UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

DESIGN OF COMPRESSIVE ANTENNA ARRAYS

by

Heinrich Edgar Arnold Laue

Submitted in partial fulfillment of the requirements for the degree
Philosophiae Doctor (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

January 2020

SUMMARY

DESIGN OF COMPRESSIVE ANTENNA ARRAYS

by

Heinrich Edgar Arnold Laue

Promoter:	Prof. W. P. du Plessis
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Philosophiae Doctor (Electronic Engineering)
Keywords:	Antenna feed networks, beamforming, reduced-control arrays, compressive sensing (CS)

Reduced-control antenna arrays reduce the number of controls required for beamforming while maintaining a given array aperture. A reduced-control array for direction finding (DF), inspired by the concept of compressive sensing (CS), was recently proposed which uses random compression weights for combining antenna-element signals into fewer measurements. However, this compressive array had not been studied in terms of traditional characteristics such as directivity, sidelobe level (SLL) or beamwidth. In this work, random compression weights are shown to be suboptimal and a need for the optimisation of compressive arrays is expressed. Existing codebook optimisation algorithms prove to be the best starting point for the optimisation of compressive arrays, but are computationally complex. A computationally efficient codebook optimisation algorithm is proposed to address this problem, which inspires the compressive-array optimisation algorithm to follow. Compressive antenna arrays are formulated as a generalisation of reduced-control arrays and a framework is presented for their optimisation in terms of SLL. By allowing arbitrary compression weights, compressive arrays are shown to improve on existing reduced-control techniques. A feed network consisting of interconnected couplers and fixed phase shifters is proposed, enabling the implementation of compressive arrays in microwave hardware. The practical feasibility of compressive arrays is illustrated by successfully manufacturing a 3-GHz prototype compressive array with integrated antenna elements.

ACKNOWLEDGEMENTS

I would like to thank GEW Technologies, the National Commission on Research, Science and Technology, Namibia, and the University of Pretoria for their financial support. I would also like to thank my undergraduate-bursary provider, Namibia Water Corporation, for affording me the time to pursue postgraduate studies on a full-time basis.

My supervisor and mentor, Prof. Warren: Thank you for kindling the fire of curiosity in me. Your attention to detail, high standards, insight and integrity have inspired me to become a better researcher. You have taught me the value of learning from others, that we can do more together than apart.

I wish to thank Dr Tinus Stander and Hannes Venter of the Carl and Emily Fuchs Institute for Microelectronics and Prof. Wimpie Odendaal and Lukas Naudé at the Compact Antenna Test Range, University of Pretoria for assisting with the measurements of the prototypes. Thank you to Peet Kruger and his team at the mechanical workshop, University of Pretoria for manufacturing the mountings for the prototypes. I also wish to thank the staff at the University of Pretoria library, in particular Rachel Mahlangu, for sourcing many publications with tremendous efficiency.

Thank you to the anonymous reviewers of my publications that formed the basis of much of this thesis. Your input has greatly improved the quality of the work.

I sincerely thank my fellow postgraduates for their support—Wade Smith, Nicholas Osner, Frans-Paul Pieterse, Daniel Maidment, Chané Pieterse, Joshua Sendall, Hannes Venter, Marius-Corné Meijer, Jehane du Toit, and Natasha Ferreira. Your companionship eased the pressures of life as a postgraduate.

I am grateful towards my parents, Edgar and Ronelle Laue. The curiosity that you allowed to bloom from a young age has led me to where I am today. I also thank my sisters, Corlia Laue and Ester Greffrath, for their friendship and support.

Finally, praise is due to my Lord and Saviour, Jesus Christ. You have used this endeavour to grow me not only academically, but also as a person.

It is by logic that we prove, but by intuition that we discover. – Henri Poincaré

LIST OF ABBREVIATIONS

ADC	analogue-to-digital converter
AWGN	additive white Gaussian noise
BCASC	best complex antipodal spherical codes
CBGC	coherence-based Grassmannian codebook
CS	compressive sensing
CST	Computer Simulation Technology
DCT	discrete cosine transform
DF	direction finding
DFT	discrete Fourier transform
DMD	digital micromirror device
DoA	direction of arrival
FFT	fast Fourier transform
i.i.d.	independent and identically distributed
IDFT	inverse discrete Fourier transform
IFFT	inverse fast Fourier transform
IFT	iterative FFT technique
LNA	low-noise amplifier
MDCT	modified discrete cosine transform
MR	magnetic resonance
MVDR	minimum variance distortionless response
RF	radio-frequency
RFIC	radio frequency integrated circuit
RHS	right-hand side
RIP	restricted isometry property
RMS	root mean square
SLL	sidelobe level
SMA	subminiature version A
SNR	signal-to-noise ratio
SQP	sequential quadratic programming
UCA	uniform circular array
UHF	ultra-high frequency
ULA	uniform linear array
VSWR	voltage standing wave ratio

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

Beamforming antenna arrays consist of a group of antennas that, when combined together, are able to electronically steer a beam in different directions while the array itself remains stationary [2]. Beamforming arrays are used extensively in radar [2, 3], telecommunications [2], radio astronomy [4], and radio direction finding (DF) [5].

In traditional phased arrays, each element is connected to a phase shifter, and possibly, an amplifier [3]. Digital antenna arrays contain a receiver and/or transmitter at each antenna element. These hardware components are the beamforming controls which enable adaptive beamforming. Advantages of digital beamforming include the ability to steer multiple beams simultaneously in software, array reconfigurability, improved dynamic range, and precise array calibration [2, 3, 6]. The use of phase shifters and/or amplifiers at each element in traditional phased arrays, and transmitters and/or receivers at each element in digital antenna arrays places limitations on the number of elements that may be used due to the size, weight, complexity, and cost involved [3, 6, 7]. With uniform spacing between antenna elements, this limits the aperture of the array, and therefore, the achievable resolution.

Over the decades, significant effort has gone into developing reduced-control arrays which maintain a given aperture but require either fewer antenna elements, or fewer beamforming controls in an effort to reduce cost, size, and weight [3, 7–9].

One way of reducing the cost and complexity of an array is to take a filled array with a large aperture and turn some of the elements off, referred to as array thinning [9]. This results in an array with a beamwidth similar to that of the filled array, but with decreased directivity proportional to the number of active elements [2, 10].

Another way of reducing the number of controls is to combine the signals at the antenna elements before reaching the transmitters and/or receivers, thereby reducing the number of controls for the same number of antenna elements [2–4, 7, 11]. Existing overlapped subarrays combine antenna elements while considering a reduced steering range, resulting in narrow beams with high directivity being steered over a relatively small angular range [2, 3, 7].

An array that combines the antenna-element signals so that each output is a function of all antenna elements for a full field of view, inspired by the concept of compressive sensing (CS), was proposed in [11] for DF. Such compressive arrays were proposed to use random compression weights to combine the antenna-element signals, since this approach has worked well in other CS applications [12, 13]. However, a recent hardware implementation of a compressive array with random weights exhibited ambiguities when performing DF, implying that weights would need to be optimised if compressive

arrays are to be useful in practice [14]. Typical array specifications such as directivity, beamwidth and sidelobe level (SLL) are all functions of the radiation pattern of an array [2]. The optimisation of compressive arrays is therefore a pattern-synthesis problem.

One approach to array pattern synthesis is to minimise SLL in a pre-defined sidelobe region [15]. Interestingly, the minimisation of SLL is analogous to the problem of codebook optimisation [1]. Codebook optimisation algorithms aim to minimise the similarity between codewords (vectors) in a codebook (matrix), while minimising SLL involves minimising the similarity between the steering vectors of an array. Unfortunately, the codebook optimisation problem is inherently combinatorial in nature, leading to computationally expensive algorithms [16]. Any approach to optimising compressive arrays would need to take such complexity into account if a computationally feasible algorithm is to be developed.

1.1.2 Research gap

Random compression weights were proposed for use in a compressive array since that is what is widely used in CS [11]. Such a compressive array was implemented in hardware with two subarrays and four antenna elements for DF, but some steering angles exhibited ambiguities, and a need was expressed for optimising the feed network [14]. In addition, compressive arrays have not been characterised in terms of directivity, SLL, or beamwidth, all of which are standard measures for characterising an antenna array [2].

Random compression weights are suitable for signal reconstruction in a CS framework, but only under the assumption that the dimensions of the system are sufficiently large [17]. This may not be the case for antenna arrays where arrays with very few elements are still of practical interest.

Existing codebook optimisation techniques may be applied to CS systems with predictable performance, including systems of small dimensions [18, 19]. However, even then the optimisation is only performed on a discrete number of vectors. Codebook optimisation algorithms are, therefore, unsuitable in their current form for optimising SLL which relies on continuous sampling of the sidelobe patterns.

These observations show that there is a need for deeper investigation into the design of compressive arrays as a pattern-synthesis problem. The suitability of random compression weights is questionable, and current codebook optimisation techniques cannot solve a continuous SLL optimisation problem.

Existing array synthesis methods for conventional arrays consider not only uniform minimisation of SLL, but also allow arbitrary shaping of the sidelobe amplitude patterns [20]. Allowing arbitrary sidelobe masks to be specified would make a compressive array design algorithm suitable for a wider range of applications.

In addition to the design of compression weights for compressive arrays is the problem of implementing such designs in hardware. The only hardware architecture proposed to date [14] is restricted to random weights. Unless hardware constraints are added to the already-complex problem of optimising compression weights, a technique for implementing optimised compressive arrays would have to be able to implement arbitrary weights. No such technique has yet been proposed.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

Based on the context and motivation given above, two main research questions were identified. The first question is whether compressive-array compression weights can be designed to meet pre-

defined radiation-pattern requirements. Under this question, the following sub-questions have been identified.

Suitability of random compression weights: Are random compression weights, widely used in CS applications, suitable for designing compressive arrays?

Compressive arrays based on optimised codebooks: Can existing numerically optimised codebooks be used to design compressive arrays, and will there be an increase in performance compared to random compression weights?

Addressing computational complexity in codebook optimisation algorithms: Can the computational complexity of existing codebook optimisation algorithms be addressed?

Developing a compressive array SLL minimisation algorithm: How is codebook optimisation related to SLL minimisation, and can insight gained from optimising discrete codewords in a codebook be used to develop an algorithm for optimising SLL in a compressive array?

Arbitrary sidelobe shaping: Can the shape of sidelobe regions in compressive arrays be controlled by means of arbitrary amplitude-pattern masks?

The second question is whether compressive arrays are feasible to implement in hardware. Under this question, the following sub-questions have been identified.

Hardware implementation of arbitrary compression weights: Can a hardware architecture be developed to implement arbitrary compression weights in compact and cost-effective microwave circuitry?

Practical feasibility of compressive arrays: Is it feasible to implement optimised compressive array systems in microwave hardware?

1.3 APPROACH

1.3.1 Hypotheses

In relation to the research questions listed above, the following hypotheses were formulated.

Suitability of random compression weights: Random weights can, at best, be assumed to be suitable for a compressive array with a large number of antenna elements [17]. The implementation of a small two-subarray, four-element compressive array for DF exhibited ambiguities at some steering angles [14]. It is expected that the performance resulting from the use of random weights will decrease in direct proportion with the dimensions of the system. This warrants the use of numerically optimised weights with predictable properties that do not rely on statistical measures.

Compressive arrays based on optimised codebooks: The codewords in an optimised codebook represent the steering vectors in a compressive array, and not the compression weights themselves. In order to find the compression weights from the steering vectors, an inverse transformation will have to be applied. Since there are only a discrete number of compression weights, the rank of the transformation will also be discrete and limited. It is therefore expected that optimised codebooks will only allow control over the array patterns at a limited number of angles.

Addressing computational complexity in codebook optimisation algorithms: The formulation of the state-of-the-art best complex antipodal spherical codes (BCASC) codebook optimisation algorithm requires minimising the Euclidean distance between codewords and all their antipodals, leading to a computationally expensive algorithm [16]. If this approach is extended to the even more complex

problem of optimising SLL at large number of points, it is expected that such an approach will become computationally infeasible for antenna arrays of even moderate sizes. It is expected that the discrete codebook optimisation problem can be reformulated to directly minimise the similarity between the codewords as measured by their pair-wise absolute dot products (coherence), since this implicitly includes the effect of the antipodals and is more consistent with the desired goal. This would provide a better starting point for developing an SLL optimisation algorithm for compressive arrays.

Developing a compressive array SLL minimisation algorithm: Antenna amplitude patterns can be calculated by taking the absolute dot product of the conjugated beamforming weights and the steering vectors, and the SLL is given as the maximum pattern value outside the main beam. Similarly, coherence consists of a maximum absolute dot product. It is expected that coherence and SLL will become equivalent if the steering weights used in a compressive array are properly defined. However, this requires defining coherence as a continuous value over all angles in the sidelobe regions. Standard codebook optimisation algorithms consider only a discrete number of vectors and will therefore not suffice. Instead, the problem will have to be reformulated to minimise SLL over continuous sidelobe patterns using insight gained from discrete coherence optimisation techniques.

Arbitrary sidelobe shaping: In codebook optimisation algorithms, the similarity between codewords must be minimised for all possible combinations of codewords, which translates to optimising the sidelobe patterns over all steering angles in a compressive array. This implies that a good SLL at, for example, broadside does not guarantee a good SLL elsewhere. It is expected that this property may be exploited to design compressive arrays where the sidelobe specifications change based on steering angle, opening up a new set of design freedoms that may be used to compromise sidelobe performance at certain angles for better performance elsewhere.

Hardware implementation and practical feasibility of compressive arrays: A two-subarray ultra-high frequency (UHF) compressive array with four elements and a random sensing matrix was successfully implemented using lumped-element splitters and combiners, but requires a large number of crossovers [14]. Partially and completely overlapped subarrays can be seen as a special case of the compressive array, since the antenna elements are also weighted and combined to reduce the number of controls [2, 7]. Such feed networks have successfully been implemented in practice [2, 3, 7, 21]. It is expected that existing techniques for implementing compressive arrays and overlapped feed networks may be extended, or new techniques devised, to implement compressive feed networks with arbitrary compression weights.

1.3.2 Methodology

In order to test and evaluate the hypotheses given above, the following methodology was followed.

Suitability of random compression weights and compressive arrays based on optimised codebooks: Optimised codebooks designed using an existing algorithm were transformed to compression weights and compared to random compression weights in terms of coherence and DF accuracy as a measure of performance in a practical application. Compressive arrays optimised using the approach to follow were also compared to and evaluated against random and codebook-based compressive arrays.

Addressing computational complexity in codebook optimisation algorithms: The computational complexity of the BCASC codebook optimisation algorithm was addressed by reformulating the problem to directly minimise coherence. The proposed and BCASC algorithms were implemented

in a numerical computation environment and compared in terms of achieved coherence and run times over multiple tests and multiple codebook dimensions.

Developing a compressive array SLL minimisation algorithm with arbitrary sidelobe shaping: Using insight gained from developing the proposed codebook optimisation algorithm, the problem was extended to optimise SLL over all angles in the sidelobe regions. The problem was formulated so that arbitrary sidelobe masks may be applied for shaping the sidelobe regions.

Hardware implementation and practical feasibility of compressive arrays: Existing overlapped subarray implementation techniques were extended to allow the implementation of arbitrary compressive feed networks. A compressive feed network was first designed and manufactured in isolation to show that a given feed network response can be realised accurately. Subsequently, a complete prototype compressive array with integrated antenna elements was designed to show that compressive arrays are feasible to implement in practice. The microwave circuitry was simulated using a full-wave electromagnetic solver, manufactured, tested, and evaluated against the specifications.

1.4 RESEARCH GOALS

The concept of compressive arrays has been suggested [11] and a prototype implemented [14], but such arrays were proposed to use only random compression weights and were not evaluated in terms of any of the measures traditionally used for antenna arrays [2]. The goal of this work is to develop a framework for the design of compressive arrays as a pattern-synthesis problem, and to evaluate the potential of compressive arrays against existing reduced-control arrays. The goal of the work is also to investigate whether optimised compressive arrays are feasible to implement in microwave hardware.

1.5 RESEARCH CONTRIBUTION

The contributions of this work are summarised below.

Suitability of random compression weights and compressive arrays based on optimised codebooks: The use of numerically optimised codebooks in the design of compressive arrays has been proposed and compared to designs based on random weights. Codebook-based compressive arrays outperform random compressive arrays in terms of both coherence and mean DF performance in a noisy environment, showing that random weights are suboptimal for small CS systems.

Addressing computational complexity in codebook optimisation algorithms: An algorithm for optimising codebooks by directly minimising coherence has been proposed. The proposed algorithm outperforms the state-of-the-art BCASC algorithm by almost an order-of-magnitude reduction in median run time while maintaining similar coherence results.

Developing a compressive array SLL minimisation algorithm with arbitrary sidelobe shaping: A generalised framework for the optimisation of compressive arrays with arbitrary sidelobe specifications has been proposed. The algorithm is inspired by the coherence-based codebook optimisation algorithm also developed as part of this work. The coherence optimisation problem is extended to consider the minimisation of SLL in a compressive array. The approach is validated against a Chebyshev uniform linear array (ULA) and shown to improve on an existing weighted thinned array, a conventional ULA with a limited steering range, and a dual-transform overlapped subarray system for the same number of beamforming controls.

Hardware implementation and practical feasibility of compressive arrays: A novel network of interconnected couplers and phase shifters has been proposed for implementing an arbitrary

complex-valued feed network response in microwave hardware. The proposed approach is validated by approaching the standard implementation of a 4×4 Butler matrix. Three independently designed arrays are combined to share the same 8-element array, illustrating the versatility of the proposed approach. A planar compressive feed network with two subarrays and four antenna-element ports is implemented and manufactured in microstrip for a 3.15-GHz ULA and achieves the desired response within 0.6 dB and 5.2°. A complete prototype compressive array with two subarrays and an integrated array of four printed dipole elements is successfully manufactured in a planar circuit, achieving the desired SLL within 1.1 dB at the design frequency of 3 GHz.

The theory of CS is a relatively new concept which challenges the well-established Nyquist-Shannon sampling paradigm, making it initially challenging to both understand and accept. As a by-product of the work outlined above, a tutorial on CS has been developed which consists of new analogies, numerical examples and a simple reconstruction algorithm to explain the fundamental concepts in CS. This tutorial forms the basis of Chapter 2.

1.6 RESEARCH OUTPUTS

The publications resulting from this work are outlined below.

Suitability of random compression weights and compressive arrays based on optimised codebooks: This part of the work has led to the publication of the following conference paper [22]:
H. E. A. Laue and W. P. du Plessis, "Compressive direction-finding antenna array," in *IEEE-APS Topical Conf. on Antennas and Propag. in Wireless Commun. (APWC)*, Sep. 2016, pp. 158–161.

Addressing computational complexity in codebook optimisation algorithms: This part of the work has led to the publication of the following journal paper [23]:
H. E. A. Laue and W. P. du Plessis, "A coherence-based algorithm for optimizing rank-1 Grassmannian codebooks," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 823–827, June 2017.

Developing a compressive array SLL minimisation algorithm with arbitrary sidelobe shaping: This part of the work has led to the publication of the following journal paper [1]:
H. E. A. Laue and W. P. du Plessis, "Numerical optimization of compressive array feed networks," *IEEE Trans. Antennas Propag.*, vol. 66, no. 7, pp. 3432–3440, July 2018.

Hardware implementation of arbitrary compression weights: This part of the work has led to the publication of the following journal paper [24]:
H. E. A. Laue and W. P. du Plessis, "A checkered network for implementing arbitrary overlapped feed networks," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 11, pp. 4632–4640, Nov. 2019.

An introduction to CS: This part of the work has led to the publication of the following accredited-magazine article [25]:
H. E. A. Laue, "Demystifying compressive sensing [Lecture notes]," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 171–176, July 2017.

1.7 THESIS OVERVIEW

An overview of the thesis as contained in the relevant chapters is given below.

Chapter 1 introduces the problem of reducing the number of controls in beamforming arrays and highlights some of the limitations of the recently-proposed compressive DF array [11]. The assumption that random weights are optimal for compressing antenna-element signals is challenged, and is it suggested that compression weights should instead be optimised for low SLL.

Chapter 2 introduces the fundamental concepts in CS in the form of a tutorial with analogies, numerical examples, and a simple algorithm, thereby providing intuitive explanations for the CS principles referred to throughout the thesis.

Chapter 3 gives an overview of existing reduced-control arrays including thinned and sparse arrays, subarray systems, and compressive antenna arrays. An overview of discrete codebook optimisation is also given, which is the most promising starting point for developing a compressive array optimisation algorithm.

Chapter 4 investigates whether random compression weights are indeed optimal for compressive arrays when applied to DF. A new technique for designing compressive arrays using optimised codebooks is proposed and evaluated against conventional arrays and random compressive arrays.

Chapter 5 addresses the issue of computational complexity in the existing state-of-the-art codebook optimisation algorithm. A computationally efficient codebook optimisation algorithm is proposed which offers a more promising starting point for developing a compressive array optimisation algorithm.

Chapter 6 develops a framework and algorithm for optimising compressive arrays for low SLL. It is shown how compressive arrays are a generalisation of existing reduced-control arrays. Compressive arrays designed using the proposed approach are evaluated against existing reduced-control arrays and shown to implement designs which could not previously be considered.

Chapter 7 addresses the issue of how a compressive feed network can be implemented in hardware by proposing a novel feed network layout which can implement arbitrary complex-valued compression weights. Measured results are presented for a network implemented in microstrip.

Chapter 8 describes the successful design and manufacturing of a prototype compressive array with integrated antenna elements in a microwave circuit. Various practical considerations that arise when implementing compressive arrays in hardware are highlighted.

Chapter 9 summarises the main contributions of the work and highlights some potential considerations for future work.

Addendum A derives the proposed codebook and compressive array optimisation algorithms.

A glossary is also provided which defines terminology used throughout the thesis.

CHAPTER 2 AN INTRODUCTION TO COMPRESSIVE SENSING

2.1 CHAPTER OVERVIEW

The conventional Nyquist-Shannon sampling theorem has been fundamental to the acquisition of signals for decades, relating a uniform sampling rate to the bandwidth of a signal. However, many signals can be compressed after sampling, implying a high level of redundancy. The theory of CS presents a sampling framework based on the rate of information of a signal and not the bandwidth, thereby minimising redundancy during sampling. This means that a signal can be recovered from far fewer samples than conventionally required.

When considering new fields that challenge existing paradigms, it is often useful to start by gaining a high-level understanding of the principles underlying that field. This chapter employs a new set of analogies, illustrations and numerical examples to provide intuitive explanations for the fundamental principles in CS, allowing the reader who is new to CS to understand the references to CS in the chapters to follow.

Two analogies illustrating the core concepts in CS are presented in Sections 2.2 and 2.3. Section 2.4 explains the concept of sparsity—the fundamental assumption behind CS. Section 2.5 explains the concept of incoherence and how compressed measurements are taken. In Section 2.6, the CS problem is reformulated to consider it from a more in-depth mathematical perspective. A simple reconstruction algorithm in Section 2.7 shows how perfect reconstruction of the original signal from sub-Nyquist samples is possible, and considers the rationale behind computationally efficient ℓ_p-norm recovery algorithms. Finally, conclusions are drawn and the chapter is summarised in Section 2.8.

