \vdots & & \vdots \\ \frac{\partial a_{n1}}{\partial \theta} & \frac{\partial a_{n2}}{\partial \theta} & \dots & \frac{\partial a_{nn}}{\partial \theta} \end{bmatrix}. \tag{A27}$$

Based on this definition, we now derive a few important quantities that we are likely to encounter when deriving Cramer-Rao bounds.

Proposition 10 (Chain Rule)**.** *Let* $\mathbf{A}(\theta) \in \mathbb{R}^{m\times n}$ *and* $\mathbf{B}(\theta) \in \mathbb{R}^{n\times p}$*. Then the derivative of their product with respect to* θ *is given by the chain rule:*

$$\frac{\partial \mathbf{AB}}{\partial \theta} = \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{B} + \mathbf{A}\frac{\partial \mathbf{B}}{\partial \theta}. \tag{A28}$$

Proof. Put $\mathbf{C}(\theta) = \mathbf{AB}$. Consider a direct calculation of $\dfrac{\partial \mathbf{C}}{\partial \theta}$ from the individual elements $\frac{\partial c_{ij}}{\partial \theta}$:

$$\begin{aligned}\frac{\partial c_{ij}}{\partial \theta} &= \frac{\partial}{\partial \theta}\sum_{k=1}^{n} a_{ik}b_{kj} \\ &= \sum_{k=1}^{n}\frac{\partial a_{ik}b_{kj}}{\partial \theta}.\end{aligned}$$

We can now employ the conventional chain rule on the scalar derivatives inside the summation:

$$\begin{aligned}\frac{\partial a_{ik}b_{kj}}{\partial \theta} &= \sum_{k=1}^{n}\left[\frac{\partial a_{ik}}{\partial \theta}b_{kj} + a_{kj}\frac{\partial b_{ik}}{\partial \theta}\right] \\ &= \sum_{k=1}^{n}\frac{\partial a_{ik}}{\partial \theta}b_{kj} + \sum_{k=1}^{n}a_{kj}\frac{\partial b_{ik}}{\partial \theta},\end{aligned}$$

which we see corresponds to the chain rule products defined above.

Proposition 11. *Let* $\mathbf{A}$ *be a nonsingular* $n \times n$ *matrix whose elements are dependent on a scalar* θ*. Then*

$$\frac{\partial \mathbf{A}^{-1}}{\partial \theta} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{A}^{-1} \tag{A29}$$

Proof. Recall that by definition,

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}.$$

Now, applying the derivative with respect to θ to both sides of the equation above, we obtain

$$\frac{\partial \mathbf{A}^{-1}\mathbf{A}}{\partial \theta} = \mathbf{0}.$$

Using the chain rule (Proposition 10), we see that

$$\frac{\partial \mathbf{A}^{-1}}{\partial \theta}\mathbf{A} + \mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \theta} = \mathbf{0},$$

where we can now solve for $\frac{\partial \mathbf{A}^{-1}}{\partial \theta}$ to obtain Eq. (A29).

Finally, we recall another important matrix calculus identity that proves useful for Cramer-Rao analysis:

Proposition 12 (Jacobi's Formula)**.** *Let* $\mathbf{A}(\theta)$ *be differentiable with respect to the scalar* θ*. Then the derivative* $\frac{\partial|\mathbf{A}|}{\partial\theta}$ *exists and is given as*

$$\frac{\partial|\mathbf{A}|}{\partial\theta} = |\mathbf{A}| tr\left[\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial\theta}\right]. \tag{A30}$$

This page intentionally left blank

Appendix B

ADAPTIVE BEAMFORMING

When the output of an N-element array is experiencing interference from an unknown but finite number of interference sources, adaptive beamforming techniques such as MVDR and MMSE can be used to suppress their contribution in order to preserve signal integrity from the direction of interest. Again, the array output thus can be decomposed as

$$\mathbf{x}(t) = \mathbf{t}(t) + \mathbf{i}(t) + \mathbf{n}(t),$$

with $\mathbf{t}(t) = s(t)\mathbf{a}(\theta)$, where $s(t)$ is the source signal of interest. For this discussion, we will assume that it is a stochastic source with zero mean and variance $E\left[|s|^2\right] = \sigma^2$. Furthermore, we assume that $s(t)$ is uncorrelated with the interference and noise components:

$$E\left[s^*(t)\mathbf{i}\right] = \mathbf{0} \tag{B1}$$

$$E\left[s^*(t)\mathbf{n}\right] = \mathbf{0}, \tag{B2}$$

and that the covariance matrix corresponding to the interference and noise components is known (written here as $\mathbf{R}$ for brevity). The MVDR beamformer seeks weights $\mathbf{w}$ that the corresponding output $y(k) = \mathbf{w}^*\mathbf{x}(k)$ maximizes SINR:

$$\text{SINR} := \frac{E\left[|\mathbf{w}^H\mathbf{s}|^2\right]}{E\left[|\mathbf{w}^H(\mathbf{i}+\mathbf{n})|^2\right]} = \frac{\sigma_1^2|\mathbf{w}^H\mathbf{a}(\theta_1)|^2}{\mathbf{w}^H\mathbf{R}\mathbf{w}}. \tag{B3}$$

If we fix the numerator, then clearly, maximizing SINR can be accomplished by minimizing the denominator of the ratio. This plan of attack can be achieved formally through the constrained optimization program

$$\min_{\mathbf{w}} \mathbf{w}^H\mathbf{R}\mathbf{w} \text{ subject to } \mathbf{w}^H\mathbf{a}(\theta_1) = 1. \tag{B4}$$

The unique minima for the convex program in Eq. (B4) is derived analytically by calculating the gradient with respect to $\mathbf{w}^*$ and employing a Lagrange multiplier $\mu > 0$ to convert the gain constraint into a penalty term. This results in a linear equation in terms of $\mathbf{w}$ and μ:

$$\mathbf{R}\mathbf{w} = \mu\mathbf{a}(\theta). \tag{B5}$$

After inverting the covariance data incorporating the gain constrain in Eq. (B4), we find that the optimal weights are given as

$$\mathbf{w} = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{R}^{-1}\mathbf{a}(\theta)}.$$

We now derive weights corresponding to a MMSE beamformer. The MSE criteria is defined as

$$J(\mathbf{w}) = E\left[|y(t) - s(t)|^2\right]. \tag{B6}$$

To bring out the explicit dependence of $J(\mathbf{w})$ on the weights, we can expand Eq. (B6) as follows:

$$\begin{aligned} J(\mathbf{w}) &= E\left[yy^* - ys^* - sy^* + ss*\right] \\ &= E\left[\mathbf{w}^H\mathbf{x}\mathbf{x}^H\mathbf{w}\right] - 2E\left[ys^*\right] + \sigma^2. \end{aligned} \tag{B7}$$

For the first term in the expansion, we have

$$\begin{aligned} E\left[\mathbf{w}^H\mathbf{x}\mathbf{x}^H\mathbf{w}\right] &= \mathbf{w}^H E\left[(\mathbf{t}+\mathbf{i}+\mathbf{n})(\mathbf{t}+\mathbf{i}+\mathbf{n})^H\right]\mathbf{w} \\ &= \sigma^2\mathbf{w}^H\mathbf{a}\mathbf{a}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w}, \end{aligned} \tag{B8}$$

where we have used the fact that the target and interference/noise sources are uncorrelated. Considering the second term, we see that

$$\begin{aligned} E\left[ys^*\right] &= \mathbf{w}^H E\left[(\mathbf{t}+\mathbf{i}+\mathbf{n})s^*\right] \\ &= \sigma^2\mathbf{w}^H\mathbf{a}(\theta). \end{aligned} \tag{B9}$$

Putting Eqs. (B8) and (B9) back into Eq. (B7), we have

$$J(\mathbf{w}) = \sigma^2\mathbf{w}^H\mathbf{a}\mathbf{a}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w} - 2\sigma^2\mathbf{w}^H\mathbf{a}(\theta) + \sigma^2 \tag{B10}$$

$$= \mathbf{w}^H\mathbf{R}\mathbf{w} + \sigma^2|\mathbf{w}^H\mathbf{a} - 1|^2. \tag{B11}$$

Thus, we see that the MSE objective function has a similar form to the objective of a MVDR beamformer, where the first term is the denominator of the SINR expression and the right term is a gain control term, except we are now scaling the gain in the look direction θ by the source power σ^2. Now, if we compute the gradient of $J(\mathbf{w})$ with respect to $\mathbf{w}^*$ and set it to zero, we arrive at the equation

$$\left(\mathbf{R} + \sigma^2\mathbf{a}\mathbf{a}^H\right)\mathbf{w} = \sigma^2\mathbf{a}. \tag{B12}$$