2.2 ANALOGY 1—LISTENING WITH HALF AN EAR

Three journalists are taking notes at a press conference. The first is inexperienced and quite naïve; so afraid of missing something important, he frantically writes down every word being said. The second journalist is more experienced and while also listening attentively to every word, he interprets what is said and summarises the facts concisely in his notes. The third journalist is quite lazy; not listening attentively at all, he only picks up every second or third word being said. The experienced journalist sees the lazy one daydreaming and is greatly surprised afterwards to find their notes almost identical. ‘How did you get all the facts, when you were clearly only listening with half an ear?’, he asks. ‘Did you not know’, the lazy one replies, ‘that the speaker is known to waffle, using ten words to convey a single concept? I am not likely to miss anything important when I know he could not be saying very much, though his words may be many.’

The naïve journalist represents conventional Nyquist-rate sampling, with words analogous to samples. This scheme makes no effort to interpret what is being sampled; the sampling rate is based purely on the signal bandwidth. The experienced journalist represents conventional signal compression. Sampling is still done at the Nyquist rate, but the system interprets the sampled signal and expresses it more concisely for storage or transmission. The lazy, or rather, the efficient journalist represents compressive sampling. This scheme samples well below the Nyquist rate by assuming that the unique concepts or information being conveyed is little, and that this information is distributed over the many conventional samples so that missing a particular sample is unlikely to lead to significant loss of information.

2.3 ANALOGY 2—FILLING IN THE GAPS

Imagine playing a game where a friend chooses a word and reveals the letters and their positions one by one, at random, until you correctly identify all the letters. You have gotten as far as 'co–pr–ss'. You see that the answer must be 'compress', and that the missing letters are 'm' and 'e'. How did you identify these two letters? You did not see the letters as unrelated parts to be identified on their own, but saw them as collectively conveying a single concept—a word.

Compressive sampling is analogous to such a game, where letters represent samples. Instead of seeing samples as unrelated to each other, CS identifies the concepts the samples are collectively conveying. These concepts are few compared to the number of samples, like many letters convey a single word. Some samples required by the Nyquist theorem are missing, but by identifying the concepts being conveyed, the CS algorithm is able to fill in the missing samples.

On the other hand, imagine playing the game without the assumption that the letters form a word. This significantly increases the complexity of the problem, since any combination of letters is a possible solution. Without any vocabulary to draw from, there is no way to identify the missing letters and you will have to wait until all the letters have been revealed.

Conventional sampling is like the latter scenario. It does not interpret what the samples are conveying, but treats each sample as an individual concept to be identified correctly. All the Nyquist samples must be taken; there is no way of filling in any gaps.

2.4 SPARSIFYING BASES

Consider a discrete signal vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}}$ of N samples taken at the conventional Nyquist rate of at least twice the signal bandwidth. A signal vector in a conventional sampling domain can be expressed in a different domain/basis for analysis or processing. For example, time domain signals may be transformed to the Fourier domain to analyse their frequency content.

Consider a set of orthonormal basis vectors placed as columns in the transform matrix $\mathbf{A} = [\mathbf{a}_1\, \mathbf{a}_2\, \cdots\, \mathbf{a}_N]$. The signal vector $\mathbf{x}$ can then be expressed as a weighted sum of basis vectors [12]

$$\mathbf{x} = \sum_{n=1}^{N} s_n \mathbf{a}_n = \mathbf{A}\mathbf{s}, \tag{2.1}$$

where the $N \times 1$ vector $\mathbf{s}$ contains the coefficients of the signal in its new basis, found as the projection of $\mathbf{x}$ onto each of the basis vectors by the dot product $s_n = \mathbf{a}_n^{\mathrm{T}}\mathbf{x}$ or $\mathbf{s} = \mathbf{A}^{\mathrm{T}}\mathbf{x}$ in the real case and $\mathbf{s} = \mathbf{A}^{\mathrm{H}}\mathbf{x}$ in the complex case. Each coefficient is represented by its own basis vector which is separable from all others.

Sampling at the Nyquist rate guarantees perfect recovery of the original signal, suggesting that no fewer than N coefficients are required to fully describe the signal. However, a signal vector can often be

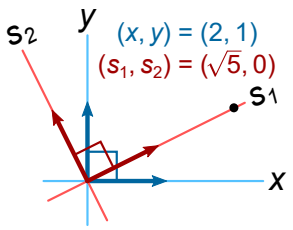


Figure 2.1. Illustration of a two-dimensional data point expressed in standard and sparsifying bases. Adapted from [25], ©2017 IEEE.

expressed in a different basis where many coefficients are zero (or close to zero) [13]. The remaining K non-zero, or significant, coefficients are sufficient to fully describe the signal. When $\mathbf{x}$ is expressed in a sparsifying basis, it results in a K-sparse vector $\mathbf{s}$ with only $K \ll N$ significant coefficients. Sparsity in $\mathbf{s}$ implies redundancy in $\mathbf{x}$, since N samples represent a signal with effectively only K degrees of freedom. Here, the significant coefficients in the sparsifying basis can be seen as the concepts or information being conveyed by $\mathbf{x}$.

Figure 2.1 illustrates a two-dimensional data point at $(2, 1)$ in the standard basis. However, in the sparsifying basis shown, the data point is $(\sqrt{5}, 0)$, which has only one significant coefficient.

Many signals have sparsifying bases, a fact well known in conventional compression where a signal may be expressed in a sparsifying basis so only the largest coefficients can be retained [12].

Images are typically sparse in the discrete cosine transform (DCT) or wavelet bases [26], audio signals in the modified discrete cosine transform (MDCT) basis [27], magnetic resonance (MR) images in the spatial, spatial finite differences or wavelet domains [28], and sensor array data in direction of arrival (DoA) [29].

2.5 INCOHERENCE AND COMPRESSIVE SAMPLING

CS exploits redundancy to reduce the number of samples that must be taken to fully describe a signal. It has been noted that redundancy can be quantified in terms of the number of significant coefficients in a sparsifying basis. CS aims to reduce the required number of samples without any prior knowledge of the signal, only the assumption that some sparsifying basis exists.

Consider as a first approach simply neglecting some of the conventional Nyquist samples at random. The problem here is that some samples neglected may contain crucial information. For example, what if the signal contains spikes or sharp discontinuities? Neglecting samples in these areas would lead to significant loss of information.

In order for this approach to work, it would be necessary for the information to be distributed evenly over all the conventional samples, so that no one sample conveys significantly more information than another. While the signal should be sparse in some other basis, it should certainly not be sparse in the domain in which it is sampled.

The incoherence[1] between two domains expresses the idea that a vector which is sparse in one domain will be non-sparse in the other, and occurs when the basis vectors between the domains are dissimilar [13]. Consider the time and frequency domains, where the time domain is represented by

[1]The term incoherence in this context should not be confused with the term coherence used elsewhere. Both measure the similarity between vectors, but whereas coherence measures the similarity between vectors in a single matrix/codebook/basis, incoherence measures the dissimilarity of vectors between two matrices/codebooks/bases.

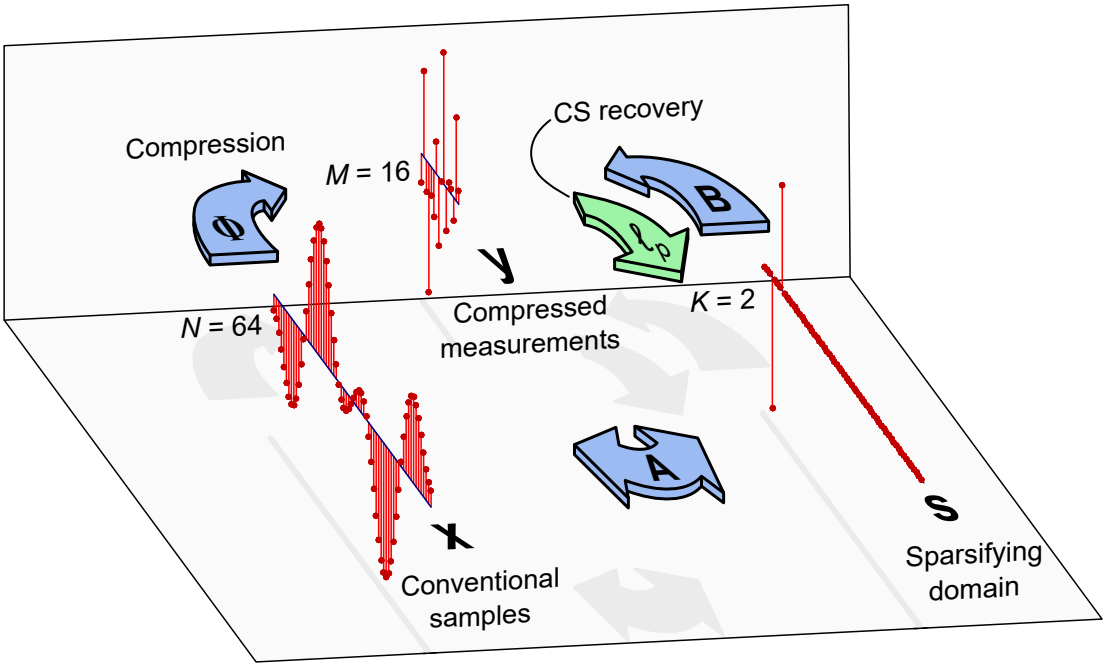


Figure 2.2. Example of a CS system with $N = 64$, $M = 16$, $K = 2$, random Gaussian $\mathbf{\Phi}$, and $\mathbf{A}$ the inverse DCT matrix. Blue arrows represent matrix multiplications. Amplitudes are not to scale. Adapted from [25], ©2017 IEEE.

the standard basis and the frequency domain by the Fourier basis. A single frequency component will result in a sine wave with most time samples being significant. Similarly, a time-domain impulse can only be represented by a multitude of frequency components. The bottom of Figure 2.2 shows an example where $\mathbf{A}$ is the inverse DCT matrix. The blue arrow represents a matrix multiplication with $\mathbf{A}$ or its inverse, depending on the direction. With only two significant coefficients in $\mathbf{s}$, almost all conventional samples in $\mathbf{x}$ are significant.

If the bases of $\mathbf{x}$ and $\mathbf{s}$ are incoherent, and if $\mathbf{s}$ is sparse, then the information of interest (the K significant coefficients in $\mathbf{s}$) will be distributed over all N samples in $\mathbf{x}$. Since this information is comparatively little ($K \ll N$), neglecting some of the samples in $\mathbf{x}$ is unlikely to lead to a significant loss of information (see Analogy 1, Section 2.2). Also, the choice of which samples to neglect becomes almost arbitrary, as long as enough are kept [13]. Aliasing is not a problem unless the samples that are kept are spaced uniformly; therefore, samples are typically neglected at random [13].

So far the requirements are sparsity in $\mathbf{s}$ and non-sparsity in $\mathbf{x}$, which is met when there is incoherence between the conventional and sparsifying bases. But what happens if $\mathbf{x}$ itself is sparse? Redundancy is still present, but neglecting samples at random will result in a loss of information. The solution is to transform the signal $\mathbf{x}$ to an intermediate domain before sampling—a domain in which the signal is non-sparse, and which is incoherent with the sparsifying domain.

How will such an intermediate domain be selected? An interesting fact is that a random basis is highly likely to be incoherent with almost any other basis [13]. This means that without prior knowledge of the sparsifying domain, the signal may be transformed to a random domain and the incoherence requirement will be met. In this way universal sampling schemes can be designed which require only the assumption that some sparsifying basis exists. No knowledge of the sparsifying basis is required during sampling, only during signal reconstruction.

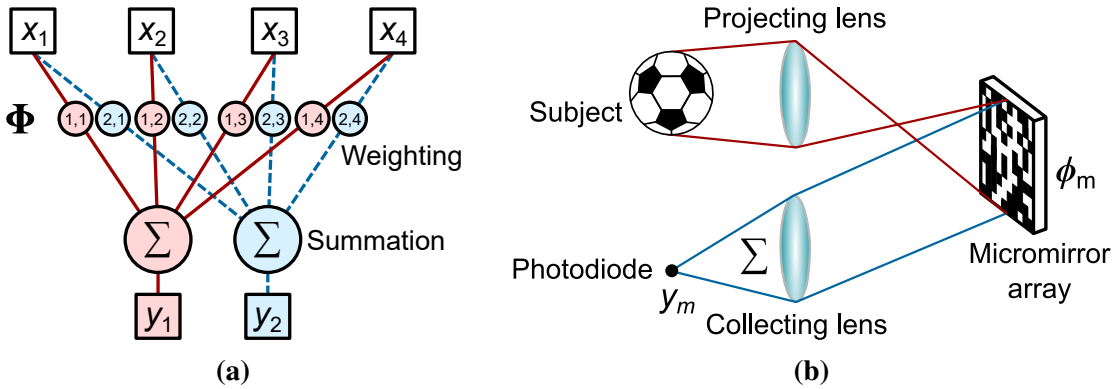


Figure 2.3. (a) General illustration of a sensing matrix implementation ($N = 4$, $M = 2$). (b) Single-pixel CS camera (adapted from [12]). Adapted from [25], ©2017 IEEE.

How is compressive sampling implemented? The simultaneous process of transforming to an intermediate domain and neglecting transformed samples may be described by the system $\mathbf{y} = \mathbf{\Phi x}$, where $\mathbf{y}$ is the $M \times 1$ vector of compressed samples (or measurements), and $\mathbf{\Phi}$ is the rectangular $M \times N$ sensing matrix, with $K < M < N$. The rows $\{\boldsymbol{\phi}_m\}_{m=1}^{M}$ describe the basis vectors of the intermediate domain, where only M out of N basis vectors are used. Each measurement is found as the projection of $\mathbf{x}$ onto the corresponding basis vector as $y_m = \boldsymbol{\phi}_m \mathbf{x}$. Figure 2.2 illustrates how a random $\mathbf{\Phi}$ is used to compress $\mathbf{x}$ into the measurement vector $\mathbf{y}$.

Each measurement y_m is a unique weighted combination of all the elements in $\mathbf{x}$, or $y_m = \sum_{n=1}^{N} \phi_{m,n} x_n$. Notice how this ensures that the information in $\mathbf{x}$ is distributed over all the measurements in $\mathbf{y}$. The processes of weighting and summation are used for hardware implementations of the sensing matrix. This is illustrated in Figure 2.3(a) for $N = 4$ and $M = 2$.

While a form of CS may be implemented in software to compress Nyquist-rate signals, the real power of CS lies in developing new hardware-based sub-Nyquist sampling schemes.

Figure 2.3(b) illustrates a CS application, the single-pixel camera, where an N-pixel image is reconstructed from $M < N$ measurements [12, 26]. An image is projected onto a digital micromirror device (DMD)—an array of tiny mirrors, each representing a pixel. For each measurement y_m, the micromirrors are randomly set to either reflect light toward a collecting lens, or away from it. This is the process of weighting (by zero or one), with the weights for each measurement taken from the corresponding row in $\mathbf{\Phi}$. The lens then collects the rays from the DMD and concentrates it onto a single photodiode—the process of summation. After the first measurement, new weights are programmed into the DMD for the next measurement.

2.6 EQUIVALENT SYSTEMS AND DESIGN REQUIREMENTS

So far it has been seen that CS is possible when a signal has a sparse representation in some domain that is incoherent with the sampling domain. To understand the problem in more detail it is useful to consider it from a slightly different angle. Consider the effect of $\mathbf{\Phi}$ and $\mathbf{A}$ not separately, but together in the following equivalent formulation of the CS problem:

$$\mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi A s} = \mathbf{B s}, \tag{2.2}$$

where $\mathbf{B}$ is the $M \times N$ compressed transform matrix. The columns $\{\mathbf{b}_n\}_{n=1}^{N}$ are vectors in an overcomplete basis, with more vectors than dimensions. Each vector still represents a coefficient in $\mathbf{s}$, but the set cannot be orthogonal and there will be some similarity between the vectors.

The goal is to recover $\mathbf{s}$ from $\mathbf{y} = \mathbf{Bs}$. Notice that the system is underdetermined since there are less equations than unknowns. It would generally be impossible to determine $\mathbf{s}$ uniquely since $\mathbf{y}$ does not present sufficient information about it. However, the assumption of sparsity in $\mathbf{s}$ is the additional information required to obtain a unique solution. Out of the infinite number of possible solutions, only the sparsest solution is considered, i.e. the sparsest vector $\mathbf{s}$ that satisfies $\mathbf{y} = \mathbf{Bs}$. This is a form of Occam's razor: out of all the possible explanations for the measured data, assume that the simplest (sparsest) one must be correct (see Analogy 2, Section 2.3).

Figure 2.2 shows the relationship between $\mathbf{\Phi}$, $\mathbf{A}$ and $\mathbf{B}$. The arrow labelled 'CS recovery' indicates that using a CS algorithm, it is possible to go upstream by finding the length-N vector $\mathbf{s}$ from the smaller length-M vector $\mathbf{y}$. Having $\mathbf{s}$, the original vector $\mathbf{x}$ can be found through $\mathbf{A}$.

Consider the following example. An underdetermined system in the form $\mathbf{y} = \mathbf{Bs}$ is given by

$$\begin{bmatrix} -0.357 \\ 0.612 \\ 0.137 \end{bmatrix} = \begin{bmatrix} -1.036 & -0.293 & -0.127 & -0.503 & -0.127 \\ -0.844 & 0.088 & -0.385 & 0.105 & 1.048 \\ 0.241 & -0.069 & -0.444 & 0.733 & 0.412 \end{bmatrix} \begin{bmatrix} 0 \\ 1.0 \\ 0 \\ 0 \\ 0.5 \end{bmatrix}, \tag{2.3}$$

where a random Gaussian sensing matrix $\mathbf{\Phi}$ with zero mean and variance $1/N$ and the inverse DCT transform matrix $\mathbf{A}$ have been used to obtain $\mathbf{B} = \mathbf{\Phi A}$.

Assuming for now that it is known which coefficients in $\mathbf{s}$ are significant, the system may be reduced to the equivalent overdetermined subsystem

$$\begin{bmatrix} -0.357 \\ 0.612 \\ 0.137 \end{bmatrix} = \begin{bmatrix} -0.293 & -0.127 \\ 0.088 & 1.048 \\ -0.069 & 0.412 \end{bmatrix} \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} \tag{2.4}$$

in the form $\mathbf{y} = \mathbf{B}'\mathbf{s}'$, where the columns in $\mathbf{B}$ corresponding to the non-zero elements in $\mathbf{s}$ have been extracted to give the submatrix $\mathbf{B}'$, and $\mathbf{s}'$ contains the non-zero coefficients in $\mathbf{s}$.

Under which conditions can $\mathbf{s}$ be recovered from $\mathbf{y}$? Firstly, the columns in $\mathbf{B}'$ should be linearly independent, since the coefficients in $\mathbf{s}$ will not be separable if they can be expressed as a linear combination of the vectors corresponding to the other coefficients. This condition would have sufficed if the K locations were known beforehand. However, they are not. Out of the infinite number of possible solutions, the solution $\mathbf{s}$ with the least number of significant coefficients will be chosen, which will in turn give the locations of the K non-zero coefficients. However, it must be ensured that there is only one K-sparse solution to choose—a unique solution.

Suppose for the sake of contradiction that two distinct K-sparse solutions $\mathbf{s}$ and $\hat{\mathbf{s}}$ exist, such that $\mathbf{y} = \mathbf{Bs} = \hat{\mathbf{y}} = \mathbf{B}\hat{\mathbf{s}}$, or $\mathbf{B}(\mathbf{s} - \hat{\mathbf{s}}) = \mathbf{0}$. Then the difference vector $\boldsymbol{\delta} = \mathbf{s} - \hat{\mathbf{s}}$ is at most $2K$-sparse [30]. By definition, the equivalent subsystem $\mathbf{B}'\boldsymbol{\delta}' = \mathbf{0}$ has a non-trivial solution if and only if the $2K$ or less columns in $\mathbf{B}'$ are linearly dependent. Conversely, if these columns are linearly independent, $\mathbf{B}(\mathbf{s} - \hat{\mathbf{s}}) = \mathbf{0}$ cannot be satisfied and no more than one K-sparse solution can exist. To guarantee a unique solution for any combination of K or less significant coefficients in $\mathbf{s}$, all subsets of $2K$ columns in $\mathbf{B}$ must be linearly independent [30].

The restricted isometry property (RIP) goes a step further and considers whether submatrices with $2K$ columns are nearly linearly dependent by placing bounds on the conditioning of these submatrices [31].

This ensures robustness in the presence of noise, since small perturbations may produce large errors when solving a nearly linearly dependent system.

Another common criterion for evaluating compresssed transform matrices is to consider the coherence of $\mathbf{B}$. (Here the similarity between vectors within a single set of non-orthogonal vectors is considered rather than the similarity between two orthonormal bases.) Since there are more columns that rows, the columns cannot be orthogonal; however, the coherence quantifies how closely $\mathbf{B}$ resembles an orthonormal set. For example, if the columns in $\mathbf{B}'$ in (2.4) are too similar, it will be difficult distinguish between them in order to recover the elements in $\mathbf{s}'$. The coherence of a matrix with normalised columns is calculated as the maximum absolute dot product $\max_{l \neq n} |\mathbf{b}_l^{\mathrm{H}} \mathbf{b}_n|$ for all $l, n \in \{1, \ldots, N\}$ (Section 3.4) [32].

Compressed transform matrices $\mathbf{B}$ with independent and identically distributed (i.i.d.) random entries have been shown to meet the RIP criterion with high probability. Examples include Gaussian matrices with zero mean and variance $1/M$, and Bernoulli matrices with equiprobable $\pm 1/\sqrt{M}$ entries [13]. For practical implementations $\mathbf{\Phi}$ may be chosen in the same way as $\mathbf{B}$ (replacing M with N), and the resulting $\mathbf{B}$ will still be able to meet the criteria for arbitrary choice of $\mathbf{A}$ [13]. As a result, the incoherence requirement between $\mathbf{\Phi}$ and $\mathbf{A}$ will also be satisfied.

2.7 MINIMISING SPARSITY—PERFECT RECOVERY ALGORITHMS

So far, it has been shown how it is possible for a CS system to preserve the information in $\mathbf{x}$ from only the measurements $\mathbf{y}$. But how is $\mathbf{x}$ recovered from $\mathbf{y}$?

In developing a CS recovery algorithm, the aim is to obtain a unique solution to an underdetermined system that is the sparsest of all solutions. This can be formulated as

$$\min \|\tilde{\mathbf{s}}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{B}\tilde{\mathbf{s}}, \tag{2.5}$$

where $\|\cdot\|_0$ is the ℓ_0-norm defined as the number of non-zero elements in a vector. The problem reads: 'Minimise the number of non-zero elements in $\tilde{\mathbf{s}}$, subject to $\tilde{\mathbf{s}}$ being a possible solution to the system.' The general ℓ_p-norm is defined as $\|\mathbf{s}\|_p = \sqrt[p]{\sum_{n=1}^{N} |s_n|^p}$.

To illustrate, consider the following combinatorial ℓ_0 minimisation algorithm:

for $k \leftarrow 1$ **to** $M - 1$
  for all combinations of k out of N coefficient locations in $\tilde{\mathbf{s}}$
    Find the least-squares solution $\hat{\mathbf{s}}'$ to $\mathbf{y} = \mathbf{B}'\tilde{\mathbf{s}}'$
    if $\hat{\mathbf{y}} = \mathbf{B}'\hat{\mathbf{s}}' = \mathbf{y} \longrightarrow$ **break**

This algorithm tries all possible combinations of sparse coefficient locations, starting with a single one, until it finds an exact least-squares solution to the subsystem.

Consider the system in (2.3) with $\mathbf{s}$ unknown. The first try would assume that only s_1 is significant and solve the resulting subsystem, giving the least-squares solution $s_1 = -0.062$. This gives $\hat{\mathbf{y}} = [0.064, 0.052, -0.015]^{\mathrm{T}} \neq \mathbf{y}$, which is not an exact solution.