Solving Eq. (B12) for $\mathbf{w}$ yields

$$\mathbf{w} = \sigma^2\left(\mathbf{R} + \sigma^2\mathbf{a}\mathbf{a}^H\right)^{-1}\mathbf{a}(\theta), \tag{B13}$$

which, at first glance, looks much different in form than Eq. (12). However, we can observe that $\left(\mathbf{R} + \sigma^2\mathbf{a}\mathbf{a}^H\right)^{-1}$ is the sum of an invertible matrix and a rank one matrix; hence, with the aid of Lemma 4 we obtain the expression

$$\mathbf{w} = \sigma^2\left(\mathbf{R}^{-1} + \frac{1}{1+g}\sigma^2\mathbf{R}^{-1}\mathbf{a}\mathbf{a}^H\mathbf{R}^{-1}\right)\mathbf{a}(\theta), \tag{B14}$$

where $g = \sigma^2 \mathrm{tr}\left[\mathbf{a}\mathbf{a}^H\mathbf{R}^{-1}\right]$. Now, observe that by definition of the trace operator and matrix multiplication,

$$\begin{aligned} \mathrm{tr}\left[\mathbf{a}\mathbf{a}^H\mathbf{R}^{-1}\right] &= \left[\mathbf{a}\mathbf{a}^H\right]_{ik}\left[\mathbf{R}^{-1}\right]_{ki} && \text{(B15)}\\ &= a_i a_k^* \left[\mathbf{R}^{-1}\right]_{ki} && \text{(B16)}\\ &= a_k^* \left[\mathbf{R}^{-1}\right]_{ki} a_i && \text{(B17)}\\ &= \mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}. && \text{(B18)} \end{aligned}$$

Incorporating this result into Eq. (B14) we see that

$$\begin{aligned} \mathbf{w} &= \sigma^2\left(\mathbf{R}^{-1} - \frac{1}{1+g}\sigma^2\mathbf{R}^{-1}\mathbf{a}\mathbf{a}^H\mathbf{R}^{-1}\right)\mathbf{a}(\theta) && \text{(B19)}\\ &= \sigma^2\left((1+g)\mathbf{R}^{-1}\mathbf{a} - \sigma^2\mathbf{R}^{-1}\mathbf{a}\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}\right)/(1+g) && \text{(B20)}\\ &= \sigma^2\left((1+g)\mathbf{R}^{-1}\mathbf{a} - g\mathbf{R}^{-1}\mathbf{a}\right)/(1+g) && \text{(B21)}\\ &= \frac{\sigma^2\mathbf{R}^{-1}\mathbf{a}}{1+\sigma^2\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}}. && \text{(B22)} \end{aligned}$$

Multiplying the numerator and the denominator of this result by the factor $\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}$, we see that

$$\begin{aligned} \mathbf{w} &= \frac{\sigma^2\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}}{1+\sigma^2\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}}\frac{\mathbf{R}^{-1}\mathbf{a}}{\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}} && \text{(B23)}\\ &= \beta(|s|)\mathbf{w}_{\mathrm{mvdr}}. && \text{(B24)} \end{aligned}$$

Thus, the MMSE beamformer is simply a rescaled version of the MVDR filter incorporating the a priori knowledge of $|s|^2$, and thus improving estimation error of the source in the mean-squared sense.

This page intentionally left blank

Appendix C

MAXIMUM LIKELIHOOD ESTIMATION AND CRAMER RAO LOWER BOUNDS

We consider the MLE with respect to the linear array model,

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k), \tag{C1}$$

where we wish to estimate the underlying parameters of the M sources given measurements from an N element array. Each array measurement $\mathbf{x}(k)$ yields a sample of the random vector $\boldsymbol{x}$ that we assume follows a circularly symmetric Gaussian distribution,

$$f_{\boldsymbol{x}}\left(\mathbf{x}|\mathbf{R}_x\right) = \pi^{-N}\left|\mathbf{R}_x^{-1}\right| \exp\left(-\mathbf{x}^H\mathbf{R}_x^{-1}\mathbf{x}\right), \tag{C2}$$

and is assumed to be independent of the other measurements. Thus, the MMV data $\mathbf{X}$, itself, is a sample of the random variable $\boldsymbol{X}$ whose distribution is given as the product of K Gaussian vectors:

$$f_{\boldsymbol{X}}\left(\mathbf{X}|\mathbf{R}_x\right) = \pi^{-NK}\left|\mathbf{R}_x^{-1}\right|^K \exp\left(-\sum_{k=1}^{K}\mathbf{x}_k^H\mathbf{R}_x^{-1}\mathbf{x}_k\right). \tag{C3}$$

It is important to note that based on our assumptions in Chapter 2 regarding the covariance data, the matrix $\mathbf{R}_x$ is completely defined by the underlying parameter vector

$$\boldsymbol{\alpha} = [\theta_1, \dots, \theta_M, \sigma_0^2, \sigma_1^2, \dots, \sigma_M^2]^T. \tag{C4}$$

Therefore, we can write $f_{\boldsymbol{X}}\left(\mathbf{X}|\mathbf{R}_x\right)$ equivalently as $f_{\boldsymbol{X}}\left(\mathbf{X}|\alpha\right)$. The log-likelihood function associated with Eq. (C3) (now seen as a function of the parameter vector conditioned on the data) is then

$$L\left(\boldsymbol{\alpha}|\mathbf{X}\right) = -NK\log\pi + K\log\left|\mathbf{R}_x^{-1}\right| - \sum_{k=1}^{K}\mathbf{x}_k^H\mathbf{R}_x^{-1}\mathbf{x}_k, \tag{C5}$$

where we note that we left off the dependence of $\mathbf{R}_x$ on $\boldsymbol{\alpha}$ for notational clarity.