Eventually the combination (s_2, s_5) is reached, which leads to the subsystem in (2.4) with $\mathbf{s}'$ unknown, for which the least-squares solution is $(s_2, s_5) = (1.0, 0.5)$, giving $\hat{\mathbf{y}} = [-0.357, 0.612, 0.137]^{\mathrm{T}} = \mathbf{y}$, an exact solution. This also happens to be the first combination that gives an exact solution, as expected. While there are many more solutions, the algorithm accepts this one as correct since it is sparsest. If desired, it is now possible to calculate $\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{s}}$.

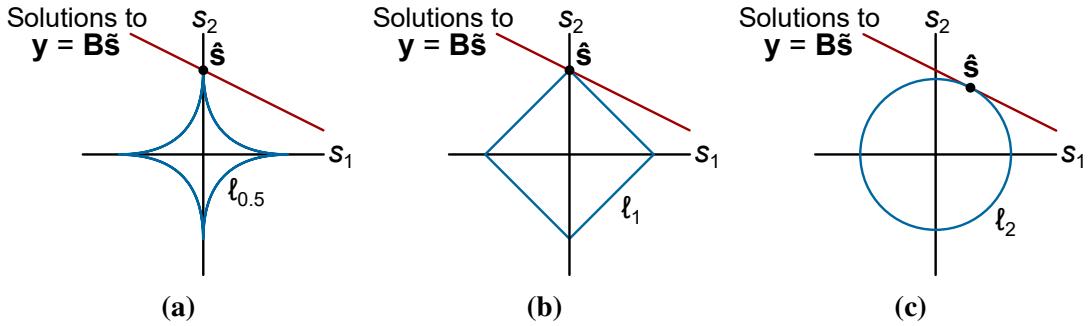


Figure 2.4. Illustration of ℓ_p-norms for (a) $p = 0.5$, (b) $p = 1$, and $p = 2$ (Euclidean norm). Blue lines occur where the vector $\tilde{\mathbf{s}}$ has constant ℓ_p-norm. Any point on a red line is a possible solution. Adapted from [25], ©2017 IEEE.

Unfortunately, a combinatorial ℓ_0 algorithm is computationally infeasible for problems of practical sizes [12]. For a practically feasible recovery algorithm, the ℓ_p-norm formulation

$$\min \|\tilde{\mathbf{s}}\|_p \quad \text{subject to} \quad \mathbf{y} = \mathbf{B}\tilde{\mathbf{s}}, \quad 0 < p \leq 1 \tag{2.6}$$

is preferred since it can be solved using a variety of efficient optimisation algorithms [33]. The ℓ_1 problem can be recast as a linear program [30, 32], and requires around $O\{K \log(N/K)\}$ measurements when using a random sensing matrix [13]. For $0 < p < 1$ the problem is non-convex with multiple minima, but local optimisers nevertheless perform well, particularly when $p = 0.5$ [33].

To visualise why the ℓ_p-norm favours sparsity when $0 < p \leq 1$, consider the graphs in Figure 2.4 for $p = 0.5$, $p = 1$, and $p = 2$. The ℓ_p-balls shown in blue represent the points at which $\tilde{\mathbf{s}}$ has constant ℓ_p-norm. Since the norms must be minimised, imagine the balls being inflated until they first touch the lines of possible solutions shown in red. For $0 < p \leq 1$, the ℓ_p-norms favour sparse solutions that lie on the axes. By comparison, minimising the common Euclidean norm ($p = 2$) is not useful since it does not obtain a solution that is necessarily sparse.

2.8 CHAPTER SUMMARY

In this chapter, it was shown how CS exploits redundancy to reduce sampling rates, and under which conditions the original signal is preserved despite the system being underdetermined. It was shown how universal sampling schemes requiring the existence but not knowledge of a sparsifying basis may be developed by using random sensing matrices. Lastly, a simple ℓ_0-minimisation algorithm was shown to recover a signal with knowledge of the sparsifying basis, and the rationale behind ℓ_p-minimisation algorithms was considered for $0 < p \leq 1$.

CHAPTER 3 TOWARDS APPLYING COMPRESSIVE SENSING TO REDUCED-CONTROL ARRAY DESIGN

3.1 CHAPTER OVERVIEW

This chapter provides an overview of existing reduced-control techniques and codebook optimisation algorithms which sets the stage for applying the concept of CS to antenna arrays.

Section 3.2 describes conventional beamforming arrays, while Section 3.3 provides a survey of existing reduced-control arrays. The concept of reducing the number of beamforming controls is motivated in Section 3.3.1. Thinned and sparse arrays are considered in Section 3.3.2 and subarrays are described in Section 3.3.3, which include the dual-transform and chess networks. Existing techniques for implementing compressive arrays are described in Section 3.3.4, where limitations of the current techniques are also highlighted. The most promising starting point for the development of a compressive array optimisation algorithm, discrete codebook optimisation, is discussed in Section 3.4. The concept of coherence is revisited in Section 3.4.1 and various codebook optimisation algorithms are listed in Section 3.4.2. The sequential approach towards codebook optimisation is discussed in some detail in Section 3.4.3. Finally, the chapter is summarised in Section 3.5.

3.2 CONVENTIONAL BEAMFORMING ANTENNA ARRAYS

Consider an array of N antenna elements placed at arbitrary locations in two-dimensional space. Define the steering vector as the baseband array response in square-root power to a unit-amplitude continuous-wave signal impinging on the array from the direction θ in the plane of the array, denoted $\mathbf{a}(\theta)$.[1] The steering vector includes the effect of the element patterns and can either be formulated mathematically, determined by simulation, or measured practically. In the case of a ULA with isotropic elements, the $N\times 1$ steering vector is given by [5]

$$\mathbf{a}(\theta) = \left[\mathrm{e}^{\mathrm{j}2\pi d_1 \sin(\theta)/\lambda}, \cdots, \mathrm{e}^{\mathrm{j}2\pi d_N \sin(\theta)/\lambda} \right]^{\mathrm{T}}, \tag{3.1}$$

where θ is the clockwise incident angle of the wave relative to broadside, d_n is the distance of element n to a reference point on the array, and λ is the wavelength.[2] The geometry associated with an N-element ULA is illustrated in Figure 3.1(a). A circular array with radius R and isotropic elements on the

[1]The techniques described are applicable in the general case with suitable alterations, but only the one-dimensional case is considered to avoid unnecessary notational complexity.

[2]It is worth noting that since the analyses are phase-based, the results are inherently narrowband. However, most antenna-array analyses are formulated in this way, so this is not considered a limitation of the results obtained.

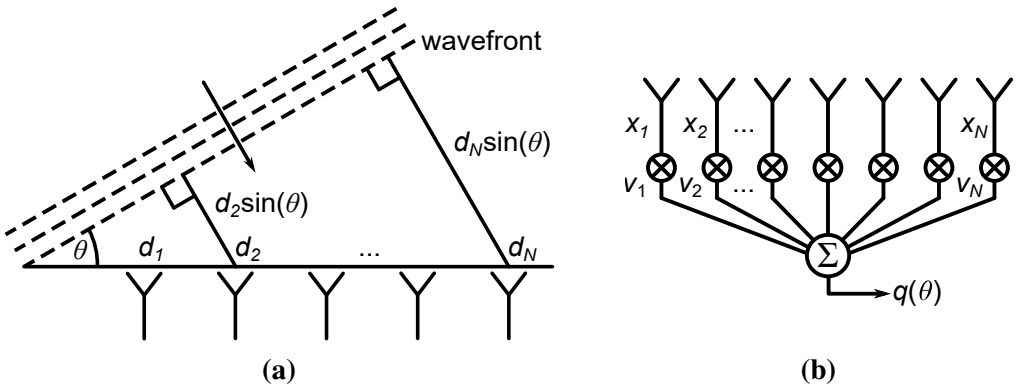


Figure 3.1. (a) An N-element ULA receiving a signal from the direction θ. (b) An N-element ULA with beamforming weights, shown in reception; taken from [1], ©2018 IEEE.

circumference of the circle has a steering vector given by [5]

$$\mathbf{a}(\theta) = \left[\mathrm{e}^{\mathrm{j}2\pi R \sin(\theta-\theta_1)/\lambda}, \cdots, \mathrm{e}^{\mathrm{j}2\pi R \sin(\theta-\theta_N)/\lambda} \right]^{\mathrm{T}}, \tag{3.2}$$

where θ_n is the clockwise angle of element n around the circle. The steering of circular arrays will only be considered in the plane of the array. In the remainder of the thesis, steering vectors are assumed to be normalised to have unit length.

Beamforming arrays apply a complex-valued weight v_n to each antenna element before summing the resulting signals, as illustrated in Figure 3.1(b) for reception [15]. On reception, the beamforming weights may be applied in hardware or in software. One advantage of applying the weights in software is that independent beams may be formed in multiple directions simultaneously [6]. On transmission, the complex-valued weights are applied as excitations at the antenna elements to perform beamforming. The resulting array radiation pattern in square-root power is given by [15]

$$q(\theta) = \left| \mathbf{v}(\theta_s)^{\mathrm{T}} \mathbf{a}(\theta) \right|, \tag{3.3}$$

where $\mathbf{v}(\theta_s)$ is the vector of beamforming weights for the desired steering direction θ_s. Generally, the phases of the beamforming weights are chosen to be the negatives of the phases in $\mathbf{a}(\theta_s)$ so that the signals from the direction θ_s add in-phase. The amplitudes of the beamforming weights are chosen to realise a particular array amplitude pattern, which involves a compromise between directivity, beamwidth and SLL [2, 34].

The distribution of the currents (with magnitudes and phases) across the elements of an array is referred to as the aperture illumination. This is best understood in terms of a transmit array. In a conventional ULA, the currents at the antenna elements are the excitation weights in $\mathbf{v}(\theta_s)$ (before modulating the excitations with a data signal), such that $\mathbf{v}(\theta_s)$ is the aperture illumination. In the case of more complex feed networks such as overlapped subarrays, the definition remains the same. However, since subarrays are effectively individual arrays that happen to share the same aperture with other subarrays, it is useful to define an aperture illumination for each subarray separately. In this way, a stimulus to a single subarray port will realise a particular aperture illumination at the antenna elements. The individual subarray aperture illuminations then add to form the total aperture illumination, which in turn determines the radiation pattern of the array as a whole.

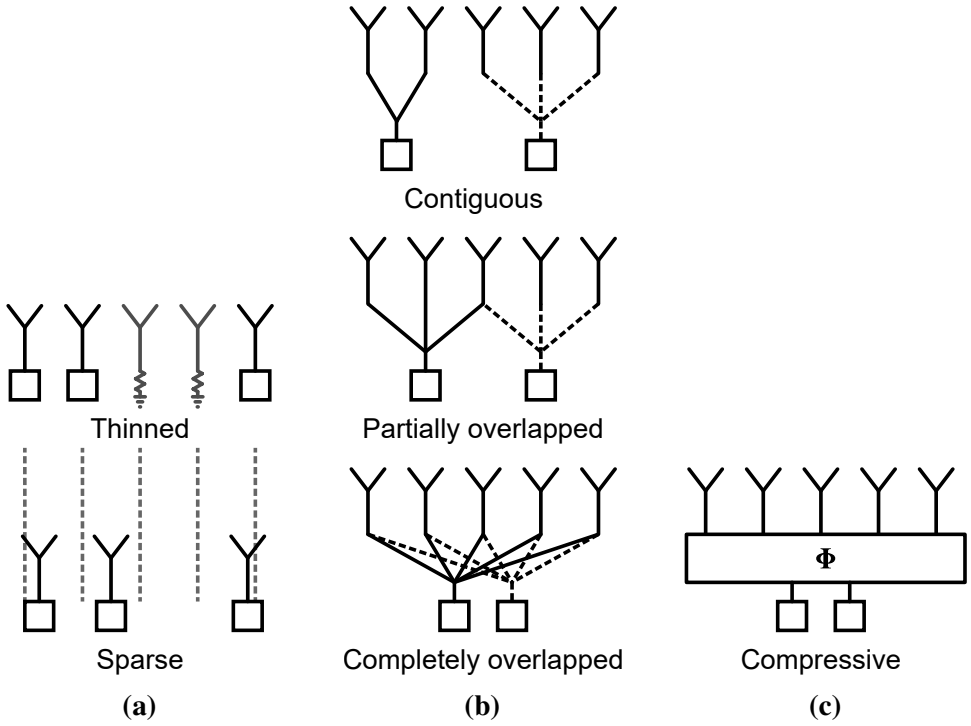


Figure 3.2. Illustrations of various reduced-control arrays. Squares represent beamforming controls. (a) Thinned arrays disable some elements in a ULA, whereas the elements in a sparse array are not restricted to a uniform grid. (b) Subarrays share none, some, or all of the antenna elements with adjacent subarrays. (c) Compressive arrays are a generalisation of all reduced-control arrays with a fixed antenna-element layout.

3.3 REDUCED-CONTROL ARRAYS

3.3.1 Motivation

As discussed in Section 1.1.1, there is a need for reduced-control arrays that utilise less beamforming controls than a conventional array with uniform spacing between the antenna elements. Reduced-control arrays offer reduced cost, size and/or weight compared to conventional arrays [3, 7–9].

Reduced-control arrays can broadly be categorised as follows and as illustrated in Figure 3.2.

Thinned and sparse arrays place a reduced number of elements non-uniformly in a given array aperture [2, 8–10, 29, 35–38].

Conventional subarrays combine antenna elements into subarrays for a limited steering range or for wideband applications [2, 3, 7, 39, 40].

Compressive arrays are a generalisation of conventional reduced-control arrays that utilise subarrays that are each a function of all the antenna elements, and are not necessarily restricted to a limited steering range [11, 14].

In terms of cost, the beamforming controls, which require active radio-frequency (RF) components, are usually the limiting factor and not the passive antenna elements [3]. This is especially true in the case of digital arrays which require an RF frontend at each antenna element. While thinned and sparse arrays reduce both the number of elements and controls, subarrays and compressive arrays reduce the number

of controls whilst utilising a filled array of elements. This means that subarrays and compressive arrays offer more degrees of freedom than thinned/sparse arrays for similar cost at the expense of a more complex feed network [2, 39, 40]. Compared to ULAs, overlapped subarrays and compressive arrays primarily offer savings in terms of cost, but with more degrees of freedom than other reduced-control techniques.

3.3.2 Thinned and sparse arrays

Thinned and sparse arrays are similar in that they both place a reduced number of antenna elements non-uniformly in a given aperture, the difference being that thinned arrays have their elements placed on the underlying grid of a filled array, as illustrated in Figure 3.2(a) [41]. Thinned arrays can, therefore, be implemented by taking a filled array and either disabling or removing some of the elements. Disabling elements (for example, by terminating them in matched loads) has the advantage that the mutual electromagnetic coupling between adjacent elements remains similar across all but the extremes of the array [38]. This leads to the elements having similar radiation patterns which is useful since the assumption of identical element patterns is still widely used in array design [2]. Allowing arbitrary element positions enables lower SLLs to be achieved, but limits the useful steering range of sparse arrays, whereas thinned arrays with half-wavelength element spacing do not have this limitation [38].

When the average inter-element spacing is large, the directivity of a thinned or sparse array is mainly a function of the number of active elements [2, 10]. The achievable SLL is also a function of the number of active elements [10]. The beamwidth and resolution of the array is mainly a function of the array aperture dimensions [10]. As an example, a uniform-excitation M-element thinned/sparse array will have a resolution similar to that of a uniform-excitation N-element ULA and a directivity similar to that of a uniform-excitation M-element ULA. Because the directivity is similar to that of the M-element ULA but the beamwidth is reduced, the SLL of the thinned/sparse array will be higher than that of either ULAs.

An additional degree of freedom that can be exploited is to apply weights to the active elements, allowing for finer control over the array patterns [38].

Many techniques for finding optimal element locations exist. For example, elements may be placed according to some probabilistic distribution [8, 35]. Sparse arrays may be made up of elements placed by sampling a random distribution, where the shape of the distribution influences the average expected array patterns [8]. A similar technique for thinned arrays is that of density tapering, where a binary decision is made for each grid point to either place or not place an element according to a pre-defined random distribution [35].

Thinned arrays may also be numerically optimised using algorithms such as genetic algorithms [36] and inverse fast Fourier transform (IFFT) techniques [9]. Weighted thinned arrays have also been designed using a modified form of the IFFT technique [38].

The use of optimisation algorithms for designing large thinned arrays is challenging due to the computational complexity of the algorithms [41]. An alternative for large arrays is the use of difference sets and almost difference sets, analytical binary sets that determine which elements are active [41–43]. Difference and almost difference sets have allowed very large linear and planar arrays to be designed without optimisation and with predictable performance. Almost difference sets have also been used for seeding genetic algorithms, leading to better thinned arrays than either technique could offer alone [44].

Many different methods exist for optimising sparse arrays [41]. A notable algorithm for optimising sparse arrays is based on the matrix pencil method [45].

Recently, CS reconstruction algorithms have been applied to the design of sparse arrays [29, 41]. The system to be solved is given by $\mathbf{q} = \mathbf{A}^{\mathrm{T}}\mathbf{v}$, where $\mathbf{q}$ is the desired complex-valued array pattern sampled at various angles, $\mathbf{A}$ is the steering matrix with steering vectors as columns, and $\mathbf{v}$ is the vector of beamforming weights to be solved for. Using a reconstruction algorithm that favours sparse solutions, the aim is to find a set of beamforming weights $\mathbf{v}$ that has only few non-zero components. Note that this is not a compressive array, but a conventional sparse array designed using sparsity-promoting algorithms.

In addition to CS-based approaches that define a pattern-matching problem as above, CS-based methods have also been developed where a maximum radiation-pattern mask can be specified instead of an exact, predefined pattern [41, 46, 47].

3.3.3 Subarrays for limited steering ranges

In contrast to thinned and sparse arrays, conventional subarrays reduce the number of controls while maintaining a filled array of elements. This is achieved by weighting and combining the antenna-element signals so that each combination forms a subarray. The elements can be shared between subarrays, leading to overlapped subarrays. The amount of overlap is used in the following classification, as illustrated in Figure 3.2(b) [7].

Contiguous subarrays do not share any antenna elements, and are effectively ULAs of potentially varying sizes placed next to each other. Each subarray is a combination of the antenna elements unique to that subarray [3, 7].

Partially overlapped subarrays share some, but not all, of the antenna elements with adjacent subarrays [2, 3, 39, 40]. Typically, adjacent subarrays overlap each other by a fixed number of elements across the array.

Completely overlapped subarrays share the entire array aperture so that each subarray is a function of all the antenna-element signals [2, 39]. Compressive arrays such as the one proposed for DF [11, 14] are also completely overlapped systems, but will be dealt with separately in Section 3.3.4.

Subarray systems can be seen as an array of elements with controlled element patterns, where each element is itself a subarray of basic antenna elements [40]. The spacing between the basic antenna elements is small, for example half a wavelength, but the spacing between the phase centres of the subarrays is large, up to several wavelengths [7]. The large spacing between the phase centres of the subarrays may cause grating lobes in the array factor. The techniques for overcoming this challenge fall into two main categories [39]. Some approaches suppress the grating lobes via irregular spacing of the subarray phase centres, similar to the way thinned and sparse arrays are able to avoid ambiguities due to the irregular element spacing [39, 48, 49]. Other approaches retain the regular spacing of the subarrays, but design the subarray patterns to be low outside the steering range, thereby suppressing the grating lobes. The result is that high-directivity beams with low SLL can be steered across a small angular range [2, 3, 7].

Various feed network layouts exist for implementing overlapped feed networks using couplers, crossovers, lenses, and/or reflectors [2, 39, 40]. Overlapped feed networks have been manufactured using striplines [50, 51], waveguides [40], lenses [52], and radio frequency integrated circuits (RFICs) [53].

Contiguous subarrays may have uniform or non-uniform illuminations across the elements unique to each subarray [41, 54, 55]. The design of contiguous subarrays involves selecting the elements to allocate to each subarray, the amplitude weights (if any) applied to the elements in each subarray, and the excitations applied to the subarrays [41]. A ULA may be divided into contiguous subarrays

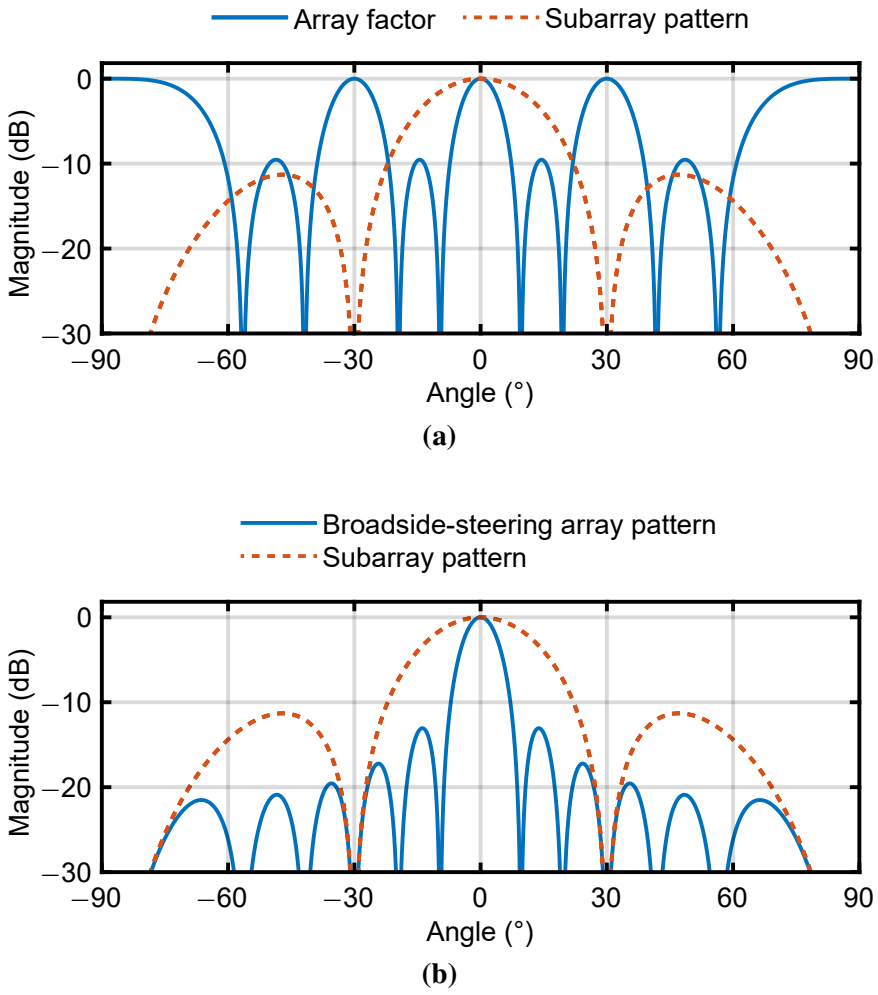


Figure 3.3. A contiguous subarray system with $N = 12$ and $M = 3$: (a) The array factor and subarray amplitude pattern. (b) The broadside-steering pattern and subarray amplitude pattern.

of random sizes [49]. One contiguous-subarray technique for planar arrays involves dividing the elements into identical L-shaped subarrays, with the L shapes rotated and fit together in a tiled manner [48]. Pattern-matching design methods such as the contiguous partition method aim to realise a reference pattern corresponding to a reference aperture illumination across the entire array [56]. The total aperture illumination achieved by the subarrays is then a quantised version of the reference illumination [39, 54].

Consider a contiguous subarray system with $N = 12$ antenna elements with half-wavelength spacing grouped into $M = 3$ groups of four antennas each [7]. The spacing between the elements is half a wavelength, so the spacing between the subarrays is $0.5 \times 4 = 2$ wavelengths. Figure 3.3(a) shows the array factor for an array of three subarrays, as well as the amplitude pattern of the subarrays. The subarray outputs are combined to perform beamforming. Figure 3.3(b) shows the array pattern when steered to broadside. Notice how the subarray pattern suppresses grating lobes in the array factor.

As illustrated above, conventional subarray systems with regularly-spaced subarrays aim to realise subarray patterns that are high in the steering range, and low elsewhere to suppress grating lobes in the array factor. The ideal subarray pattern is unity in the steering range and zero elsewhere [2].

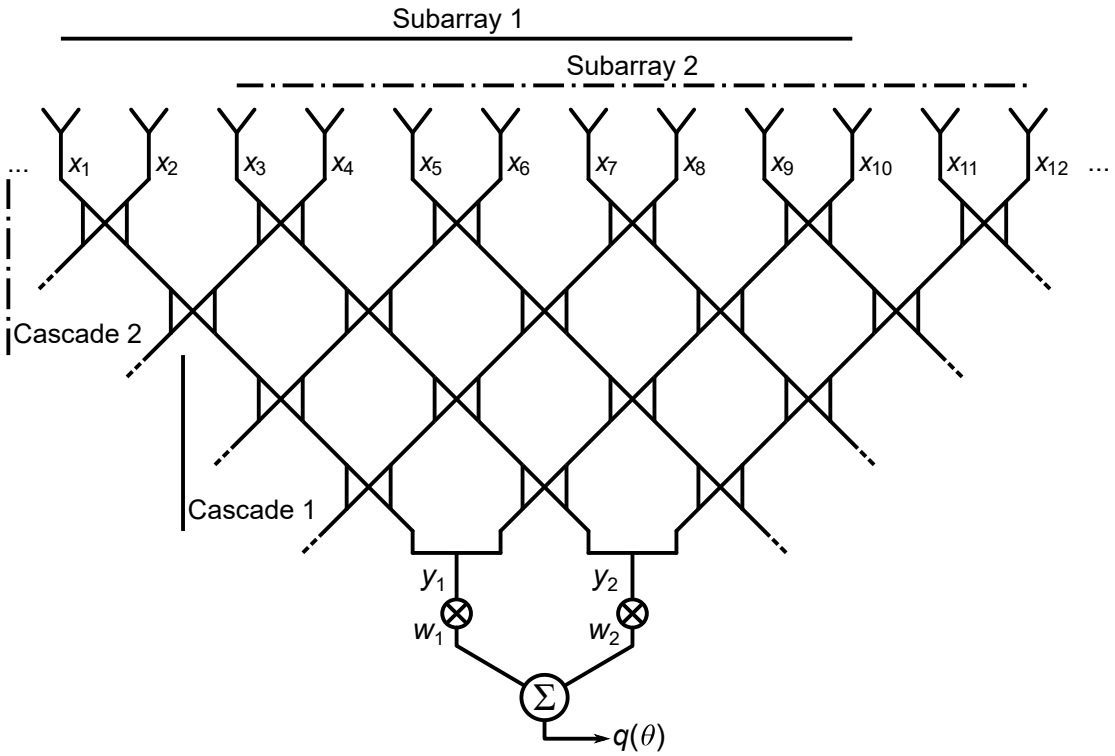


Figure 3.4. Illustration of the Skobelev chess network as described in [40, 57] with two cascades of couplers for $N = 12$ and $M = 2$. Each subarray spans 10 antenna elements.

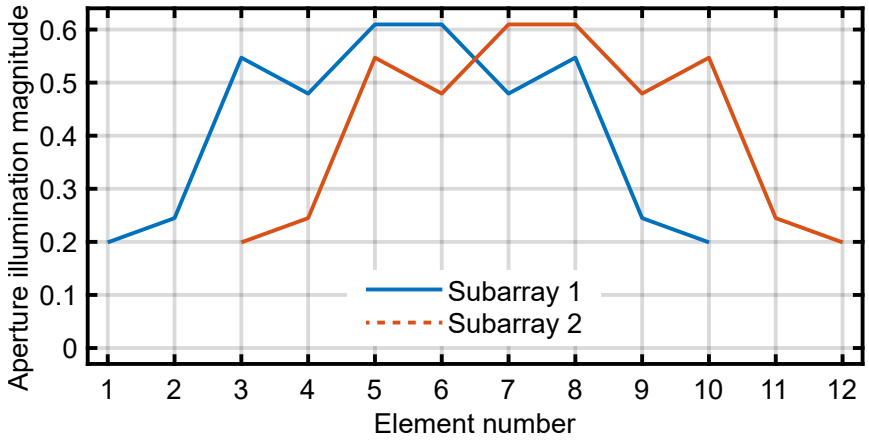
Different types of subarrays use different techniques to realise subarray patterns that approximate the ideal flat-topped subarray pattern.

The Skobelev chess network [40, 57–59] is an example of a partially overlapped subarray system, and is illustrated in Figure 3.4 for $N = 12$ antenna elements and $M = 2$ subarrays. In this example, each subarray spans ten elements. The network consists of cascades of directional couplers, where each cascade consists of two rows of interconnected couplers. The couplers in each row are identical. The outputs of the lowest cascade are connected to T-section combiners [60] to provide the M subarray outputs. The greater the number of cascades, the greater the degree of overlap between adjacent subarrays. The Skobelev chess network has the advantage that it does not require any physical crossovers and can thus be implemented compactly in a planar layout.

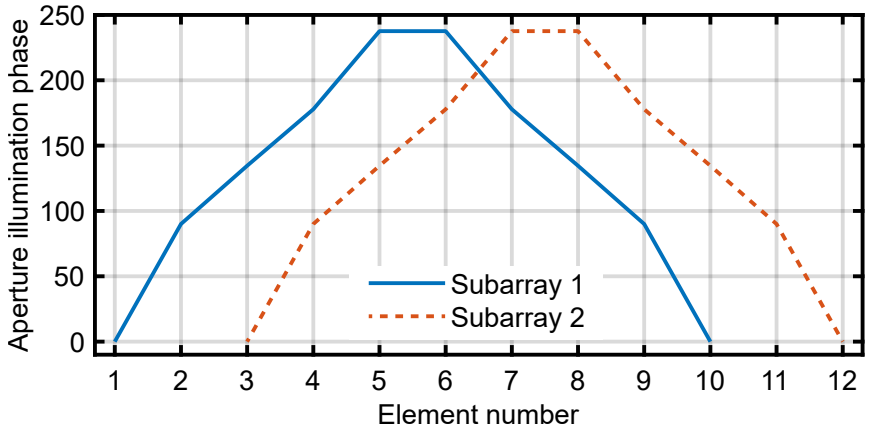
Figures 3.5(a) and (b) show the aperture illumination magnitudes and phases, respectively, for a Skobelev chess network with $N = 12$ elements with half-wavelength spacing, $M = 2$, and two cascades [40]. The resulting subarray pattern, equivalent for both subarrays, is shown in Figure 3.5(c).

A single-cascade, K_a-band Skobelev chess network has successfully been implemented in waveguide [40].

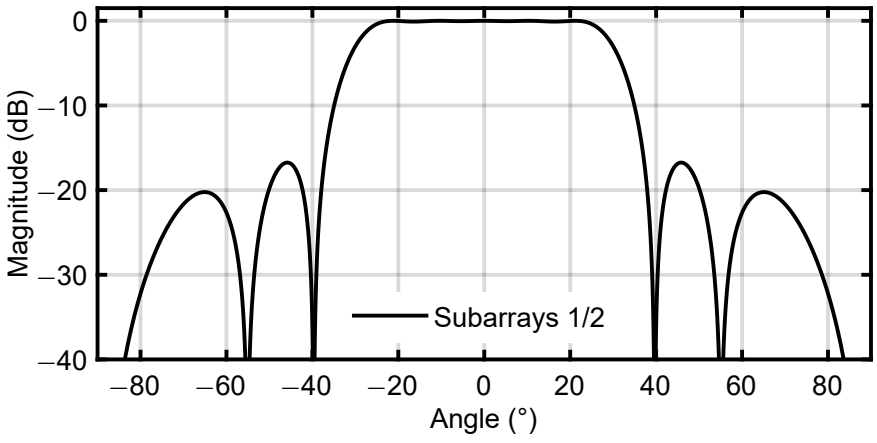
A dual-transform subarray is an example of a completely overlapped subarray [2, 39, 40]. A dual-transform system consists of two multiple-beam networks in cascade, as illustrated in Figure 3.6. Each input to a multiple-beam network results in a beam that points in a unique direction [2, 61]. In a dual-transform array, each multiple-beam network realises the discrete Fourier transform (DFT), and can be implemented as a Bulter matrix [61], lens, reflector, or can be implemented digitally [39]. A larger $N \times N$ multiple-beam network has its M outputs corresponding to the beams closest to broadside connected to a smaller $M \times M$ multiple-beam network. Each output of the smaller network corresponds



(a)



(b)



(c)

Figure 3.5. A Skobelev chess network with $N = 12$, $M = 2$ and two cascades: (a) Aperture illumination magnitudes. (b) Aperture illumination phases. (c) Subarray patterns.

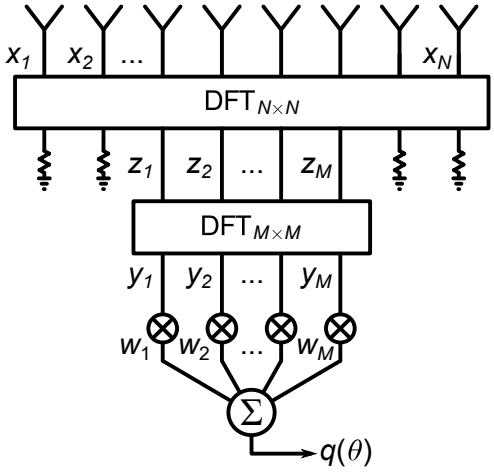


Figure 3.6. Illustration of a dual-transform completely overlapped subarray system as described in [2].

to a subarray. Since the $M \times M$ network has the same number of inputs and outputs, it is often implemented digitally [39].

Progressive phases are applied to the subarray ports to steer the beam, and an amplitude taper can also be applied to the subarray ports to improve the SLL in the steered array pattern [2]. For example, a standard length-M Chebyshev amplitude taper can be applied to the subarray ports [2].

Consider a Butler-matrix-based dual-transform array. Ignoring the terminated ports, the response of the upper $M \times N$ matrix $\mathbf{D}^{M\times N}$ is given by [2]

$$d_{m',n}^{M\times N} = \frac{1}{\sqrt{N}} \mathrm{e}^{\frac{-\mathrm{j}2\pi}{N}\left(m'-\frac{M+1}{2}\right)\left(n-\frac{N+1}{2}\right)}, \tag{3.4}$$

where m' refers to the index of an element in the $M \times 1$ intermediate signal vector $\mathbf{z}$ (see Figure 3.6) so that

$$\mathbf{z} = \mathbf{D}^{M\times N}\mathbf{x}. \tag{3.5}$$

The response of the lower $M \times M$ matrix $\mathbf{D}^{M\times M}$ is given by

$$d_{m,m'}^{M\times M} = \frac{1}{\sqrt{M}} \mathrm{e}^{\frac{\mathrm{j}2\pi}{M}\left(m-\frac{M+1}{2}\right)\left(m'-\frac{M+1}{2}\right)}, \tag{3.6}$$

so that

$$\mathbf{y} = \mathbf{D}^{M\times M}\mathbf{z}. \tag{3.7}$$

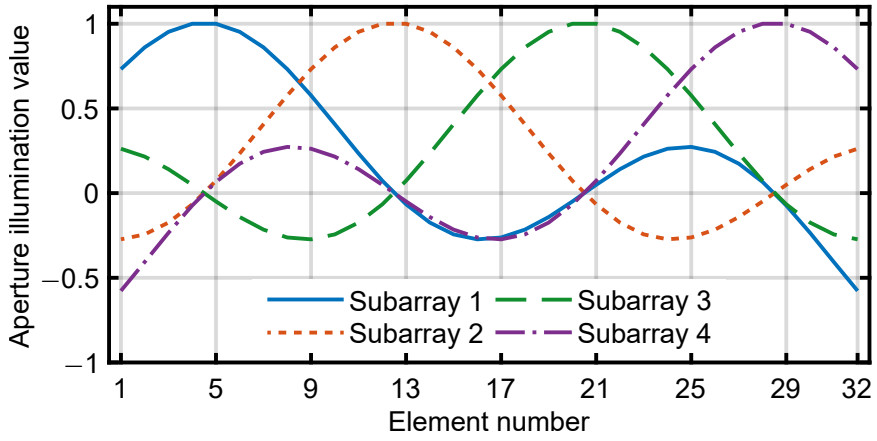
The exponents in (3.4) and (3.6) have opposite signs since one of the Butler matrices should implement the DFT, and the other the inverse discrete Fourier transform (IDFT). The one matrix functions as a multiple-beam network, and the other is reversed to function as a beam-focusing network [2].

Combining (3.5) and (3.7), the system response can be described by

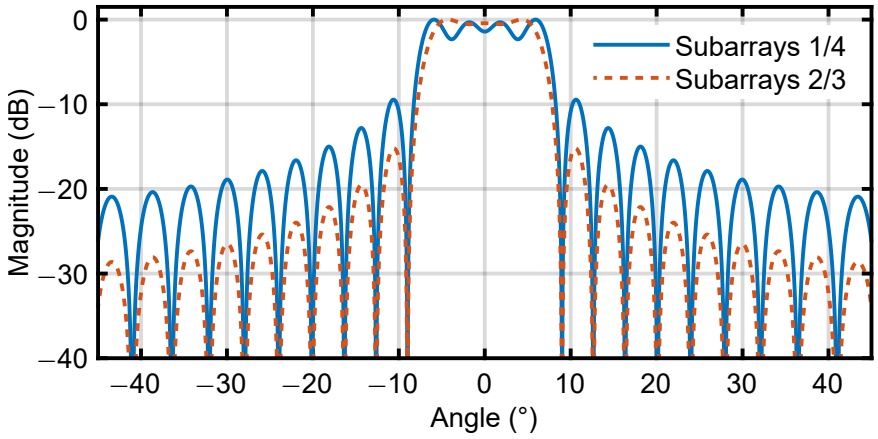
$$\mathbf{y} = \mathbf{D}^{M\times M}\mathbf{z} = \mathbf{D}^{M\times M}\mathbf{D}^{M\times N}\mathbf{x} = \mathbf{\Phi}\mathbf{x}, \tag{3.8}$$

where $\mathbf{\Phi}$ is the $M \times N$ sensing matrix that describes the response of the complete feed network.

Consider a dual-transform subarray system with $N = 32$ elements with half-wavelength spacing and $M = 4$ subarrays. The sinc-shaped aperture illuminations corresponding to the subarrays are shown in Figure 3.7(a). The corresponding subarray amplitude patterns are plotted in Figure 3.7(b). The outer



(a)



(b)

Figure 3.7. A dual-transform subarray system with $N = 32$ and $M = 4$: (a) Aperture illuminations. (b) Subarray patterns.

subarrays have larger ripple in the steering range and higher SLLs outside the steering range due to the truncation of the sinc illuminations at the edges of the array. The maximum theoretical steering angle [2] is ±5.4°.

For practical dual-transform systems, lenses and reflectors are often preferred due to the complexity and physical size of Butler matrices for arrays of even moderate sizes [40]. A large contributor towards the complexity of Butler matrices is the large number of crossovers required [62, 63]. This makes the use of lenses and reflectors cost-effective, but also restricts the main application of the dual-transform technique to physically large arrays [39]. An example of a dual-transform array that has successfully been manufactured using lenses is the array presented in [52]. There, a cylindrical bootlace lens implements the larger multiple-beam matrix, and a Rotman lens implements the smaller multiple-beam matrix.

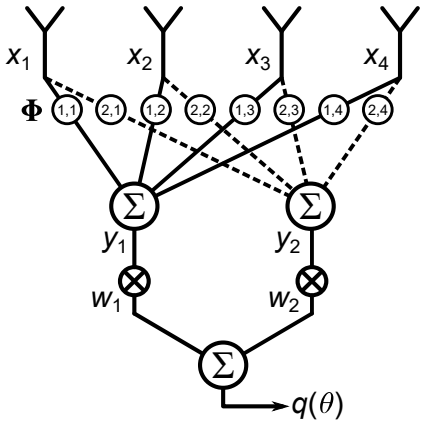


Figure 3.8. Illustration of a compressive array with $N = 4$ antenna elements and $M = 2$ subarrays.

3.3.4 Compressive antenna arrays

Inspired by the concept of CS (Chapter 2), a completely overlapped compressive DF array has previously been proposed [11, 14]. This compressive array uses a random sensing matrix $\mathbf{\Phi}$ to connect each subarray to all the antenna elements [11]. The random sensing matrix was originally proposed to be real-valued [11], but recently, a compressive DF array has been designed with a random complex-valued sensing matrix [14]. The feed network configuration for a compressive array is illustrated in Figure 3.8 for $N = 4$ antenna elements and $M = 2$ subarrays.

CS predicts that if a sampled signal is sparse in some domain, the number of samples can be reduced without any loss of information, or with limited loss of information in the case of noisy systems (Chapter 2). A DF ULA with one receiver per antenna element must have no more than half-wavelength spacing between the elements if the full field of view is to be scanned. Good DF resolution requires a large array aperture. The need for half-wavelength spacing and a large aperture can quickly lead to a large number of receivers being required. However, the number of DoAs is typically much smaller than the number of antenna elements, implying a high level of redundancy. Typical signals from DF arrays are, therefore, sparse in DoA.

The compressive array is unique in that the sensing matrix is not designed with the hardware implementation in mind as all other reduced-control arrays are. The lack of constraints on the sensing matrix implies that a hardware architecture for such arrays would need to be able to implement arbitrary sensing matrices.

In [11], the use of attenuators is suggested as a possibility for implementing the compression weights, or the use of 180° phase shifters if the sensing matrix is restricted to ±1 entries.

Recently, a practical design for a compressive DF array at 868 MHz with $N = 4$ unequally-spaced antenna elements and $M = 2$ subarrays was presented [14]. Each antenna element was connected to a two-way splitter, of which each output was connected to a four-way combiner to produce the subarray outputs. Thus, four two-way splitters and two four-way combiners were used. The splitters and combiners were implemented as lumped-element Wilkinson power dividers [60]. The lumped-component values of the splitters were modified to introduce random magnitude and phase variations in the signals, thereby aiming to implement a random complex-valued sensing matrix. The splitters and combiners were manufactured on separate boards and connected using cables. The compressive array was able to resolve some DoAs correctly, but had ambiguities at certain steering angles. The steering angles with ambiguities were found to correspond to the steering vectors that had small Euclidean

distances to some of the other steering vectors [14]. This implies that the coherence of the compressed transform matrix (Section 2.6) was high. The issue of coherence and its relationship to Euclidean distance will be dealt with in Section 3.4.

The drawback of the hardware configuration in [14] is the large number of crossovers required. If cables are used to implement the crossovers as was done in [14], then MN cables are required in total. This number would soon become enormous. A small array of, say, $N = 8$ and $M = 4$, would already require $8 \times 4 = 32$ cables connecting $8 + 4 = 12$ separate boards. This technique was developed to implement a random sensing matrix, but to implement an arbitrary pre-defined sensing matrix, phase-matched cables would be required. These limitations make this approach unsuitable for arrays of practical sizes [64].

3.4 DISCRETE CODEBOOK OPTIMISATION

3.4.1 Coherence

Coherence, the largest similarity between vectors in a codebook, is a widely-used metric for evaluating compressed transform matrices in CS systems (Section 2.6). In [14], the design and construction of a compressive DF array showed that the random sensing matrix that was implemented was not optimal, and it was concluded that some form of optimisation of the sensing matrix would have to be performed. Although the sensing matrix is what is implemented in hardware, it is the compressed transform matrix that must be designed for low coherence, which in turn will cause the incoherence requirement between the sampling and sparsifying domains to be met (Section 2.6).

The minimisation of the similarity between vectors in a codebook falls within the wider context of codebook optimisation, and in particular, Grassmannian codebook optimisation [65]. Compressed transform matrices used in CS are referred to as rank-1 codebooks because the codewords are made up of vectors and not matrices. Minimising the similarity between vectors is the same as maximising the distance between the lines spanned by the vectors [66].

Apart from their application to CS, optimised Grassmannian codebooks are also of interest in multi-antenna communication systems [66–68].

3.4.2 Codebook optimisation algorithms

The complex Grassmannian manifold $\mathbb{G}(M, R)$ is the set of all R-dimensional subspaces in $\mathbb{C}^M$. For example, the manifold $\mathbb{G}(3, 1)$ is the set of all lines in 3-dimensional complex space. Finding the best packing of subspaces involves maximising the minimum distance between a set of N subspaces [65]. A rank-R Grassmannian codebook consists of N codewords, with each codeword being an $M \times R$ matrix with orthonormal columns. Each subspace is represented by a codeword since the span of R orthogonal vectors is an R-dimensional subspace.

Various approaches towards designing Grassmannian codebooks exist [16]. In [68], optimal codebooks are constructed analytically from difference sets for certain dimensions. For other dimensions, a numerical method based on the Lloyd algorithm is proposed [68]. An alternating projection method which searches for optimal codebooks by alternately enforcing structural and spectral conditions is proposed in [69]. A sequential optimisation framework proposed in [66] formulates a minimax codebook optimisation problem which is solved by sequentially approximating the max operator by the p-norm as $p \to \infty$.

The BCASC algorithm [16] follows a similar sequential procedure to [66] to obtain rank-1 codebooks (also referred to as antipodal spherical codes), which are shown to be better than those obtained by existing approaches. The BCASC algorithm seeks to maximise the minimum Euclidean distance between codewords themselves. Rank-1 codewords represent lines, and all phase shifts of a codeword lie on the same line and are termed complex antipodals. It is therefore necessary to consider codewords and all their antipodals when computing Euclidean distances. This drastically increases the computational complexity of the problem since complex codewords have an infinite number of antipodals.

3.4.3 Sequential codebook optimisation

Consider an $M \times N$ matrix $\mathbf{B}$ representing a rank-1 codebook with N column vectors $\left\{\mathbf{b}_n = \left[b_{1,n} \cdots b_{M,n}\right]^{\mathrm{T}}\right\}_{n=1}^{N}$ as codewords. The coherence of $\mathbf{B}$ is defined as [16]

$$\mu(\mathbf{B}) = \max_{n \neq l} \frac{|\mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l|}{\|\mathbf{b}_n\| \, \|\mathbf{b}_l\|}, \tag{3.9}$$

where $n, l \in \{1, \ldots, N\}$. In CS literature, the compressed transform matrix $\mathbf{B}$ is assumed to have unit-norm columns, in which case the denominator in (3.9) is 1 and may be suppressed [32]. In the algorithms proposed in Chapters 5 and 6, a unit-norm constraint is placed on the matrices being designed for low coherence or SLL. The significance of this constraint in the context of compressive arrays is discussed in Sections 6.3.1 and 8.6.4.