C.1 Calculating the Gradient of the Likelihood Function

The MLE is obtained by maximizing Eq. (C5) with respect to $\boldsymbol{\alpha}$, which requires that we find the zeros of its gradient with respect to $\boldsymbol{\alpha}$. To simplify calculation of the gradient, we seek to rewrite the likelihood function in a more convenient form; first, we safely can drop the constant $\log\pi$, since it will vanish for any derivatives with respect to $\boldsymbol{\alpha}$. Second, we note

$$\mathbf{x}^H\mathbf{R}_x^{-1}\mathbf{x} = \operatorname{tr}\left[\mathbf{x}^H\mathbf{R}_x^{-1}\mathbf{x}\right], \tag{C6}$$

since $\mathbf{x}^H\mathbf{R}_x^{-1}\mathbf{x}$ is a scalar. Making use of the commutative property of the trace operator (Eq. (A3)), we have

$$\operatorname{tr}\left[\mathbf{x}^H\mathbf{R}_x^{-1}\mathbf{x}\right] = \operatorname{tr}\left[\mathbf{R}_x^{-1}\mathbf{x}\mathbf{x}^H\right]. \tag{C7}$$

Applying Eq. (C7) to the summation in Eq. (C5), we obtain

$$L\left(\alpha|\mathbf{X}\right) = -K\log|\mathbf{R}_x| - K\operatorname{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right], \tag{C8}$$

where, as before, $\mathbf{S}_x$ is the sample covariance matrix:

$$\mathbf{S}_x = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{x}_k^H. \tag{C9}$$

We now proceed to compute the gradient of the likelihood function with respect to the implicit spatial frequency parameters, which we denote as $\mathbf{D}L = \left[\frac{\partial L}{\partial\alpha_1},\ldots,\frac{\partial L}{\partial\alpha_{2M+1}}\right]^T$. Deriving this is simplified significantly by use of the chain rule and some the basic matrix calculus results derived in Appendix A. Considering the first term in Eq. (C8) we have

$$\begin{aligned}\frac{\partial}{\partial\alpha_i}\left[-K\log|\mathbf{R}_x|\right] &= -\frac{K}{|\mathbf{R}_x|}\frac{\partial|\mathbf{R}_x|}{\partial\alpha_i} && \text{(C10)}\\ &= -K\operatorname{tr}\left[\mathbf{R}_x^{-1}\frac{\partial\mathbf{R}_x}{\partial\alpha_i}\right], && \text{(C11)}\end{aligned}$$

where we note the use of Jacobis formula (Proposition 12) for arriving at the result. Now, considering the second term,

$$\begin{aligned}&K\frac{\partial\operatorname{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right]}{\partial\alpha_i} && \text{(C12)}\\ &= -K\operatorname{tr}\left[\frac{\partial\left[\mathbf{R}_x^{-1}\mathbf{S}_x\right]}{\partial\alpha_i}\right] && \text{(C13)}\\ &= -K\operatorname{tr}\left[\frac{\partial\mathbf{R}_x^{-1}}{\partial\alpha_i}\mathbf{S}_x\right] && \text{(C14)}\\ &= K\operatorname{tr}\left[\mathbf{R}_x^{-1}\frac{\partial\mathbf{R}_x}{\partial\alpha_i}\mathbf{R}_x^{-1}\mathbf{S}_x\right] && \text{(C15)}\\ &= K\operatorname{tr}\left[\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1}\frac{\partial\mathbf{R}_x}{\partial\alpha_i}\right], && \text{(C16)}\end{aligned}$$

where, in the last step, we made use of the commutative property $\operatorname{tr}[\mathbf{AB}] = \operatorname{tr}[\mathbf{BA}]$. Putting expressions Eqs. (C11) and (C16) together, we obtain the final form of the likelihood gradient:

$$(\mathbf{D}L)_i = K\operatorname{tr}\left[\left(\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1} - \mathbf{R}_x^{-1}\right)\frac{\partial\mathbf{R}_x}{\partial\alpha_i}\right],\ \text{for } i = 1,\ldots,2M+1. \tag{C17}$$