The lower bound on the achievable coherence is given by [16]

$$\mu_c(M, N) = \begin{cases} \sqrt{\frac{N-M}{M(N-1)}} & \text{if } N \leq M^2, \\ \max\left(\sqrt{\frac{1}{M}}, \sqrt{\frac{2N-M^2-M}{(M+1)(N-M)}}, 1 - 2N^{\frac{-1}{M-1}}\right) & \text{if } M^2 < N \leq 2(M^2-1), \\ \max\left(\sqrt{\frac{2N-M^2-M}{(M+1)(N-M)}}, 1 - 2N^{\frac{-1}{M-1}}\right) & \text{if } 2(M^2-1) < N. \end{cases} \tag{3.10}$$

The bound for $N \leq M^2$ in (3.10) is also known as the Welch bound [16, 68].

The coherence optimisation problem is given by [16, 68]

$$\min \max_{n \neq l} \left|\mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l\right| \quad \text{subject to} \quad \|\mathbf{b}_n\|^2 = 1 \quad \forall\, n. \tag{3.11}$$

In order to obtain a smooth approximation to the max operator a series of sub-problems given by [16, 66]

$$\min \left(\sum_{n \neq l} \left|\mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l\right|^p \right)^{1/p} \quad \text{subject to} \quad \|\mathbf{b}_n\|^2 = 1 \quad \forall\, n \tag{3.12}$$

with $p \geq 1$ may be solved with increasing values of p, thereby approximating the ∞-norm by the p-norm with $p \to \infty$ [66].

Coherence is well-defined on the Grassmannian manifold and is inversely equivalent to the distance metric $\sqrt{1 - \left|\mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l\right|^2}$ common in line-packing problems [66, 67]. Minimising the coherence of a codebook is equivalent to maximising the minimum distance between the lines represented by the codewords [16].

The approach in [66] uses optimisation algorithms that are able to enforce the orthonormality constraint in higher-rank codebooks. However, it is not necessary to resort to complicated algorithms when rank-1 codebooks which have only unit-length constraints are considered.

By considering rank-1 codebooks only, the BCASC algorithm is able to use a simpler optimisation technique which maximises the minimum Euclidean distance between codewords [16]. The Euclidean distance between the codewords alone does not lie on the Grassmannian manifold since it measures distances between codewords and not the lines spanned by the codewords. To ensure that optimisation is performed on the manifold, this approach requires maximising the distance between codewords and all their antipodals leading to

$$\max \min_{\substack{n \neq l \\ \psi \in [0, 2\pi)}} \left\| \mathbf{b}_n - \mathbf{b}_l e^{j\psi} \right\| \text{ subject to } \|\mathbf{b}_n\|^2 = 1 \ \forall n. \tag{3.13}$$

The value of this approach is shown by the fact that the BCASC algorithm achieves lower coherence values for rank-1 codebooks than previous approaches [16]. However, this approach has been criticised for its computational complexity which restricts its practical application to codebooks of small dimensions [70].

3.5 CHAPTER SUMMARY

This chapter provided a brief survey of existing conventional and reduced-control beamforming arrays. Reduced-control arrays can be broadly categorised into thinned and sparse arrays, conventional subarrays, and compressive arrays. Thinned and sparse arrays place a limited number of antenna elements non-uniformly in a given aperture. Conventional subarrays combine antenna-element signals to realise high-directivity beams across limited steering ranges, where the subarrays may be non-overlapping, partially overlapped, or completely overlapped. In the completely overlapped case, all subarrays share the entire aperture. Compressive arrays are a generalisation of reduced-control arrays, but have not yet been investigated in much detail. The research to date on compressive arrays suggests that some form of optimisation is required. The chapter ended with an overview of codebook optimisation, which is the most promising starting point for an investigation into the optimisation of compressive arrays.

CHAPTER 4 EVALUATING THE OPTIMALITY OF RANDOM COMPRESSIVE ARRAYS

4.1 CHAPTER OVERVIEW

The theory of CS has inspired the concept of a compressive DF antenna array where N antenna elements are combined into M measurements before sampling with expected performance similar to that of a conventional N-element array, given that the number of DoAs is sufficiently small. The use of real- and complex-valued random Gaussian sensing matrices in designing such an array has previously been proposed. In this chapter, the use of codebooks with numerically optimised coherence to design compressive arrays is proposed as an alternative for arrays of small sizes. A codebook-based array consistently outperforms a real- and complex-valued Gaussian array in mean DF accuracy. A compressive array that has been optimised for low SLL (from Chapter 6) outperforms all other compressive arrays and comes closest to the performance of an N-element ULA, confirming that optimising a compressive array for low SLL leads to improved DF performance. Random compressive arrays can, therefore, not be assumed to be optimal for arrays of small sizes and the optimisation of compressive arrays is warranted.

This chapter starts by describing the compressive DF problem in Section 4.2. Section 4.3 describes existing compressive array designs and presents a method for designing compressive arrays based on optimised codebooks. Various compressive array designs are compared in terms of coherence and DF performance in Section 4.4. Finally, some conclusions are drawn and the chapter is summarised in Section 4.5.

4.2 COMPRESSIVE DIRECTION FINDING

The baseband signal received by a conventional DF antenna array with N elements is given by [11]

$$\mathbf{x}(t) = \sum_{k=1}^{K} \mathbf{a}(\theta_k) z_k(t) + \mathbf{n}(t), \tag{4.1}$$

where $\mathbf{x}(t)$ is the $N \times 1$ signal vector, $z_k(t)$ represents an incoming signal from direction θ_k, K is the number of sources, $\mathbf{a}(\theta_k)$ is the $N \times 1$ steering vector for the direction θ_k, and $\mathbf{n}(t)$ is an $N \times 1$ additive white Gaussian noise (AWGN) vector.

The noise has been defined as being present at the antenna elements under the assumption that each antenna element has a low-noise amplifier (LNA) with significant gain, so that the noise introduced by the feed network and receivers is negligible. In the case of a ULA, the steering vector is given by (3.1).

This chapter is an updated and extended version of a published conference paper [22].

A set of N_s steering angles are chosen giving an $N \times N_s$ steering or transform matrix $\mathbf{A}$ as [11]

$$\mathbf{A} = \left[\mathbf{a}(\theta_1),\ \mathbf{a}(\theta_2),\ \cdots,\ \mathbf{a}(\theta_{N_s}) \right]. \tag{4.2}$$

The angular spectrum is an $N_s \times 1$ vector containing the signals received from the sampled directions, given by

$$\mathbf{s}(t) = \left[s_1(t),\ s_2(t),\ \cdots,\ s_{N_s}(t) \right]^{\mathrm{T}}, \tag{4.3}$$

allowing the received signal to be written as [11]

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t). \tag{4.4}$$

A compressive DF array uses weighted combinations of the antenna signals to reduce the number of receivers required (Chapter 3). The $M \times 1$ measurement vector is obtained from [11]

$$\mathbf{y}(t) = \mathbf{\Phi}\mathbf{x}(t) = \mathbf{\Phi}\left[\mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)\right] = \mathbf{B}\mathbf{s}(t) + \mathbf{\Phi}\mathbf{n}(t), \tag{4.5}$$

where $\mathbf{\Phi}$ is the $M \times N$ sensing matrix to be implemented in microwave hardware before sampling. $\mathbf{B}$ is the $M \times N_s$ compressed steering matrix given by

$$\mathbf{B} = \mathbf{\Phi}\mathbf{A} = \left[\mathbf{b}(\theta_1),\ \mathbf{b}(\theta_2),\ \cdots,\ \mathbf{b}(\theta_{N_s}) \right], \tag{4.6}$$

where $\mathbf{b}(\theta_s)$ are the compressed steering vectors.

The coherence of the compressed steering matrix $\mathbf{B}$ can be calculated similar to (3.9), but the normalised absolute dot products between neighbouring steering vectors must not be considered, as their similarity will be high due to the fine angular grid. Instead, a minimum distance between the steering angles should be defined for calculating the coherence. The steering range may also be limited by considering only the coherence across the desired steering range.

The angular spectrum is estimated from the measurements by solving (4.5) for $\mathbf{s}(t)$. Since $N_s \gg M$, the system is underdetermined. However, CS theory suggests that this system can nevertheless be solved with high probability of success given that the number of sources K is relatively small ($K \ll N_s$), the number of measurements $M > K$ is sufficiently large, and that $\mathbf{B}$ is robust (Chapter 2) [12].

Conventional super-resolution DF algorithms have successfully been applied to compressive arrays by substituting $\mathbf{B}$ for $\mathbf{A}$ [11]. The angular power spectrum of the minimum variance distortionless response (MVDR) or Capon beamformer can be written as [11]

$$P(\theta) = \frac{1}{\mathbf{a}(\theta)^{\mathrm{H}}\mathbf{R}_x^{-1}\mathbf{a}(\theta)}, \tag{4.7}$$

where $\mathbf{R}_x$ is the covariance matrix of the received signal vector given by $\mathbb{E}[\mathbf{x}(t)\,\mathbf{x}(t)^{\mathrm{H}}]$, with $\mathbb{E}[\,]$ the expectation operator. MVDR can thus be applied to a compressive array by replacing $\mathbf{x}(t)$ with $\mathbf{y}(t)$ and $\mathbf{a}(\theta)$ with $\mathbf{b}(\theta)$.

4.3 SENSING MATRIX DESIGN

Real-valued sensing matrices ($\mathbf{\Phi}$) with independent, identically distributed (i.i.d.) zero-mean entries drawn from the Gaussian distribution are widely used in CS applications [12] and have been proposed for use in compressive arrays [11, 14]. Complex-valued sensing matrices with real and imaginary components drawn from the Gaussian distribution have also been proposed for compressive arrays [14]. The resulting compressed steering/transform matrix $\mathbf{B} = \mathbf{\Phi}\mathbf{A}$ is also assumed to be i.i.d. Gaussian due

to the properties of the distribution (Chapter 2) [12]. These random compressed steering/transform matrices have been shown to satisfy the RIP criterion (Chapter 2) with high probability, based on the assumption that the dimensions of the matrix are sufficiently large [17]. However, arrays of very small sizes are still of practical interest. The number of antenna elements in an array is often much smaller than the number of samples in other typical CS applications which take, for example, time-domain samples using analogue-to-digital converters (ADCs) or spatial samples using image sensors.

The relatively small size of compressive arrays suggests the use of numerically optimised matrices. The design of low-coherence codebooks has been studied for application to telecommunication systems [67], and their potential for use in CS applications has also been noted [16]. Various approaches towards optimising codebooks are described in Section 3.4 and Chapter 5.

The matrix $\mathbf{\Phi}$ can be obtained from an optimised $M \times N$ matrix $\mathbf{B}$ by creating an orthogonal $N \times N$ matrix $\mathbf{A}$ through the careful selection of $N_s = N$ sampling angles and solving

$$\mathbf{\Phi} = \mathbf{B}\mathbf{A}^{-1}. \tag{4.8}$$

Note, however, that this method only allows control of the system response at the N sampling angles. Chapter 6 presents an algorithm for optimising the sensing matrix $\mathbf{\Phi}$ over continuous sampling angles.

4.4 NUMERICAL SIMULATION RESULTS

Conventional 4- and 16-element ULAs and compressive arrays with 4 subarrays and 16 elements were compared, all with isolated isotropic elements and half-wavelength spacing. Real and complex Gaussian arrays and an array based on an optimised codebook were chosen for comparison. In the case of the codebook-based array, the complex-valued 4×16 matrix $\mathbf{B}$ was found using the BCASC algorithm [16] (Sections 3.4 and 5.2). The best out of ten designs for all compressive arrays were chosen based on the coherence of $\mathbf{B}$ over steering angles from $-60°$ to $60°$. A steering range from $-60°$ to $60°$ was chosen for evaluating the coherence since this is considered a wide steering range [2]. The DF results for the compressive array in Section 6.4.2 is also presented for comparison, although its design will not be described here in detail.

Figure 4.1 shows the coherence patterns, the normalised absolute dot products between steering vectors, of a conventional 16-element ULA with a uniform excitation. Figure 4.2 shows the coherence patterns of the chosen compressive arrays. Angles are shown in $u = \sin(\theta)$ since the N orthogonal angles in $\mathbf{A}$ in (4.8) are equally spaced in u (as indicated by the grid lines). Steering vectors are compared between steering angles $|\theta_s| \leq 60°$ and pattern angles $|\theta_a| \leq 90°$ when calculating coherence. Although the steering range is limited, the steering vectors in $|\theta_s| \leq 60°$ are still compared to the steering vectors in $|\theta_a| \leq 90°$, since it is not desirable to have a high similarity with steering vectors outside the steering range. Such a high similarity would introduce an ambiguity which could cause a DoA outside the steering range to appear as a DoA within the steering range.

Two main features of the antenna patterns determine the DF performance of any array. These factors are the width of the main lobe and the coherence value which is the largest value outside the main lobes on the line $\theta_s = \theta_a$ in Figures 4.1 and 4.2. The width of the main lobe influences the accuracy with which a DoA can be resolved, whereas the coherence influences the performance by introducing ambiguities, especially at low signal-to-noise ratio (SNR).

In Figure 4.1, the first sidelobes determine the coherence of the ULA as -13.15 dB and the entire pattern translates uniformly with steering angle (in u). The main lobes are well-defined. At each

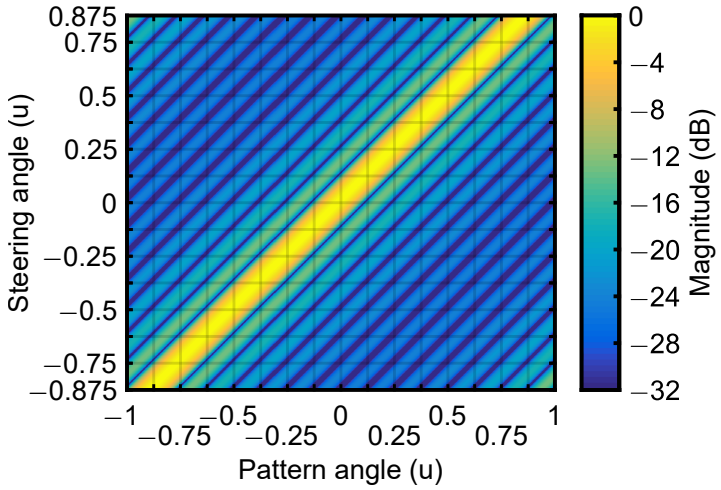


Figure 4.1. Coherence patterns for a 16-element ULA.

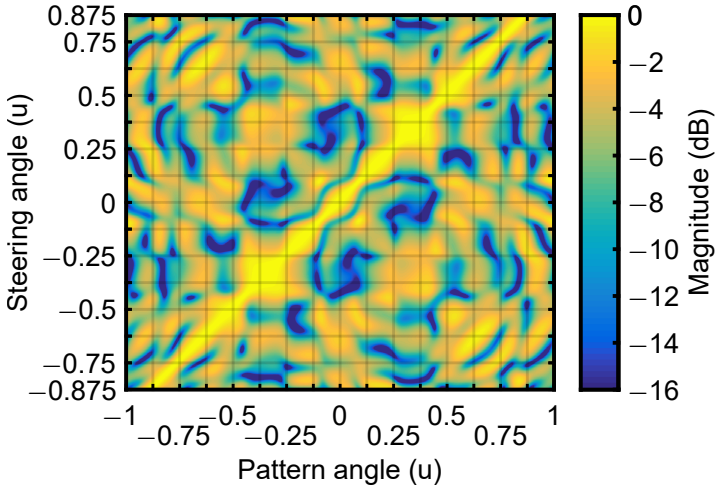
steering angle, the nulls and the main-lobe peak are located at 16 pattern angles equally spaced in u, for example at the grid intersections in Figure 4.1.

In Figure 4.2(a), the real Gaussian array can be seen to exhibit large similarity between its steering vectors, with a coherence of −0.36 dB. The main lobes are not well-defined and their widths vary significantly with steering angle. The complex Gaussian array in Figure 4.2(b) has a coherence of −0.6 dB and has more well-defined main lobes, although some variation in the main-lobe widths remains. The codebook-based array in Figure 4.2(c) achieves a coherence of −0.81 dB and has more uniform main-lobe widths than the complex Gaussian array. The coherence patterns for the SLL-optimised compressive array (Section 6.4.2) is equivalent to the array patterns shown in Figure 6.3, and the array has nearly uniform main-lobe widths and a coherence of −2.89 dB.

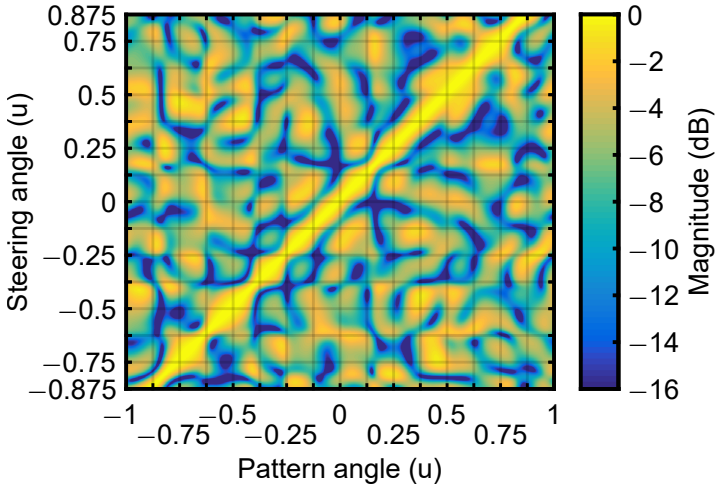
Figure 4.3 shows the normalised absolute dot products of the compressive arrays at broadside steering—cross-sections of the plots in Figure 4.2 at $u_s = 0$. Also shown is the Welch bound for $M = 4$, $N = 16$ which is −7 dB. The codebook-based array satisfies the Welch bound at the N sampling angles, but not necessarily elsewhere as the coherence was only optimised at these angles. This can also be seen by the fact that most of the large lobes in Figure 4.2(c) occur between the grid lines, while the Welch bound is satisfied at the intersections of the grid lines. By comparison, the Gaussian arrays do not meet the Welch bound at the grid intersections.

Coherence is a well-established measure of the optimality of a compressed transform matrix and is not application-specific (Sections 2.6 and 3.4) [32, 71]. The fact that the optimised compressive arrays have better coherence than the random compressive arrays suggests that it cannot be assumed that random compressive arrays are optimal for arrays of small sizes, irrespective of the application. In the case of the DF application, the effect of the coherence patterns on DF performance is illustrated below by comparing the DF accuracy of the various arrays.

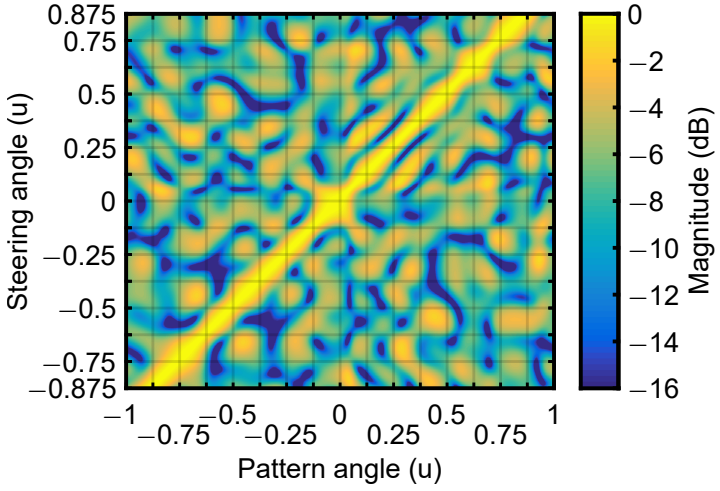
The DF performance of the compressive arrays as well as conventional M- and N-element arrays were evaluated using the MVDR algorithm with 100 time samples. The incoming signal DoA was swept from −60° to 60° in 121 steps, and the statistics were generated from 1 000 tests per step. The angular spectra were calculated across $\theta \in [-90°, 90°]$ using N_s =18 001 sampling angles for an angular grid resolution of 0.01°. Cubic spline interpolation was further performed on the angular spectra using 100 times more sampling angles, and the DoAs estimated as the location of the peak in each spectrum. The SNR at the antenna elements was swept from −5 dB to 30 dB in 0.5 dB increments.



(a) Real Gaussian array



(b) Complex Gaussian array



(c) Codebook-based array

Figure 4.2. Coherence patterns for various compressive DF arrays.

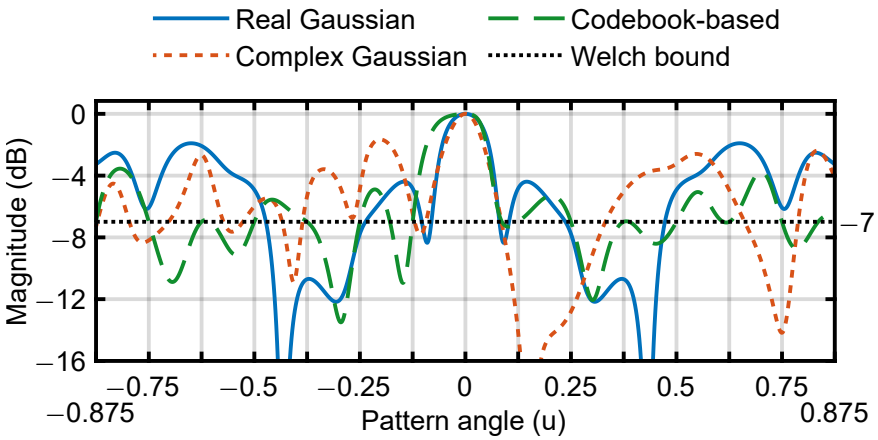


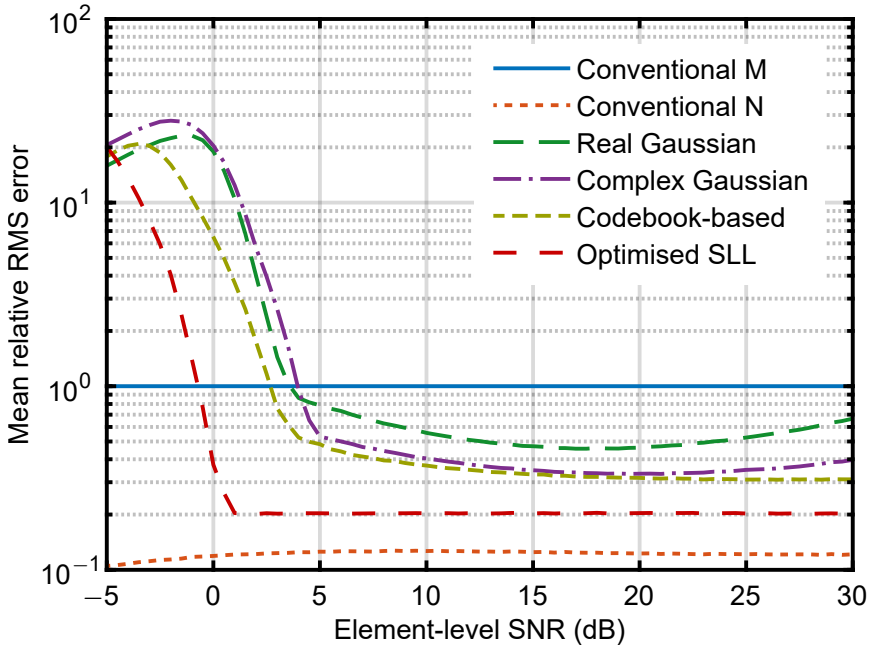
Figure 4.3. Coherence patterns for DF arrays with $M = 4$ and $N = 16$ with broadside beam steering.

In order to compare the performance of the various arrays, the root mean square (RMS) error in the DoA estimate was taken as a fraction of the RMS DF error of the M-element conventional array at each steering angle, and the mean and maximum of these values taken over all steering angles. The resulting mean and maximum relative RMS errors for the various arrays are shown in Figure 4.4. Also shown is the performance of the SLL-optimised compressive array from Section 6.4.2 that has been optimised across continuous sampling angles.