C.2 Calculating the Hessian of the Likelihood Function

Another important quantity relating to the MLE of the underlying parameters is the Hessian with respect to the likelihood function, which is defined as

$$\mathbf{D}^2 L = \begin{bmatrix} \frac{\partial^2 L}{\partial \alpha_1^2} & \frac{\partial^2 L}{\partial \alpha_1 \alpha_2} & \cdots & \frac{\partial^2 L}{\partial \alpha_1 \alpha_{2M+1}} \\ \frac{\partial^2 L}{\partial \alpha_2 \alpha_1} & \frac{\partial^2 L}{\partial \alpha_2^2} & \cdots & \frac{\partial^2 L}{\partial \alpha_2 \alpha_{2M+1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L}{\partial \alpha_{2M+1} \alpha_1} & \frac{\partial^2 L}{\partial \alpha_{2M+1} \alpha_2} & \cdots & \frac{\partial^2 L}{\partial \alpha_{2M+1}^2} \end{bmatrix}. \tag{C18}$$

To compute this, we can consider the derivative with respect to an arbitrary component of the gradient computed in Eq. (C17):

$$\frac{\partial^2 L}{\partial \alpha_i \alpha_j} = \frac{\partial}{\partial \alpha_i} K \mathrm{tr}\left[\left(\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1} - \mathbf{R}_x^{-1}\right)\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right], \text{ for } i, j = 1, \ldots, 2M+1. \tag{C19}$$

Since the trace operator is linear, it can be interchanged with the partial derivative; hence,

$$\begin{aligned} & \frac{\partial}{\partial \alpha_i} K \mathrm{tr}\left[\left(\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1} - \mathbf{R}_x^{-1}\right)\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right] && \text{(C20)} \\ & = K \mathrm{tr}\left[\frac{\partial}{\partial \alpha_i}\left(\left(\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1}\right)\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right) - \frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right)\right] && \text{(C21)} \\ & = K \mathrm{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j} + \mathbf{R}_x^{-1}\mathbf{S}_x\frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right) - \frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right)\right], && \text{(C22)} \end{aligned}$$

where in Eq. (C22), we used the matrix derivative chain rule applied to the first term in Eq. (C21). We elect to leave the Hessian for the physical array model in the current form of Eq. (C22) since, as we shall observe, when computing the associated Cramer-Rao lower bound, we can cancel out some of the terms without explicitly computing the remaining partials.

C.3 Cramer-Rao Lower Bound for the Physical Array Model

Recall that the CRLB is a lower bound of the estimation error *over all possible unbiased estimators* with respect to the unknown implicit parameters $\boldsymbol{\alpha}$. We derive the CRLB for the DOA estimation with respect to the linear model under the assumption that underlying sources are Gaussian random sources. For this we require the Fisher Information Matrix (FIM) which is given as

$$[\mathbf{J}]_{ij}(\boldsymbol{\alpha}) = -E\left[\frac{\partial^2}{\partial \alpha_i \partial \alpha_j}\log f_{\mathbf{x}}(\boldsymbol{x}|\boldsymbol{\alpha})\right]. \tag{C23}$$

The CRLB matrix then is simply the inverse of the FIM:

$$\mathbf{C}(\boldsymbol{\alpha}) = \mathbf{J}^{-1}(\boldsymbol{\alpha}). \tag{C24}$$

We are now in a position to derive the main CRLB result. Using the results in Eq. (C22), and the fact that $E\left[\cdot\right]$ is a linear operator, we have

$$
\begin{aligned}
[\mathbf{J}]_{ij}(\boldsymbol{\alpha}) &= -E\left[K\mathrm{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial u_l} + \mathbf{R}_{yy}^{-1}\mathbf{S}_K\frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right) - \frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right)\right]\right] && \text{(C25)}\\
&= K\mathrm{tr}\left[E\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\mathbf{S}_x\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j} + \mathbf{R}_x^{-1}\mathbf{S}_x\frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right) - \frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right)\right]\right] && \text{(C26)}\\
&= K\mathrm{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}E\left[\mathbf{S}_x\right]\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j} + \mathbf{R}_x^{-1}E\left[\mathbf{S}_x\right]\frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right) - \frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right)\right]. && \text{(C27)}
\end{aligned}
$$

Using the fact that

$$E\left[\mathbf{S}_x\right] = \mathbf{R}_x, \qquad \text{(C28)}$$

we can show that Eq. (C27) reduces to

$$
\begin{aligned}
&K\mathrm{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\mathbf{R}_x\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j} + \mathbf{R}_x^{-1}\mathbf{R}_x\frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right) - \frac{\partial}{\partial \alpha_i}\left(\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right)\right] && \text{(C29)}\\
&= K\mathrm{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right]. && \text{(C30)}
\end{aligned}
$$

Thus, from Eq. (C30), we obtain the CRB $\mathbf{C}(\boldsymbol{\alpha}) = \mathbf{J}^{-1}(\boldsymbol{\alpha})$ as

$$\left[\mathbf{C}^{-1}\right]_{ij} = K\mathrm{tr}\left[\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_i}\mathbf{R}_x^{-1}\frac{\partial \mathbf{R}_x}{\partial \alpha_j}\right], \text{ for } i, j = 1, \ldots, 2M+1. \qquad \text{(C31)}$$

Appendix D

COMPRESSIVE SENSING

Compressive sensing (CS) is a relatively recent mathematical framework for enabling the reconstruction of a signal from a small number of measurements. The term "small" here is often used in reference to uniform sampling at the Nyquist rate (in space or time). Fundamentally, CS encompasses measurement paradigms in conjunction with regularization techniques that enable physically meaningful solutions of the underlying underdetermined linear systems. For array processing, this is accomplished through sparse array architectures (random or deterministic) that are amenable toward processing with CS-based reconstruction algorithms. To get a basic understanding of such algorithms, first let us consider the noiseless DOA estimation problem for a single snapshot of array data,

$$\mathbf{A}\mathbf{s}(k) = \mathbf{x}(k), \tag{D1}$$

where the aim is to recover $\mathbf{s}(k)$, or at the very least, its angular support. Here, the signal $\mathbf{s}$ is taken over an angular grid $\boldsymbol{\theta}$ that discretizes the array's FOV. The assumption here is that the true angular support of $\mathbf{s}$, $\hat{\boldsymbol{\theta}}$ is a subset of the angular grid $\boldsymbol{\theta}$, i.e., $\hat{\boldsymbol{\theta}} \subset \boldsymbol{\theta}$. When the number of sources in the grid M exceeds the number of measurements N, Eq. (D1) is underdetermined, and thus an infinite number of solutions $\mathbf{s}$ will fit the data $\mathbf{x}$. By introducing some prior information regarding $\mathbf{s}$, we can effectively reduce the solution space in the hopes that solutions within this space yield the information that we seek. One important prior in CS is the concept of sparsity — that is, when the number of sources impinging upon the array (denoted as s) is much smaller than the number of angles in the grid, the true solution $\hat{\mathbf{s}}$ to Eq. (D1) contains mostly zeros. In other words, ideally, we should restrict our search only toward vectors that are s-sparse, i.e., vectors that have at most s nonzero elements. Under such a constraint, in many cases underdetermined systems such as (D1) admit *unique solutions*. Formally, such a search is given by the program

$$\begin{aligned} \min \|\mathbf{s}\|_0 \text{ subject to} \\ \mathbf{A}\mathbf{s}(k) = \mathbf{x}(k); \end{aligned} \tag{D2}$$

that is, *if a unique solution s-sparse solution does exist*, solving program Eq. (D2) is a concrete way of finding it. To characterize the existence of a unique sparse solution to a particular underdetermined system, we look to the null space of $\mathbf{A}$, which we denote as $N(\mathbf{A})$. Specifically, we need to ensure that $N(\mathbf{A})$ is void of any linear combinations of s-sparse vectors. It is not hard to show that the space of $2s$-sparse vectors encompasses all possible linear combinations of s-sparse vectors [45] and therefore a unique sparse solution can be ensured if and only if $N(\mathbf{A})$ does not contain any $2s$-sparse vectors. One measure that checks to ensure this null space condition is the Kruskal rank of a matrix [46].

Definition 9. *The Kruskal rank of a matrix* $\mathbf{A}$*, denoted as krank*$(\mathbf{A})$ *is the largest positive integer k such that* every set *of k columns of* $\mathbf{A}$ *is linearly independent.*

Theorem 2 (Unique Recovery of a Sparse Signal)**.** *A unique s-sparse solution to Eq.* (D2) *exists if and only if*

$$krank(\mathbf{A}) > 2s.$$

For the DOA estimation problem, using the fact that krank($\mathbf{A}$) $\leq$ rank($\mathbf{A}$) in conjunction with Theorem 2, we have the following corollary:

Corollary 1. *The maximum number of sources that can be uniquely recovered from SMV data for a given array $\mathbb{S}$ with $|\mathbb{S}| = N$ is $N/2 - 1$.*