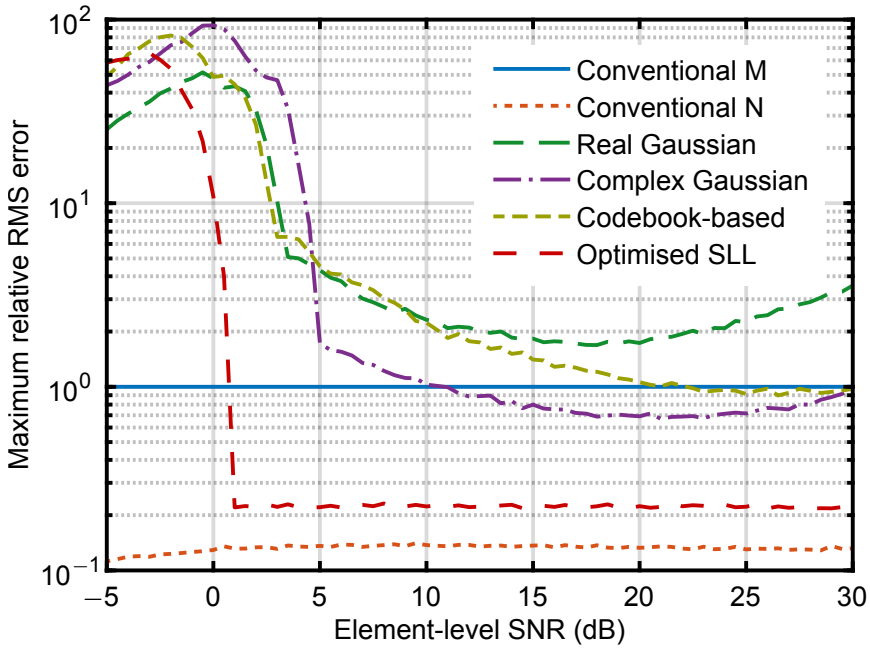
The real Gaussian array has the worst performance of the compressive arrays from 4.5 dB element-level SNR upwards in the mean sense, and performs worse than the M-element ULA across all SNRs in the maximum sense. The codebook-based array outperforms the complex Gaussian array in the mean sense at all SNRs, and has a mean relative RMS error of less than 1 across a wider range of SNRs. In the maximum sense, the complex Gaussian array outperforms the codebook-based array from 5 dB element-level SNR upwards. However, the random arrays and the codebook-based array do not perform particularly well in the maximum sense at any SNR. The SLL-optimised array from Section 6.4.2 is by far the best compressive array, converging to one-fifth of the RMS error of an M-element ULA in the mean sense, and only slightly worse in the maximum sense.

At higher SNRs, the compressive arrays generally perform better than a conventional M-element array in the mean sense, but worse than an N-element ULA. At lower SNRs, the compressive arrays perform worse than an M-element ULA which can be explained by the relatively high coherence values of the compressive arrays, leading to ambiguities in the presence of noise. The real Gaussian array has the worst performance, and such arrays are clearly not optimal for DF applications. Using complex-valued sensing matrices presents a better approach, but the fact that the codebook-based array had better performance than the complex Gaussian array in the mean sense again suggests that random compressive arrays cannot be assumed to be optimal. The suboptimality of random arrays is confirmed by the superior performance of an array that has been optimised for low SLL across continuous sampling angles, which will be described in detail in Chapter 6. The performance of the SLL-optimised array also confirms that optimising a compressive array for low SLL leads to improved DF performance.

The assumption behind these conclusions is that the number of receivers is relatively small, which is considered a reasonable assumption for practical DF systems. As the number of antenna elements and receivers increases, the performance of random compressive arrays is expected to improve. At which point random compressive arrays begin to become useful is yet to be determined. Optimising compressive arrays remains the better approach for predictable results at least until such point that the



(a)



(b)

Figure 4.4. (a) Mean and (b) maximum relative MVDR RMS DoA estimation error for $M = 4$ and/or $N = 16$ conventional and compressive arrays.

computational complexity of a compressive-array optimisation algorithm becomes too expensive due to the size of the array.

4.5 CHAPTER SUMMARY

The use of discrete codebooks with optimised coherence has been proposed for designing compressive DF arrays. An $M = 4$, $N = 16$ codebook-based compressive array was found to result in better coherence and mean DF accuracy than real and complex-valued random Gaussian compressive arrays, and to outperform an M-element ULA in the mean sense over a wider range of SNRs than the random arrays. These results suggest that random compressive arrays cannot be assumed to be optimal for arrays of relatively small sizes. The SLL-optimised array from Section 6.4.2 outperformed all other compressive arrays, showing that optimising a compressive array for low SLL is an effective way of improving the DF performance of a compressive array.

CHAPTER 5 ADDRESSING COMPUTATIONAL COMPLEXITY IN COHERENCE OPTIMISATION ALGORITHMS

5.1 CHAPTER OVERVIEW

In Chapter 4, it was shown that random compressive arrays cannot be assumed to be optimal and that the optimisation of compressive arrays is warranted. The use of existing optimised codebooks in designing compressive arrays was promising, but only allowed control over array patterns at N sampling angles. There is, therefore, a need for an algorithm that can optimise a compressive array over continuous sampling angles. The obvious starting point in developing such an algorithm would be the state-of-the-art BCASC algorithm (Section 3.4) [16, 19]. However, optimising sensing matrices across continuous sampling angles would constitute a problem of considerably greater complexity than the already-complex problem of optimising discrete codebooks. Although it achieves the best-known coherence results, the BCASC algorithm has been criticised for its computational complexity [70]. Before attempting to extend discrete codebook optimisation algorithms to the problem of optimising compressive arrays across continuous sampling angles, it would be wise to first attempt to address the issue of computational complexity in the BCASC algorithm.

In this chapter, an iterative algorithm for the numerical optimisation of rank-1 Grassmannian codebooks is presented. This algorithm achieves similar coherence results to the BCASC algorithm, but requires a median of 9.52 times less computation time. This improvement is achieved by reformulating the problem to directly minimise coherence instead of maximising the Euclidean distance between codewords, removing the need to consider large numbers of complex antipodals. The run-time improvement allows the optimisation of the larger codebooks required in many applications [70], and provides a better starting point for the development of a compressive array optimisation algorithm.

Section 5.2 describes the existing BCASC codebook optimisation algorithm. In Section 5.3, a new algorithm is proposed which minimises codebook coherence directly. The implementation of the algorithms in software, an important factor considering the focus being computational complexity, is described in Section 5.4, and coherence results for codebooks of various sizes are presented in Section 5.5. Finally, some conclusions are drawn and the chapter is summarised in Section 5.6.

This chapter is a modified version of a published journal paper [23]. Portions of this chapter are ©2017 IEEE; reprinted, with permission, from [23].

The MATLAB code for generating the main results in this chapter is available online at `https://doi.org/10.24433/CO.e059a3c5-cf64-4f77-a83d-d5b8aebda64a`.

5.2 BEST COMPLEX ANTIPODAL SPHERICAL CODES

The current state-of-the-art BCASC algorithm, as described in [16], is outlined below.

Ignoring codeword antipodals, (3.13) can be reformulated as

$$\min \sum_{n \neq l} \|\mathbf{b}_n - \mathbf{b}_l\|^{-p} \quad \text{subject to} \quad \|\mathbf{b}_n\|^2 = 1 \quad \forall\, n, \tag{5.1}$$

where $p \to \infty$. As p increases, more weight is assigned to the smaller pairwise distances between codewords, until the minimum pairwise distance dominates the expression. The method of Lagrange multipliers can be used to show that the conditions for a local minimum are

$$\mathbf{b}_n = \underline{\sum_{n \neq l} \frac{\mathbf{b}_n - \mathbf{b}_l}{\|\mathbf{b}_n - \mathbf{b}_l\|^{p+2}}} \quad \forall n, \tag{5.2}$$

where underlining denotes the normalisation $\underline{\mathbf{u}} = \mathbf{u}/\|\mathbf{u}\|$.

The optimised codebook $\mathbf{B}$ can be found by interpreting (5.2) as a set of forces at optimisation step k ($\mathbf{f}_n^{(k)}$ is given by the right-hand side (RHS) of (5.2)) and adding a gain factor α multiplied by the matrix of forces ($\mathbf{F}^{(k)} = \left[\mathbf{f}_1^{(k)} \cdots \mathbf{f}_N^{(k)}\right]$) to the current matrix $\mathbf{B}^{(k-1)}$ giving

$$\mathbf{B}^{(k)} = \underline{\mathbf{B}^{(k-1)} + \alpha \mathbf{F}^{(k)}}, \quad k = 0, 1, \ldots, \tag{5.3}$$

where underlining denotes the column normalisation $\underline{\mathbf{U}} = \{\mathbf{u}_n/\|\mathbf{u}_n\|\}_{n=1}^N$ to ensure that all $\|\mathbf{b}_n\|^2 = 1$. The result converges to a fixed point as $k \to \infty$ when α is small enough [16].

Extending (5.2) to consider the antipodals results in

$$\mathbf{f}_n^{(k)} = \underline{\int_0^{2\pi} \sum_{n \neq l} \frac{\mathbf{b}_n^{(k)} - \mathbf{b}_l^{(k)} e^{j\psi}}{\|\mathbf{b}_n^{(k)} - \mathbf{b}_l^{(k)} e^{j\psi}\|^{p+2}} \, d\psi}, \tag{5.4}$$

which can be solved using numerical integration. Alternatively, the integral can be approximated by a summation to give

$$\mathbf{f}_n^{(k)} \approx \underline{\sum_{q=1}^{Q} \sum_{n \neq l} \frac{\mathbf{b}_n^{(k)} - \mathbf{b}_l^{(k)} e^{j2\pi q/Q}}{\|\mathbf{b}_n^{(k)} - \mathbf{b}_l^{(k)} e^{j2\pi q/Q}\|^{p+2}}}, \tag{5.5}$$

where the number of antipodals considered, Q, determines the accuracy of the approximation. The main drawback of the BCASC algorithm is the computational complexity resulting from the need to consider a large number of antipodals.

A random codebook may be used as the starting point for the first subproblem. The result of each subproblem is then used as starting point for the next subproblem which uses a larger value of p. This process is repeated for $p \to \infty$.

5.3 COHERENCE-BASED CODEBOOK OPTIMISATION

An algorithm with a coherence metric which avoids the need to explicitly consider the antipodals is proposed below to address the computational complexity of the BCASC algorithm.

Instead of using the metric $\sum_{n \neq l} \|\mathbf{b}_n - \mathbf{b}_l\|^{-p}$ as in (5.1), consider the minimisation of the metric

$$g(\mathbf{B}) = \sum_{n \neq l} \left(\left| \mathbf{b}_n^H \mathbf{b}_l \right|^2 - \mu_t^2 \right)^p, \quad p = 2, 4, 6, \ldots, \tag{5.6}$$

which minimises the absolute difference between the squares of the pair-wise absolute dot products and some target value $\mu_t = \beta\,\mu_c(M,N)$, where $\mu_c(M,N)$ is calculated from (3.10). Squaring the absolute value provides a smooth metric, and restricting p to even values ensures that the summed terms increase monotonically from their minima.

A non-zero target value is used in (5.6) to improve the rate of convergence [66]. However, the algorithm described in Section 5.4 uses a fixed iteration gain for each subproblem making it prone to instability when $\beta = 1$, so it is necessary to reduce μ_t by choosing $\beta < 1$ to improve stability.

In order to enforce the unit-length constraint on the codewords, define the Lagrange function as

$$g(\mathbf{B},\boldsymbol{\lambda}) = g(\mathbf{B}) + \sum_{n=1}^{N} \lambda_n \left(\|\mathbf{b}_n\|^2 - 1 \right), \tag{5.7}$$

where $\boldsymbol{\lambda} = \{\lambda_n\}_{n=1}^{N}$ are the Lagrange multipliers for the N unit-length constraints. The conditions for finding the constrained minimum of (5.6) are [72]

$$\frac{\partial g(\mathbf{B},\boldsymbol{\lambda})}{\partial \mathrm{Re}\{b_{m,n}\}} = 0, \quad \frac{\partial g(\mathbf{B},\boldsymbol{\lambda})}{\partial \mathrm{Im}\{b_{m,n}\}} = 0 \text{ and } \frac{\partial g(\mathbf{B},\boldsymbol{\lambda})}{\partial \lambda_n} = 0 \tag{5.8}$$

for all $b_{m,n}$ and λ_n. These conditions are then used to derive the force vector at iteration k in Addendum A.1, leading to

$$\mathbf{f}_n^{(k)} = -2 \sum_{n \neq l} \left(\left| \mathbf{b}_n^{(k)\mathrm{H}} \mathbf{b}_l^{(k)} \right|^2 - \mu_t^2 \right)^{p-1} \left(\mathbf{b}_l^{(k)\mathrm{H}} \mathbf{b}_n^{(k)} \right) \cdot \mathbf{b}_l^{(k)}. \tag{5.9}$$

The proposed algorithm iteratively updates the codebook using (5.3) leading to the coherence-based Grassmannian codebook (CBGC) algorithm.

In (5.9), the factor $\left(\mathbf{b}_l^{(k)\mathrm{H}} \mathbf{b}_n^{(k)} \right) \cdot \mathbf{b}_l^{(k)}$ is the vector projection of $\mathbf{b}_n^{(k)}$ onto $\mathbf{b}_l^{(k)}$. Due to the negative sign in (5.9), the weighted sum of projections is subtracted from $\mathbf{b}_n^{(k)}$ in (5.3), with those larger or smaller than the target μ_t given preference via the factor $\left(\left| \mathbf{b}_n^{(k)\mathrm{H}} \mathbf{b}_l^{(k)} \right|^2 - \mu_t^2 \right)^{p-1}$, with larger p increasing the effect. Thus $\mathbf{b}_n^{(k)}$ is altered to reduce the components it has in common with other codewords.

5.4 ALGORITHM IMPLEMENTATION

Both the BCASC and CBGC algorithms were implemented using the generalisation of the BCASC algorithm shown in Algorithm 5.1. This was done to compare only the effect of the updates in (5.5) and (5.9) by using the same algorithm.

The main differences between Algorithm 5.1 and the form in [16] are the initial value of p and the convergence criteria. Algorithm 5.1 utilises codeword convergence $\left\| \mathbf{b}_n^{(k)} - \mathbf{b}_n^{(k-1)} \right\|$ rather than the convergence of force vectors to codewords $\left\| \mathbf{f}_n^{(k)} - \mathbf{b}_n^{(k)} \right\|$. This change is necessary as force vectors do not converge to codewords in the CBGC algorithm, while the codewords themselves always converge successfully.

The BCASC and CBGC algorithms were implemented in MATLAB R2016a and run on machines with two 6-core Intel Xeon E5-2630 processors and 32 GB of memory each.

The BCASC algorithm may use numerical integration with (5.4) or a finite number of antipodals with (5.5) [16]. In order to vectorise the operations and to exploit MATLAB's parallel-processing

Algorithm 5.1 Generalised coherence optimisation algorithm adapted from [16]. Taken from [23], ©2017 IEEE.

procedure OPTIMISE COHERENCE($M, N, \alpha_{\text{init}}$)
  $\mathbf{B}^{(0)}_{M\times N} \leftarrow$ random complex codebook
  $\varepsilon \leftarrow 10^{-10}$ ▷ Convergence parameter
  $\alpha \leftarrow \alpha_{\text{init}}$
  $p_{\max} \leftarrow 2^9$, $p \leftarrow 2$
  $k_{\max} \leftarrow 10^5$ ▷ Maximum number of iterations
  while $p \leq p_{\max}$ **do**
    $k \leftarrow 1$
    while $k \leq k_{\max}$ **and any** $\left\|\mathbf{b}_n^{(k)} - \mathbf{b}_n^{(k-1)}\right\| \geq \varepsilon$ **do**
      Calculate $\mathbf{F}^{(k)} = \left[\mathbf{f}_1^{(k)}\, \mathbf{f}_2^{(k)} \cdots \mathbf{f}_N^{(k)}\right]$
      $\mathbf{B}^{(k)} = \underline{\mathbf{B}^{(k-1)} + \alpha\mathbf{F}^{(k)}}$ ▷ Update $\mathbf{B}$
      $k \leftarrow k + 1$ ▷ Increase iteration number
    $p \leftarrow 2p$ ▷ p for next subproblem
    $\alpha \leftarrow \alpha_{\text{init}}/(p-1)$ ▷ α for next subproblem

capabilities, the discrete approximation was used, allowing the antipodals to be precomputed at each iteration.

The number of antipodals considered by the generalised BCASC algorithm (Q) was selected by comparing the results of ten tests for $Q = 50$, 100 and 200 as shown in Table 5.1. All coherence values were identical to three significant figures. In most cases the run times were longer for $Q = 50$ than for $Q = 100$ as a result of incomplete consideration of the antipodals, while $Q = 200$ only served to increase the run times. Therefore, $Q = 100$ was used in subsequent tests.

The results obtained using the original BCASC algorithm described in [16, Fig. 1] for $Q = 100$ without the changes mentioned above are also shown in Table 5.1. The changes resulted in coherences that are equivalent to within 0.15%, and run times that are faster or only 3.6% slower in one case. These observations demonstrate that the BCASC algorithm is not compromised by the changes made.

The starting codebook $\mathbf{B}^{(0)}$ was generated by normalising the columns of a matrix whose elements had their real and imaginary parts drawn from a zero-mean Gaussian distribution. The variance of this distribution has no effect as the codes are normalised. The initial gain value, α_{init}, was set to 0.9 as in [16]. However, the CBGC algorithm required lower gain for stability and thus used $\alpha_{\text{init}} = 0.04$. A value of $\beta = 0.5$, mid-way between the BCASC approach ($\beta = 0$) and the optimum coherence ($\beta = 1$), resulted in convergence in all test cases.

Due to the large exponents p, a force vector in the CBGC algorithm would occasionally round to a zero vector before normalisation. Testing whether this occurred was thus necessary to avoid attempting to normalise a zero vector.

5.5 RUN TIME AND COHERENCE RESULTS

The minimum and maximum coherences and run times for ten tests are compared in Table 5.2 for codebooks of various dimensions obtained using the BCASC and CBGC algorithms. The lower bound from (3.10) and the BCASC coherences published in [16] are also shown.

The CBGC algorithm resulted in coherences equal to or better than those of the generalised implementation of the BCASC algorithm except $M = 4$, $N = 20$, where only the maximum coherence was

Table 5.1. BCASC Coherence and Run Times of Ten Tests for Various Values of Q using Generalised and Original Forms. Taken from [23], ©2017 IEEE.

		Minimum coherence, Q =				Mean run time (s), Q =			
M	N	50	100	200	100 [16, Fig. 1]	50	100	200	100 [16, Fig. 1]
4	7	0.354	0.354	0.354	0.354	292.0	2.22	3.35	8.21
4	16	0.447	0.447	0.447	0.447	1 010	10.70	13.78	39.67
4	64	0.688	0.688	0.688	0.687	23 816	25 471	26 891	30 531
5	10	0.333	0.333	0.333	0.333	1 915	2 312	3 590	2 231
5	16	0.390	0.390	0.390	0.390	1 855	1 751	2 218	2 837

worse. The CBGC algorithm also had minimum coherences equal to or better than the BCASC results published in [16] except for $M = 4$, $N = 64$.

The run times of the CBGC algorithm were significantly better than those of the BCASC algorithm except for $M = 4$, $N = 20$, where the minimum run time was marginally worse. When the mean run times of the BCASC and CBGC algorithms were compared for the codebook sizes in Table 5.2, the CBGC algorithm was found to be a median of 9.52 times faster than the BCASC algorithm.

The number of iterations in Table 5.2 shows no clear benefit for either algorithm, with the CBGC algorithm requiring a median of only 14% fewer iterations, and each algorithm requiring fewer iterations for some of the problems.

A greater run-time improvement may be anticipated as the CBGC force vector in (5.9) has $Q = 100$ times fewer terms than the BCASC force vector in (5.5) and the median of the number of iterations are similar. However, the optimised implementation of (5.5) is better able to exploit the vectorisation [73] capabilities of the MATLAB development environment used during testing, thereby diminishing the run-time improvement achieved.

5.6 CHAPTER SUMMARY

An iterative algorithm which makes use of a coherence-based metric for optimising rank-1 Grassmannian codebooks has been proposed. Evaluating this algorithm against the Euclidean-distance-based BCASC algorithm within the same algorithmic framework has shown that the proposed algorithm results in almost an order-of-magnitude improvement in run time while achieving comparable or better coherence results. The proposed algorithm is, therefore, a promising starting point for the development of an algorithm that optimises compressive arrays across continuous sampling angles.

Table 5.2. Coherences, Run Times and Iterations of Ten Tests for the BCASC ($Q = 100$) and CBGC Algorithms. Taken from [23], ©2017 IEEE.

		Coherence						Run time (s)				Total number of iterations			
		μ_c	BCASC	Minimum		Maximum		Minimum		Maximum		Minimum		Maximum	
M	N	(3.10)	from [16]	BCASC	CBGC	BCASC	CBGC	BCASC	CBGC	BCASC	CBGC	BCASC	CBGC	BCASC	CBGC
2	8	0.7500	0.7950	0.7950	0.7942	0.7950	0.7942	1 827.09	4.09	1 861.76	5.40	800 203	9 419	800 302	12 516
3	16	0.6202	0.6491	0.6499	0.6491	0.6499	0.6494	3 354.15	184.49	3 522.21	294.97	457 639	211 024	466 508	339 380
4	5	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.24	0.03	0.32	0.09	162	61	195	83
4	6	0.3162	0.3277	0.3282	0.3276	0.3282	0.3276	21.61	5.54	23.28	5.76	11 066	16 592	11 883	17 119
4	7	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	2.04	0.22	2.39	0.28	977	518	1 143	647
4	8	0.3780	0.3780	0.3780	0.3780	0.3780	0.3780	2 041.57	255.73	2 069.23	263.36	800 022	600 033	800 023	600 034
4	9	0.3953	0.4022	0.4025	0.4022	0.4025	0.4022	63.34	23.05	66.32	24.09	20 905	48 475	21 939	50 194
4	10	0.4082	0.4118	0.4124	0.4118	0.4124	0.4118	39.80	11.20	47.50	11.61	11 320	21 301	12 313	21 813
4	16	0.4472	0.4472	0.4472	0.4472	0.4472	0.4472	9.44	0.83	12.29	1.13	1 170	947	1 501	1 298
4	20	0.5000	0.5000	0.5000	0.5000	0.5268	0.5316	57.15	57.86	2 074.32	447.25	5 019	55 682	184 045	409 252
4	64	0.6000	0.6869	0.6877	0.6874	0.6888	0.6884	20 944.52	2 828.72	29 267.22	3 176.66	339 264	626 272	480 875	701 432
5	6	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.63	0.06	0.77	0.08	289	109	357	134
5	7	0.2582	0.2670	0.2676	0.2667	0.2676	0.2667	270.53	4.10	275.53	4.57	121 747	10 498	123 379	11 567
5	8	0.2928	0.2955	0.2957	0.2954	0.2957	0.2954	623.44	12.35	633.72	12.69	229 401	28 091	232 326	28 684
5	9	0.3162	0.3207	0.3212	0.3205	0.3212	0.3205	247.97	57.24	276.22	59.78	76 273	118 701	84 569	121 465
5	10	0.3333	0.3333	0.3333	0.3333	0.3334	0.3333	106.86	0.95	3 784.14	377.52	27 297	1 699	900 000	700 002
5	16	0.3830	0.3889	0.3897	0.3889	0.3914	0.3904	951.62	116.40	2 978.83	321.67	110 763	132 058	347 026	366 100
5	43	0.4956	–	0.5007	0.5005	0.5380	0.5376	5 142.81	94.65	14 943.86	1 443.43	139 375	35 525	407 761	545 384
6	37	0.4082	–	0.4148	0.4082	0.4164	0.4151	6 559.04	242.48	12 449.61	1 160.20	208 980	107 876	397 956	507 834
7	42	0.3492	–	0.3720	0.3541	0.3773	0.3757	6 116.10	199.19	27 019.78	2 139.75	154 812	56 563	685 308	570 066

CHAPTER 6 DEVELOPING A COMPRESSIVE ARRAY SLL OPTIMISATION ALGORITHM

6.1 CHAPTER OVERVIEW

In Chapter 4, it was shown that random compressive arrays are suboptimal, and that a technique is required for optimising such arrays. In Chapter 5, a computationally efficient algorithm for optimising discrete codebooks was proposed, and it was suggested that this algorithm would be a suitable starting point for developing a compressive array SLL optimisation algorithm.