In other words, no more source than half the number of elements can be imaged when restricted to SMV data. However this limitation can be overcome by leveraging temporal diversity. To show this, as before, we assume all the sources are on the grid and that our measurements are noiseless. We further assume that the spatial support of the sources does not change across the observation period T. In this case, the rows of $\mathbf{A}$ in

$$\mathbf{A}\mathbf{S}(k) = \mathbf{X}(k) \tag{D3}$$

share a common joint sparsity support $\boldsymbol{\theta}^*$. Then, similarly to the SMV case, we can seek to uniquely recover the MMV source data according to the program

$$\begin{aligned} \min_{\mathbf{S}} \in \mathbb{C}^{M\times K} \|\mathbf{S}\|_{0,r} \text{ subject to} \\ \mathbf{A}\mathbf{S}(k) = \mathbf{X}(k), \end{aligned} \tag{D4}$$

where we define

$$\|\mathbf{S}\|_{0,r} = \left\| \left(\sum_{k=1}^{K} |\mathbf{S}_{:,l}|^2 \right)^{1/2} \right\|_0 .$$

The results of [47] establish the conditions in which this possible.

Theorem 3. *Consider the MMV model in Eq.* (D3)*, where it is assumed that $\boldsymbol{\theta}^* \subset \boldsymbol{\theta}$ indicates the common directional support of the underlying sources across the rows of* $\mathbf{S}$*. Then the support $\boldsymbol{\theta}^*$ can be uniquely identified by the solution of Eq.* (D4) *if and only if*

$$M < \frac{krank(\mathbf{A}) + rank(\mathbf{X}_K)}{2},$$

Such conditions include the mutual coherence measure

$$\mu(\mathbf{A}) = \max_{i,j<S,i\neq j} \frac{\left|\mathbf{a}_i^H \mathbf{a}_j\right|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}.$$

For example, recovery with convex relaxed methods can be achieved as long as $M < (1 + \mu^{-1}(\mathbf{A}))/2$ [45], and while weaker recovery conditions could be established with other metrics, such as the restricted isometry property (RIP), mutual coherence is the only metric that is amenable for computing in polynomial time.

Appendix E

MATLAB CLASS HEADERS

```
classdef NAP_testbed < handle

    properties
        Valons; % Valon dual-channel sources 1:4
        Rigols; % Rigol dual-channel AWGs 1:4
        VSG; % Cal Arb/VSG combo
        DSO; % Keysight scopes
        fRF=3.4e9; % RF center frequency
        fIF_CAL = 25e6; % VSG/Arb IF
        fIF_AWG = 50e6; % Rigol/upconverter IF
        fIF_RX = 640e6; % Receive IF at scopes
        Tp = 30e-6; % Far-field pulse width (not noise)
        BW = 20e6; % Far-field pulse bandwidth (not noise)
        Tpre=100e-6; % extra delay before pulse to let noise gate turn off
        Tgate=100e-6; % extra delay before raising gate at end
        Tp_CAL = 200e-6; % cal pulse length
        BW_CAL = 40e6; % cal pulse bandwidth
        channels=1:4; % which Rigol/Valon combos to use, 1:4
        fs_Capture = 2e9; % scope sample rate
        sampleMode = 'RTIM'; % regular ('RTIM') or hi-res ('HRES') sampling
        T_capture = 2e-3; % capture time interval
        T_disp = 500e-6; % display interval
        Vpp_capture = 0.4; % Volts P-P setting on scopes
        trigdelay = 100e-6; % Make about the same as Tpre to keep from capturing junk
        FFsources = struct('type',repmat({'off'},1,8),'arb',repmat({0},1,8),'range',repmat({0},1,8),...
            'Doppler',repmat({0},1,8),'Amplitude',repmat({-20},1,8),'Kp',repmat({100},1,8),'d',repmat({0},1,8));
        FFDist = 16*0.3048; % distance to far-field source plane
        FFcal_src = []; % which sources to turn on early to get a clean far-field cal pulse
                        % before all the other sources turn on
        Aspect_Angle = 0;
    end % properties
```

```
% matlab class to control Valon programmable signal generator
classdef Valon < handle

    properties
        handle; % instrument-conrol serial fid
    end % properties

    % properties to set once at init time that should not change
    properties (SetAccess=private)
        port; % which com port is used
        baud=9600; % current baud rate
        ID; % read from instrument
        SerialNum;
    end

    properties (Dependent)
        % state variables stored in the Valon; some are per-channel and
        % others are per-device

        % per-device:
        Status;
        ReferenceSource;
        ReferenceFreq;
        ReferenceDouble;
        Lock;

        % per-channel:
        % Note: frequency quantization depends on frequency range.  For
        % 3-6GHz with a 10MHz reference, the step appears to be 4884 Hz, which
        % I think is 4MHz/819?  Experimentally, the device also seems to accept
        % frequencies that are multiples of 5000 Hz.  (Need to check what really comes out.)
        % Available frequencies: 3000e6, 3000e6+4884, 3000e6+5000, 3000e6+9768, 3000e6+10000
        %                        3001e6, 3001e6+1220, 3001e6+5000, 3001e6+6104, 3001e6+10000
        %                        3002e6, 3002e6+2442, 3002e6+5000, 3002e6+7326, 3002e6+10000
        %                        3003e6, 3003e6+3662, 3003e6+5000, 3003e6+8546, 3003e6+10000
        % and so one, cycling every 4MHz.  Around a center freq of 3.35 GHz, we have offsets
        % [-12212 -10000 -7328 -5000 -2444, 0, 2442, 5000 7326 10000 12210]*Hz
        % The manual lists 9768 Hz steps for a 20MHz
        % reference, which I confirmed directly;  The manual
        % suggests that turning off the double will give another factor of
        % 1/2, but I didn't find that to be the case.
        Output_Enable;
        Power; % determined by Attenuation
        Attenuator;
        Frequency;

    end % dependent properties
```

```
% Rigol.m : class to control a Rigol DG1032Z AWG

% matlab class to control Valon programmable signal generator
classdef Rigol < handle

    properties
        handle; % instrument-control VISA fid
    end % properties

    % properties to set once at init time that should not change
    properties (SetAccess=private)
        addr; % VISA address
        ID; % read from instrument
        Model;
        SerialNum;
    end

    properties (Dependent)
        % state variables stored in the Rigol; some are per-channel and
        % others are per-device

        % per-device
        RefSource;

        % per-channel  settings
        WaveformMode; % sine/square/ramp/pulse/noise/arb
        % Rigol has 2 Arb modes: SRATE/Sifi mode can have a variable (virtual) sample rate up to
        % 60MS/s and memory depth to 8MS.  FREQ/DDS mode has a fixed DAC sample rate of 200 MS/s and
        % a fixed memory length of 8192, which yields a PRI of 40.96us and a PRF of 24.414kHz.
        ArbMode; % FREQ or SRATE (DDS or SiFi modes)
        ArbSampleRate; % Sample rate in SRATE/Sifi mode
        Period; % seems to be general setting, applies to Arb waveform in DDS mode
        Burst; % burst mode on/off
        BurstMode; % TRIGgered|INFinity|GATed
        BurstCount; % number of cycles per burst (except noise)
        BurstDelay; % delay before furst pulse of burst (except noise)
        BurstTrigSource; % trig/gate source for burst mode (INT|EXT|MAN)
        BurstTrigSlope; % POS or NEG, also for gate
        BurstIdle; % value to hold between bursts; #|FPT|TOP|CENTER|BOTTOM
        Modulation; % on/off
        Impedance;
        Amplitude;
        AmpUnits; % amplitude units: VPP|VRMS|DBM
        Offset;
        Frequency; % This is the PRF for Arb waveforms, and fundamental for sin/square
        Sync; % ON/OFF: sync signal output (ext trigger setting might disable anyway)
        OutputEnable;

    end % dependent properties
```

```
classdef N5182A < handle

    properties
        handle; % instrument-control VISA fid
    end % properties

    % properties to set once at init time that should not change
    properties (SetAccess=private)
        addr; % VISA address
        ID; % read from instrument
        Model;
        SerialNum;
    end

    properties (Dependent)

        RefSource; % reference source, 'INT' or 'EXT'
        Amplitude;
        Frequency;
        OutputEnable;
        ALC;
        ArbEnable;
        ArbWaveform;
        ArbSampleRate;
        ArbFreqOffset; % Arb can digitally shift to an IF
        ArbMode; % continuous, triggered, sequence
        ArbTriggerSource; % button/software/external

    end % dependent properties
```

```
classdef Infinium < handle

    properties
        handle; % instrument-control VISA fid
    end % properties

    % properties to set once at init time that should not change
    properties (SetAccess=private)
        addr; % VISA address
        ID; % read from instrument
        Model;
        SerialNum;
        ActiveChannels;
    end

    properties (Dependent)
        % state variables stored in the Rigol; some are per-channel and
        % others are per-device

        % per-device
        AcqMode; % RTIM or HRES
        NumPoints; % this is the number of points to capture, not to display
        Tcapture; % alternate way to set the points
        RefSource;
        TrigSource;
        TrigLevel;
        TrigSlope;
        TrigDelay;
        TrigSweep; % trig or auto
        SampleRate;
        Tdisplay; % sets the display width, make this short for speed
        Nchan;
        Bandwidth; % acq. bandwidth limit (down to 1GHz)

        % per-channel
        VertRange;
        VertOffset;

    end % dependent properties
```