In this chapter, a framework for the optimisation of narrowband compressive arrays with arbitrary sidelobe-pattern requirements is presented. This framework includes other reduced-control arrays as special cases. The framework is inspired by the coherence optimisation algorithm presented in Chapter 5. The coherence optimisation problem is extended to consider the problem of minimising SLL in a compressive array. The presented approach allows the sidelobe amplitudes to be specified independent of pattern and steering angle, which is useful as shaped sidelobe patterns are often desirable in antenna arrays [15].

The proposed compressive array framework is not restricted to a particular hardware configuration, but represents a flexible reduced-control array design methodology. For example, a compressive feed network may be implemented using microwave circuitry to enable an array to have a larger number of antenna elements than beamforming controls, or as a software algorithm to reduce data rates.

The versatility of the proposed algorithm is illustrated by synthesising linear and circular compressive arrays with a variety of constraints, including hard and soft nulls. Comparisons to optimal solutions [34], thinned arrays synthesised using the iterative FFT technique (IFT) [38], conventional arrays synthesised by the matrix inversion method [15], and a dual-transform completely overlapped subarray system [2] demonstrate the performance of compressive arrays synthesised using the proposed approach.

Section 6.2 starts by defining compressive arrays as a generalisation of reduced-control arrays, and shows how radiation patterns may be derived for compressive arrays. The continuous coherence optimisation problem for minimising SLL is presented in Section 6.3 along with an algorithm for accomplishing this goal. Results for a number of designs which highlight various advantages of the proposed approach are presented in Section 6.4. Finally, the chapter is summarised in Section 6.5.

This chapter is a modified and updated version of a published journal paper [1]. Portions of this chapter are ©2018 IEEE; reprinted, with permission, from [1].

The MATLAB code for generating the main results in this chapter is available online at `https://doi.org/10.24433/CO.24b779d2-852f-4af7-a489-f206fde12c23`.

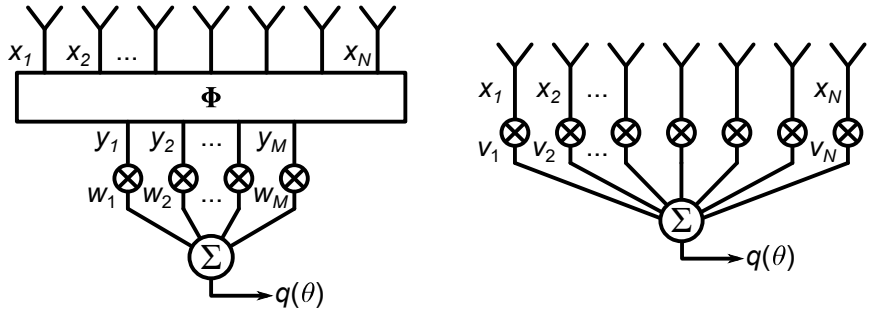


Figure 6.1. (a) A compressive array with N elements and M subarrays with beamforming weights for array pattern calculations. (b) An N-element ULA with weights that result in the same patterns as the corresponding compressive array. Taken from [1], ©2018 IEEE.

6.2 A GENERALISED VIEW OF REDUCED-CONTROL ARRAYS

6.2.1 Defining compressive arrays

Consider the general compressive array illustrated in Figure 6.1(a), shown in the receive configuration. The results obtained are, however, equally valid for a transmit array due to reciprocity. The system is represented by $\mathbf{y} = \mathbf{\Phi x}$, where $\mathbf{y}$ is the $M\times1$ vector of signals at the beamforming controls, $\mathbf{x}$ is the $N\times1$ vector of signals at the antenna elements, and $\mathbf{\Phi}$ is the $M \times N$ sensing matrix or feed network response matrix representing a compressive feed network with $M \leq N$ (Chapter 2, 4). Row m of $\mathbf{\Phi}$ describes how the N antenna-element signals are weighted and combined to form the output at the mth subarray. Equivalently, row m of $\mathbf{\Phi}$ describes the aperture illumination at the antenna elements produced by the mth subarray on transmit. The $M\times1$ compressed steering vector (the response $\mathbf{y}$ to a reference wave) is then $\mathbf{b}(\theta) = \mathbf{\Phi a}(\theta)$ (Chapter 4) [11].

A conventional uniform-excitation array with one element per beamforming control can be represented by the $N \times N$ sensing matrix $\mathbf{\Phi} = \mathbf{I}$. A non-weighted thinned array will have an $M \times N$ sensing matrix made up of M rows taken from an $N \times N$ identity matrix [11]. A partially overlapped subarray [7] will have at least one non-zero entry per row, while a completely overlapped subarray [2] can be represented by a fully populated sensing matrix. The compressive DF array in [11] uses a sensing matrix with random Gaussian entries (Chapter 4).

The term compressive array arises as the synthesis of antenna arrays with complex-valued sensing matrices with $M < N$ is considered, similar to the approach used in CS (Chapter 2) [12]. As outlined above, the compressive array formulation includes most existing reduced-control arrays with fixed element positions as special cases, making it a generalisation of the reduced-control concept. The generalised definition of compressive arrays does not include any hardware constraints. Instead, all constraints on $\mathbf{\Phi}$ are removed to allow for more degrees of freedom in an effort to find the best-performing designs. By using this generalisation, it is shown how sensing matrices may be designed to improve on existing criteria and fulfil a variety of criteria that could not previously be considered for reduced-control arrays.

A compressive feed network can be implemented in hardware or in software, for transmission and/or reception. Overlapped subarrays suggest that feed networks can be implemented in microwave circuity where each output is a weighted combination of multiple antenna elements (Chapter 3) [2, 3, 7, 21]. The hardware implementation of a compressive array will be considered in Chapters 7 and 8. The

proposed method could also be applied to an array with a receiver at each element by using a software implementation of the sensing matrix. This would enable reduced data rates for transmission to a central processing station, for example.

6.2.2 Compressive array patterns

A set of M complex-valued excitations at the M subarray ports is defined to characterise a compressive array. The feed network, described by $\mathbf{\Phi}$, results in compressed steering vectors $\mathbf{b}(\theta)$ which change in both amplitude and phase with steering angle. This requires each steering angle to have its own set of weights to achieve the desired pattern when steering in that direction.

Considering the illustration in Figure 6.1(a), the array square-root power pattern is defined as the weighted sum of the M subarray outputs due to a reference wave from the direction θ as

$$q(\theta) = \left|\mathbf{w}(\theta_s)^{\mathrm{T}}\mathbf{b}(\theta)\right|, \tag{6.1}$$

where $\mathbf{w}(\theta_s)$ is the $M\times 1$ vector of complex weights for the steering angle θ_s.

The aim of a compressive array design algorithm is to find $\mathbf{\Phi}$ and $\mathbf{w}(\theta_s)$ for all steering angles of interest that minimise the SLL. The SLL is defined as the largest pattern magnitude in a pre-defined sidelobe region relative to the main-beam peak. Since $\mathbf{b}(\theta)$ is a function of $\mathbf{\Phi}$, both $\mathbf{w}(\theta_s)$ and $\mathbf{b}(\theta)$ must be optimised over all steering angles θ_s and pattern angles θ. However, the problem can be greatly simplified if $\mathbf{w}(\theta_s)$ is chosen as the complex conjugate of $\mathbf{b}(\theta_s)$. This leads to

$$q(\theta) = \left|\mathbf{b}(\theta_s)^{\mathrm{H}}\mathbf{b}(\theta)\right|. \tag{6.2}$$

This expression is similar in form to the coherence criterion from CS theory (Chapter 2, 5).

In order to validate the compressive array algorithm for an $M = N$ array against a Chebyshev array (Section 6.4.1), it is necessary to define a ULA with the same patterns as the corresponding compressive array. To do this, consider the combined effect of the sensing matrix $\mathbf{\Phi}$ and the beamforming weights $\mathbf{w}(\theta_s)$. Substituting $\mathbf{b}(\theta) = \mathbf{\Phi}\mathbf{a}(\theta)$ into (6.2) gives

$$q(\theta) = \left|\mathbf{a}^{\mathrm{H}}(\theta_s)\mathbf{\Phi}^{\mathrm{H}}\mathbf{\Phi}\mathbf{a}(\theta)\right|. \tag{6.3}$$

Since $\mathbf{a}(\theta)$ is the steering vector for a ULA, define a set of equivalent weights for a conventional ULA as $\mathbf{v}^{\mathrm{T}}(\theta_s) = \mathbf{a}^{\mathrm{H}}(\theta_s)\mathbf{\Phi}^{\mathrm{H}}\mathbf{\Phi}$, or

$$\mathbf{v}(\theta_s) = \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}^{*}\mathbf{a}^{*}(\theta_s), \tag{6.4}$$

where * represents complex conjugation. This gives

$$q(\theta) = \left|\mathbf{v}(\theta_s)^{\mathrm{T}}\mathbf{a}(\theta)\right|, \tag{6.5}$$

which is similar in form to (6.1). Applying the weights $\mathbf{v}(\theta_s)$ to a length-N ULA, as illustated in Figure 6.1(b), will result in the same patterns as the corresponding compressive array.

While varying excitations is the only way to control transmit beamforming, digital beamforming techniques on reception are not limited to weight-and-sum beamforming. For example, super-resolution methods such as MVDR and CS-based algorithms have been successfully applied to compressive arrays for DF (Chapter 4) [11].

6.3 CONTINUOUS SLL OPTIMISATION

6.3.1 Problem formulation

The use of optimised codebooks to design compressive arrays for low SLL is severely hampered by the fact that array pattern control is only possible at N angles (Chapter 4). Even if the discrete $M \times N$ codebook $\mathbf{B}$ has the lowest possible coherence, there is no way of predicting the resulting array patterns in-between the N angles. Simply increasing the number of sampling angles in $\mathbf{A}$ will not suffice since this would result in $\mathbf{B} = \mathbf{\Phi A}$ from (4.8) being underdetermined.

In order to exercise sidelobe control across continuous steering and pattern angles, it is necessary to reformulate the optimisation problem in terms of $\mathbf{\Phi}$, since the sensing matrix with MN discrete complex weights determines the continuous compressed steering vectors via $\mathbf{b}(\theta) = \mathbf{\Phi a}(\theta)$.

With the definition of the excitation weights as the complex conjugates of the compressed steering vectors (Section 6.2.2), the SLL can be computed from

$$\begin{aligned} \text{SLL} &= \max_{|\theta_a - \theta_s| \geq \theta_{\text{SLL}}} \left|\mathbf{b}(\theta_s)^{\text{H}}\mathbf{b}(\theta_a)\right|, \\ \theta_s &\in [\theta_{s1}, \theta_{s2}], \quad \theta_a \in [\theta_{a1}, \theta_{a2}], \end{aligned} \tag{6.6}$$

where θ_{SLL} is the specified start of the sidelobe region relative to the steering angle, θ_s are the steering angles, θ_a are the array pattern angles, $[\theta_{s1}, \theta_{s2}]$ is the steering range, and $[\theta_{a1}, \theta_{a2}]$ is the field of view (typically, $[-90°, 90°]$ for ULAs and $[-180°, 180°]$ for circular arrays). When designing compressive feed networks for ULAs, the start of the sidelobe region is defined to be constant in $\sin(\theta)$ space, with the result that it changes with steering angle (θ_s) in θ space.

Note that $\mathbf{b}(\theta_s)^{\text{H}}\mathbf{b}(\theta_a) = \left[\mathbf{b}(\theta_a)^{\text{H}}\mathbf{b}(\theta_s)\right]^*$ and therefore $\left|\mathbf{b}(\theta_s)^{\text{H}}\mathbf{b}(\theta_a)\right| = \left|\mathbf{b}(\theta_a)^{\text{H}}\mathbf{b}(\theta_s)\right|$, which explains the symmetry observed in the pattern plots of Section 6.4. Thus only unique combinations of θ_s and θ_a need to be considered in the optimisation, which significantly reduces the problem complexity. Unique combinations of θ_s and θ_a are given by $\theta_a - \theta_s \geq \theta_{\text{SLL}}$ (the upper triangular sidelobe regions in Figures 6.3 and 6.4).

Define the continuous SLL optimisation problem as

$$\begin{aligned} &\min_{\mathbf{\Phi}} \max_{\theta_a - \theta_s \geq \theta_{\text{SLL}}} \frac{\left|[\mathbf{\Phi a}(\theta_s)]^{\text{H}}[\mathbf{\Phi a}(\theta_a)]\right|}{R(\theta_s, \theta_a)} \\ &\text{subject to} \\ &\|\mathbf{\Phi a}(\theta_s)\|^2 = 1, \quad \theta_s \in [\theta_{s1}, \theta_{s2}], \quad \theta_a \in [\theta_{a1}, \theta_{a2}], \end{aligned} \tag{6.7}$$

where $R(\theta_s, \theta_a)$ is the sidelobe mask. This mask is used to allow the sidelobe requirements to vary over the sidelobe region. Note that the mask values may be specified independently for both steering and pattern angles, which leads to a number of novel designs as discussed in Section 6.4.

The condition $\|\mathbf{b}(\theta_s)\|^2 = 1$ is necessary to ensure that a peak is located at θ_s when the beam is steered in that direction. This constraint is equivalent to saying that the sum of the subarray pattern powers should equal 1 across the steering range. When all compressed steering vectors have equal length, an absolute dot product with any other compressed steering vector ($\left|\mathbf{b}(\theta_s)^{\text{H}}\mathbf{b}(\theta_a)\right|$, $a \neq s$) is guaranteed to be no larger than the absolute dot product of a compressed steering vector with itself ($\left|\mathbf{b}(\theta_s)^{\text{H}}\mathbf{b}(\theta_s)\right| = \|\mathbf{b}(\theta_s)\|^2$), resulting in a peak in the desired steering direction. The choice of which length the compressed steering vectors should equal is arbitrary and does not affect the final patterns; a value of 1 is chosen here for convenience. This condition is only enforced on the compressed steering

vectors in the steering range of interest, which allows the subarray patterns to be suppressed outside the steering range. As long as the compressed steering vectors outside the steering range have lengths less than 1, the peaks of the steered beams will remain correct. In the algorithm to follow, this constraint is relaxed so that the lengths of the compressed steering vectors are within ±0.01 dB because this relaxation leads to faster convergence.

Similar to the discrete coherence optimisation problem in (3.11), the continuous formulation in (6.7) is non-convex and is not guaranteed to converge to the global minimum. Even so, local optimisers produce discrete codebooks with coherence values that have not been improved upon by any other method (Chapter 5) [16, 66], so this approach is reasonable.

Unfortunately, optimising the patterns using the formulation above would involve evaluating an infinite number of combinations of steering and pattern angles. The angles are thus discretised to provide a finite set of angles over which the SLL must be optimised. This is done by defining a set of FN sampling angles for both the steering and pattern angles, where F is the oversampling factor. These angular points are uniformly spaced in $\sin(\theta)$ space over $[-1,1)$ for ULAs to compensate for the $\sin(\theta)$ factor in (3.1), and in θ space over $[-180°,180°)$ for circular arrays as the elements are uniformly distributed around the circle. Although FN sampling angles are defined, only those within the steering range and sidelobe regions are considered in the optimisation.

Since the pattern values must be evaluated over all steering angles, the problem is inherently combinatorial in nature, with up to $O\left\{\binom{FN}{2}\right\}$ combinations to consider. The oversampling factor F allows a compromise to be made between angular grid resolution (which influences the SLL) and realistic run times. A value of $F = 2M$ was empirically found to present a good compromise between achieved SLL and run time. Using $F = 2M$ means that the resulting number of sampling angles, $2MN$, is equal to the number of control variables, namely the real and imaginary parts of the MN elements in $\mathbf{\Phi}$.

In order to ensure that the calculated performance is accurate, the final beamwidth and SLL values in the results below were calculated using 16 times more sampling points [38]. The 3-dB points in the beamwidth calculations were found using cubic spline interpolation [38].

6.3.2 Algorithm and implementation

The approach to minimising SLL in terms of $\mathbf{\Phi}$ is to sequentially approximate the max operator similar to (5.6) and use a general-purpose constrained non-linear solver for the subproblems. The goal function to be minimised is given by

$$g(\mathbf{\Phi}) = \alpha \left[\frac{1}{C} \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left|(\mathbf{\Phi}\mathbf{a}_s)^{\text{H}}(\mathbf{\Phi}\mathbf{a}_a)\right|}{\alpha R_{s,a}} \right)^p \right]^{1/p}, \tag{6.8}$$

subject to the constraints

$$c_1(\mathbf{\Phi}) = \|\mathbf{\Phi}\mathbf{a}_s\|^2 - 10^{0.01/10} \leq 0 \quad \forall s \quad \text{and} \tag{6.9}$$

$$c_2(\mathbf{\Phi}) = 10^{-0.01/10} - \|\mathbf{\Phi}\mathbf{a}_s\|^2 \leq 0 \quad \forall s, \tag{6.10}$$

where the subscripts s and a refer to variables sampled at the angles θ_s and θ_a, and C is the number of terms in the summation. Including the factor $1/C$ and raising the sum to $1/p$ maintains proper scaling of the problem. The function is of the form $(\sum_N |x_n|^p)^{1/p}$ which approximates the ∞–norm by the p–norm as $p \to \infty$ [66]. The inequality-constraint functions for the unit-length requirements in (6.7) are provided in (6.9) and (6.10).

Since the dot products scaled by the sidelobe mask are typically smaller than one, raising them to large powers may produce results smaller than the lower limit of numerical representation, resulting in underflow of some of the terms in the summation. A scaling factor α, which does not affect the final function value, is thus introduced to scale the terms in the summation to minimise the incidences of underflow. The terms in the summation in (6.8) can be written as $(x_{s,a}/\alpha)^p$, and the goal is to ensure that these terms are as large as possible. Overflow during the intermediate computations can be avoided by setting the sum equal to the largest representable value and assuming that all terms are equal to the largest term, giving

$$\sum_C \left(\frac{x_{\max}}{\alpha}\right)^p = C\left(\frac{x_{\max}}{\alpha}\right)^p = \nu_{\max}, \tag{6.11}$$

where $\nu_{\max}$ is the largest machine-representable number. Solving for α gives

$$\alpha = \frac{x_{\max}}{(\nu_{\max}/C)^{1/p}}. \tag{6.12}$$

The sequential quadratic programming (SQP) algorithm [72] in MATLAB was chosen to solve the subproblems given by (6.8) to (6.10) for increasing values of p. The value of p is incremented via

$$\begin{aligned} p^{(k)} &= \min\left(p_{\max},\ 2\left\lceil \frac{rp^{(k-1)}}{2}\right\rceil\right), \\ k &= 2, 3, \ldots,\quad p^{(1)} = 2,\quad r > 1, \end{aligned} \tag{6.13}$$

where r is the power multiplication factor, k is the subproblem number, and $p_{\max}$ is the maximum value of p. For each subproblem, p is incremented to the next even number greater or equal to $rp^{(k-1)}$. Restricting p to even values ensures that (6.8) is smooth (Chapter 5). For a particular problem, the value of $p_{\max}$ should be increased until no further improvement in SLL is observed, after which the value of r is then reduced until no further improvement in SLL is observed. Values of $p_{\max} = 512$ and $r = 1.1$ have been found to produce good solutions for a wide range of problems and were used to obtain the results in Section 6.4.

The initial sensing matrix $\boldsymbol{\Phi}^{(0)}$ was chosen with real and imaginary parts drawn from a Gaussian distribution with variance $1/(2N)$ so that $\mathbb{E}[\|\boldsymbol{\phi}_m\|] = 1$, where $\boldsymbol{\phi}_m$ are the rows in $\boldsymbol{\Phi}$ [13]. The minimum step size was set to 10^{-10}, the step size used for discrete coherence optimisation in Chapter 5. The first-order optimality and constraint convergence criteria were kept at their default values of 10^{-6}. The number of iterations per subproblem was limited to a maximum of 10^5. Gradients of goal and constraint functions are derived analytically in Addendum A.2. This is important since using finite differences is computationally inefficient. Each finite-difference calculation would require calculating a large number of dot products twice, whereas the dot products only have to be calculated once to evaluate both the goal function and an analytical goal-gradient expression.

6.4 RESULTS

The results obtained for a number of test problems are outlined below, after a description of the conditions under which the results were obtained.

Isotropic antenna elements are assumed. However, the procedure is general enough that the steering vectors may be specified in terms of arbitrary element patterns including, for example, simulated or measured embedded element patterns.

Angles and beamwidths for ULAs are given in $u = \sin(\theta)$ space. The algorithm was implemented in MATLAB R2016b and run on machines with two 6-core 2.30-GHz Intel Xeon E5-2630 processors and 32 GB of memory each. In all cases, an oversampling factor of $F = 2M$ was used.

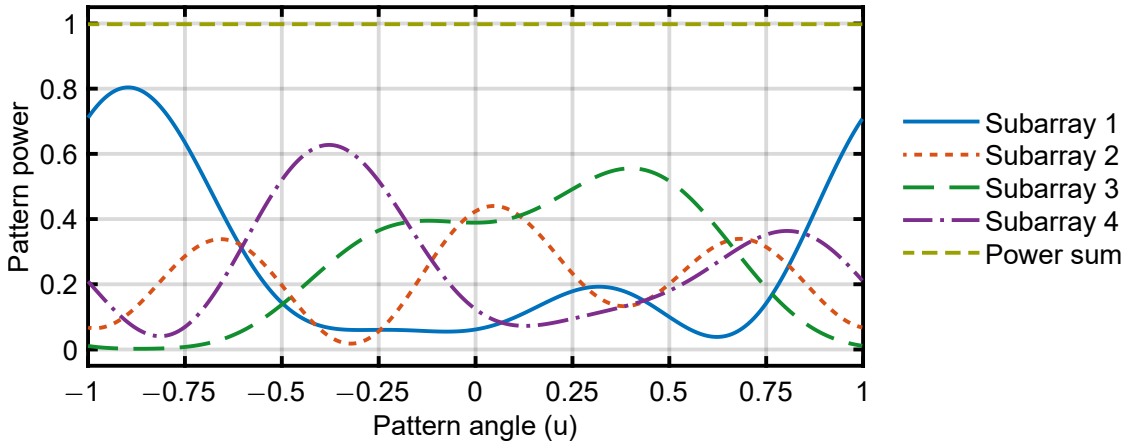


Figure 6.2. Subarray power patterns and their sum for the $M = N = 4$ compressive array in Section 6.4.1. Adapted from [1], ©2018 IEEE.

The results below are summarised in Table 6.1. The SLL and beamwidth results for the compressive arrays are for the designs with the lowest SLL from ten independent runs of the algorithm. Only the worst (i.e. largest) 3-dB beamwidths over the steering ranges are presented. The run times indicated are for the algorithm to run to $p = p_{\max}$ even if the best SLL was obtained before then.

Reduced-control arrays can be considered to either reduce the number of beamforming controls for a given aperture or to increase the aperture for a given number of beamforming controls. Under the reasonable assumption that the cost of an array is primarily determined by the number of beamforming controls (Section 3.3.1) [3], the second perspective is more useful when comparing reduced-control and conventional arrays for a given cost. Section 6.4.3 below thus compares a compressive array with M controls and N antenna elements to an M-element conventional array because an N-element conventional array would cost approximately N/M more.

6.4.1 $M = N$, uniform sidelobe mask

The first test case considers a feed network with $M = 4$ outputs for a ULA with $N = 4$ elements, the start of the sidelobe region at $u_{\text{SLL}} = 0.5$, and a uniform sidelobe mask. Since $M = N$, a Chebyshev excitation provides the optimum result for the conventional weights $\mathbf{v}(\theta_s)$, where optimality refers to the lowest possible SLL for a given main-beam region, or the smallest beamwidth for a given SLL [34]. This problem thus tests whether the proposed algorithm is able to approach the known optimum.

For the purpose of validation, the steering range was defined as $u_s \in [-1, 1)$ since a ULA has patterns that translate all the way to endfire, though with a grating lobe at endfire when half-wavelength spacing is used. When calculating SLL, the sidelobe region range was adjusted to ensure that that grating lobes were ignored at extreme steering angles as is done for Chebyshev arrays.

The compressive array achieved an SLL of -16.98 dB. The Chebyshev excitation for an SLL of -16.98 dB is $|\mathbf{v}(\theta_s)| = [0.6669,1,1,0.6669]^{\text{T}}$. The normalised magnitudes of the equivalent weights $\mathbf{v}(\theta_s)$ for the compressive array are the same, with a maximum deviation of 9.89×10^{-7} from the Chebyshev excitation over all weights and steering angles. This result demonstrates that the compressive array algorithm can achieve comparable results to the known optimum.

Figure 6.2 shows the radiation patterns of each of the M subarrays in the compressive array. The patterns are shown as power values, which add to 1 at each angle as shown and as required by (6.7).

Table 6.1. Results for Compressive Array Designs over Ten Runs with Comparisons to Existing Approaches. Taken from [1], ©2018 IEEE.

	Compressive array design algorithm				Comparison		
Problem (Section)	SLL (dB)	$BW_{3\,dB}$	Min. run time (s)	Max. run time (s)	Approach	SLL (dB)	$BW_{3\,dB}$ (u)
$M = N$ (6.4.1)	−16.98	0.4994 (u)	10.4	16.3	Chebyshev	−16.98	0.4994
Uniform mask (6.4.2)	−2.89	0.0942 (u)	4 553.8	14 497.3	IFT [38]	−2.47	0.1017
Soft null (6.4.3)	−19.85	0.2370 (u)	448.5	11 610.5	Length-M ULA [15]	−19.76	0.2556
Hard null (6.4.4)	−13.27	50.1°	14 849.2	27 296.4	—		
Small steering range (6.4.5)	−34.20	0.0382 (u)	3 657.7	3 806.1	Dual-transform [2]	−32.49	0.0398

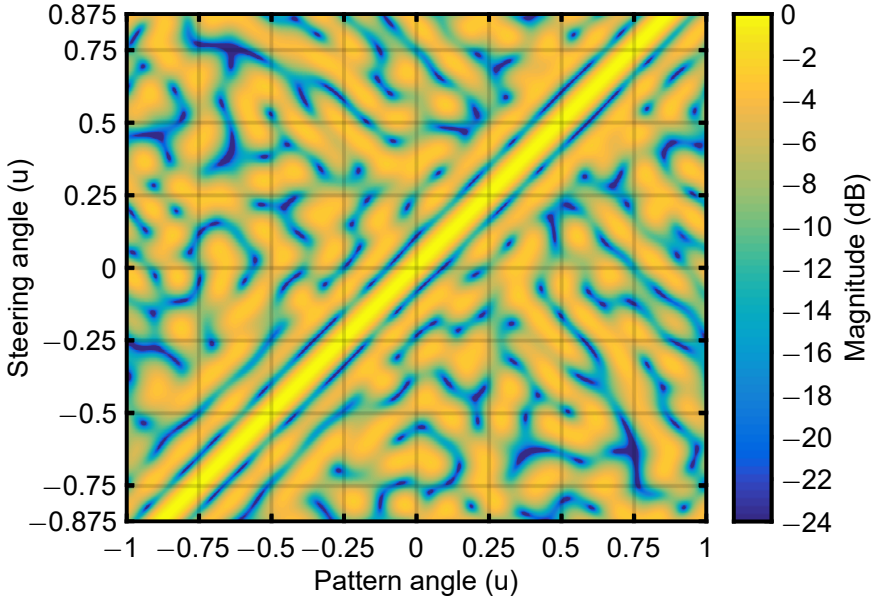


Figure 6.3. Normalised patterns as a function of steering angle for the uniform-mask compressive array in Section 6.4.2. Taken from [1], ©2018 IEEE.

The subarray patterns are simply the M individual elements in the compressed steering vector, since each element in $\mathbf{b}(\theta)$ represents the response of the array at the mth beamforming control. Even though the excitations at the N elements are equivalent to those of a Chebyshev array, the operating principle is different. A weighted ULA with M isotropic elements samples the entire angular domain at each element. In the compressive array, each subarray samples the angular space differently. The subarrays are able to scan the entire range of interest when combined, but individually, some subarrays favour certain angles. The crucial observation is that there is no angle at which all of the subarray gains are low because that would make steering in that direction impossible.

The ability of compessive arrays to implement shaped subarray patterns can be exploited to suppress the subarray patterns where no steering is desired (Section 6.4.3), or to suppress interference from a particular direction (Section 6.4.4).

6.4.2 $M < N$, uniform sidelobe mask

The true compressive array design evaluated here considers a feed network with $M = 4$ for a ULA with $N = 16$, the start of the sidelobe region at $u_{\text{SLL}} = 0.046875$, and a uniform sidelobe specification. The steering range was limited to $|u_s| \leq 0.875$ or $|\theta_s| \leq 61°$ since $|\theta_s| \leq 60°$ is already considered a wide steering range and steering to endfire presents a number of practical challenges [2]. This problem illustrates the case where beamwidths similar to that of a large, filled array are desired at the expense of higher SLL, a typical consideration for DF arrays.

The results are compared to a weighted thinned linear array (Section 3.3.2) designed using the IFT [9,38]. The IFT does not allow the steering range to be limited, so this example also serves to illustrate the performance improvements arising from the ability of compressive arrays to limit the steering range.

Figure 6.3 shows the patterns as a function of steering angle for the compressive array design. Significantly, the patterns do not translate with steering angle, but vary significantly as the main beam

is steered. Note that beamwidth and SLL values provided below are achieved at all steering angles despite the significant pattern variation. This is an example of non-translational patterns which arise in compressive arrays. One of the main consequences of non-translational patterns is that steering of the main beam is more complex as the excitation amplitudes and phases vary non-linearly, unlike translational patterns where only linear phase variation is required.

The compressive array has an SLL of −2.89 dB and a worst-case 3-dB beamwidth of 0.0942 over the specified steering range, while the IFT achieved an SLL of −2.47 dB with a 3-dB beamwidth of 0.1017 as the best result from ten runs. The compressive array design improves on the weighted thinned design by 0.42 dB in SLL and 7.4% in beamwidth (in u). This improvement is achieved by exploiting the additional degrees of freedom provided by the fact that the M feed network outputs are functions of all N antenna elements, which includes the ability to exploit a limited steering range. On the other hand, in a weighted thinned array each of the M outputs is only a function of a single antenna, and the IFT cannot exploit a limited steering range.

Compressive arrays with uniform sidelobe masks would be useful as an alternative to thinned or weighted thinned arrays where SLL is crucial, since the additional degrees of freedom in the feed network allow designs with lower SLL for the same number of beamforming controls.

6.4.3 Soft stationary null steering

Section 6.4.2 showed that a compressive array can have patterns that change as a function of steering angle. In order to exploit this property, the sidelobe mask $R_{s,a}$ can be specified independently for both steering and pattern angles, allowing the shape of the sidelobe regions to be controlled.

This problem introduces the idea of a soft stationary null, where the sidelobes are reduced at a fixed pattern angle over all steering angles except those near the soft null itself. This allows a beam to be steered in the direction of the soft null, while suppressing interference from this direction when steering to other angles. This type of array would be useful in scenarios where a high-power source must be suppressed in order to receive weak signals from other directions, but where it is also necessary to monitor the high-power source itself. This situation would occur when low-power radios and a high-power radar must be monitored by a receiver system.

Consider a feed network with $M = 8$ for a ULA with $N = 16$, the start of the sidelobe region at $u_{\text{SLL}} = 0.25$, and a steering range of $|u_s| \leq 1 - u_{\text{SLL}} = 0.75$. This limit prevents any portion of the grating lobes from appearing at extreme steering angles in the case of half-wavelength element spacing [2].

A soft null at broadside is required to be 20 dB below the SLL. The nature of the soft null means that it is only present when $|u_s| \geq 0.25$, so that the null does not enter the predefined main-beam region.

The resulting patterns are shown in Figure 6.4. The achieved SLL is −19.85 dB and the 3-dB beamwidth ranges from 0.1837 to 0.2370 depending on the steering angle. The beamwidths narrow near broadside steering due to the nulls at the start of the sidelobe region ($u = \pm 0.25$). The worst pattern level in the soft-null region is −39.87 dB, or 20.02 dB below the SLL. Due to the reduced sidelobes at broadside steering, the directivity peaks near broadside steering at 10.14 dB, while the minimum directivity over the steering range is 8.96 dB.

The results were compared to an M-element ULA with the same specifications to ensure that both cases have the same number of beamforming controls. Weights for the ULA were redesigned for each steering angle using the covariance matrix inversion method [2, 15]. The required SLL was specified as −19.85 dB, and the achieved worst-case SLL is −19.76 dB (since no u_{SLL} is specified, the SLL is

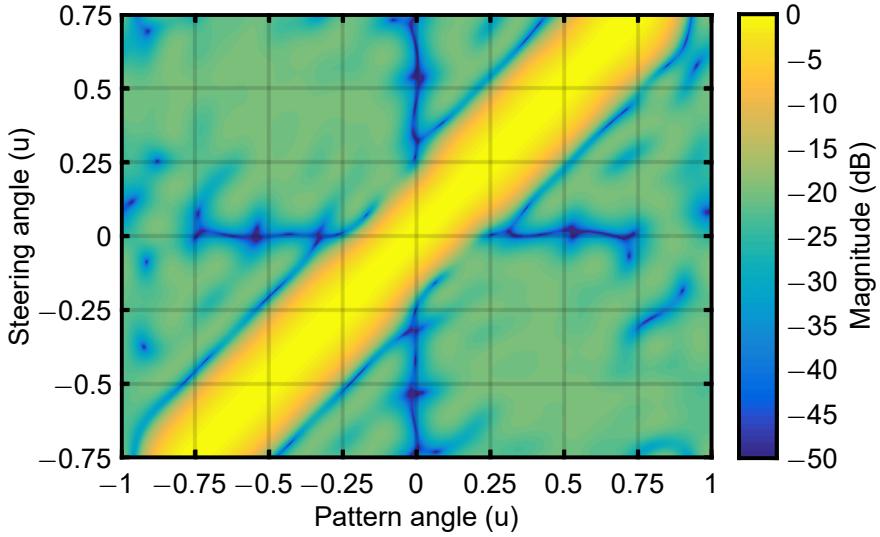


Figure 6.4. Normalised patterns as a function of steering angle for the soft-null compressive array in Section 6.4.3. Taken from [1], ©2018 IEEE.

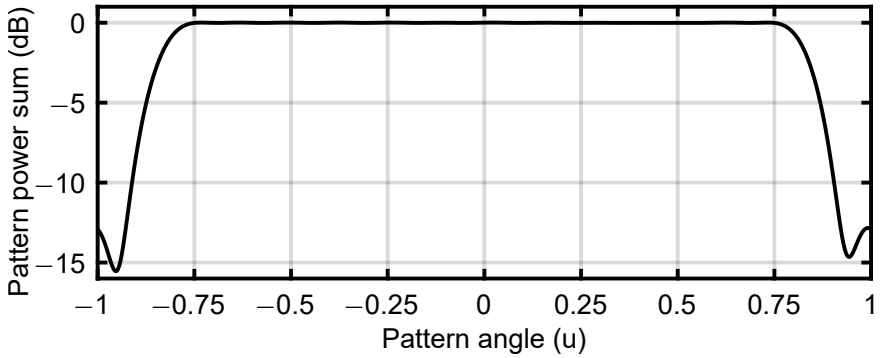


Figure 6.5. Sum of the subarray pattern powers for the soft-null compressive array in Section 6.4.3. Taken from [1], ©2018 IEEE.

defined here as the largest pattern value outside the main beams). The largest pattern value in the soft null region is −39.84 dB below the peaks, or 20.08 dB below the achieved SLL. The 3-dB beamwidths range from 0.2465 to 0.2556. Thus, the compressive array improves on the ULA by 7.3% in worst-case beamwidth. The directivity of the ULA ranges from 8.69 dB to 8.85 dB, which is consistently lower than that of the compressive array.

Figure 6.5 shows the sum of the M subarray pattern powers over pattern angle, which equals 0 dB ± 0.01 dB in the steering range as required by (6.9) and (6.10). The subarray patterns are suppressed in the out-of-scan regions ($|u| > 0.75$), which explains the higher directivity of the compressive array over the ULA.

The design of an $M = 8$, $N = 16$ soft-null compressive array, and an 8-element soft-null ULA shows that increasing N for the same M increases the control over the array patterns.

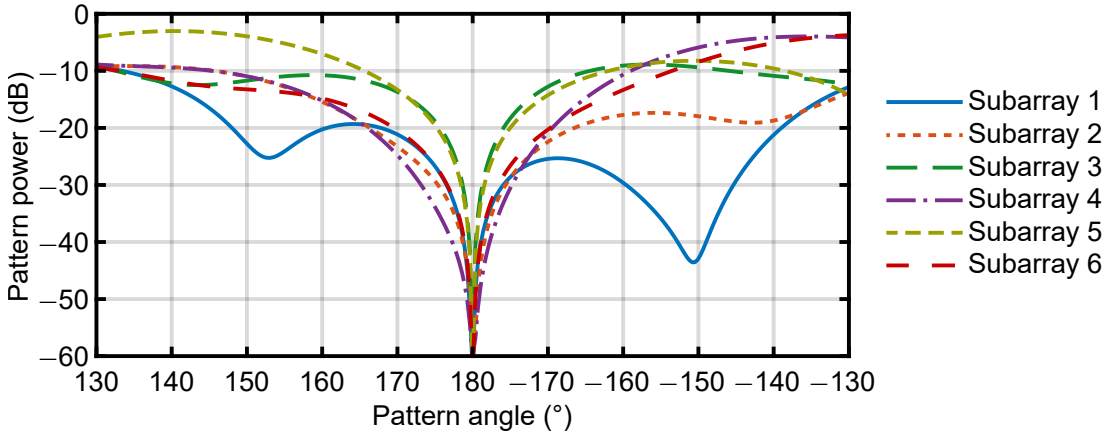


Figure 6.6. Superimposed subarray patterns in the out-of-scan region for the compressive circular array in Section 6.4.4. Adapted from [1], ©2018 IEEE.

6.4.4 Hard stationary null steering

Limiting the steering range of a compressive array means that the subarray patterns may be constrained in any manner outside the steering range. Consider an application where an interfering signal is present from a known direction with significantly higher power than the signal(s) of interest. An example of such a scenario is a passive bistatic radar where the receiver sees a direct-path signal from the transmitter of opportunity that has a much higher signal strength than the reflected signal from the target, with the direct-path signal as much as 90 dB stronger than the reflected signal [74]. A null formed via digital beamforming (e.g. a soft null) may not suffice, since most of the receivers' available dynamic range will be utilised for sampling the interfering signal to avoid saturation, with little dynamic range left for the signal(s) of interest. A hard stationary null solves this problem by placing a null in the subarray patterns themselves so that the interfering signal is suppressed before sampling.

Consider the constraint function

$$c_{3,m}(\boldsymbol{\phi}_m) = \|\boldsymbol{\phi}_m \mathbf{a}(\theta_w)\|^2 - 10^{W/10} \leq 0 \quad \forall m, \tag{6.14}$$

where $\boldsymbol{\phi}_m$ is the mth row of $\boldsymbol{\Phi}$ such that $b_m(\theta_w) = \boldsymbol{\phi}_m \mathbf{a}(\theta_w)$, θ_w is the desired null direction with $\theta_w \notin [\theta_{s1}, \theta_{s2}]$, and W is the desired magnitude in dB of the mth subarray pattern in the null direction. The null level is not specified relative to the subarray pattern peak, since a subarray pattern does not necessarily have a single distinct peak (Section 6.4.1). It is specified simply as the value of b_m in the null direction.

Consider the design of a feed network with $M = 6$ for a uniform circular array (UCA) with $N = 9$ elements with half-wavelength spacing between adjacent elements. The steering range is $\theta_s \in [-130°, 130°]$ with a null of magnitude $W = -60$ dB at 180°. The start of the sidelobe region is $\theta_{\text{SLL}} = 50°$.

Figure 6.6 shows the resulting subarray patterns, superimposed, in the out-of-scan region with the hard null visible at 180°. The achieved SLL over all steering angles is −13.27 dB and the 3-dB beamwidth ranges from 42.9° to 50.1°.

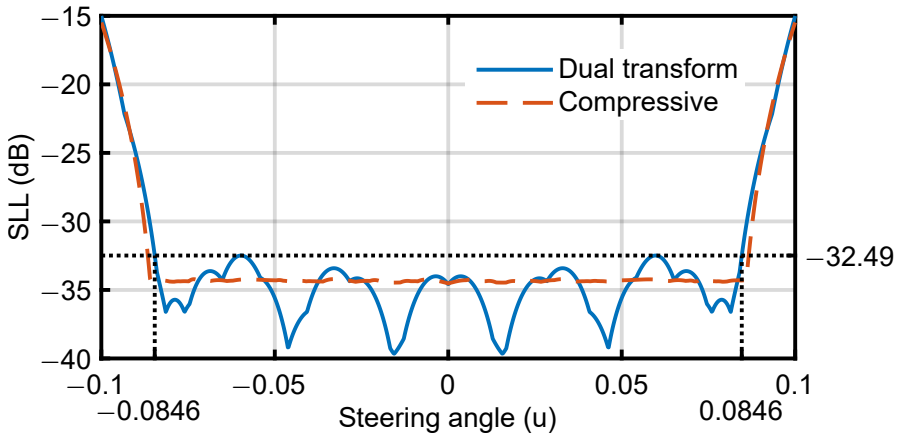


Figure 6.7. SLL as a function of steering angle for the dual-transform and compressive arrays in Section 6.4.5. Taken from [1], ©2018 IEEE.

6.4.5 Small steering range array

A dual-transform completely overlapped subarray system (Section 3.3.3) with $M = 8$ and $N = 64$ is described in [2]. N antenna elements in a ULA are the inputs to an $N \times N$ Butler matrix, but only M outputs corresponding to the beams closest to broadside are used. These M outputs are then passed through an $M \times M$ Butler matrix to produce the M subarray outputs. The subarray outputs are weighted by a 40-dB length-M Chebyshev window. The entire feed network can be described by a fully populated $M \times N$ sensing matrix since each subarray output is a function of the signals at all the antenna elements.

The dual-transform SLLs as a function of steering angle are shown in Figure 6.7 (with SLL defined here as the largest pattern value outside the main beam for a given steering angle). The specified range of steering angles is $u_s \in [-0.1094, 0.1094]$, but the sidelobes are high at the extremities. As indicated in Figure 6.7, the array achieves a worst-case SLL of −32.49 dB over a range of $u_s \in [-0.0846, 0.0846]$ which presents a more useful steering range. The worst 3-dB beamwidth over a steering range of $u_s \in [-0.0846, 0.0846]$ is 0.0398, and the directivity ranges from 16.82 dB to 16.88 dB over the same range.

A compressive array was designed for a slightly wider steering range of $u_s \in [-0.0859, 0.0859]$, with a sidelobe region starting at $u_{\text{SLL}} = 0.0508$ to match the smallest start of a −32.49-dB sidelobe region in the dual-transform array, which was found to be 0.0522. The best SLL achieved for the compressive array was −34.20 dB, an improvement of 1.71 dB over the dual-transform array. The worst-case 3-dB beamwidth for the compressive array was 0.0382 which also improves on the dual-transform design. Finally, the directivity varied from 16.97 dB to 17.15 dB which is, again, slightly better than that of the dual-transform array.

Figure 6.7 compares the SLL of the compressive and dual-transform arrays as a function of steering angle. The SLL of the dual-transform array has significant variation in the steering range, compromising SLL at certain steering angles in favour of others. On the other hand, the compressive array achieves a lower overall SLL by maintaining nearly-uniform SLLs across the entire steering range.

Figure 6.8 shows the patterns for the compressive and dual-transform arrays at steering angles with the worst SLL, namely $u_s = -0.0859$ and $u_s = -0.0846$, respectively. While the dual-transform array has some sidelobes that are lower than those of the compressive array, this characteristic leads to a

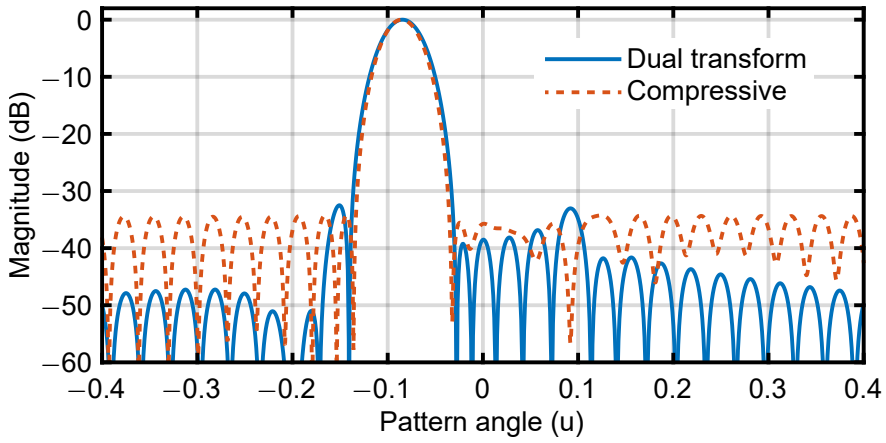


Figure 6.8. Compressive array pattern at $u_s = -0.0859$ steering angle and dual-transform array pattern at $u_s = -0.0846$ steering angle for the limited-steering problem in Section 6.4.5. Taken from [1], ©2018 IEEE.

higher SLL and a wider main beam than the compressive array which has a more uniform pattern in the sidelobe regions.

6.5 CHAPTER SUMMARY

In this chapter, a generalised framework for the optimisation of compressive arrays for low SLL has been proposed. A sequential algorithm for accomplishing this goal, inspired by the coherence optimisation algorithm from Chapter 5, has been described.

The proposed algorithm has been validated against a Chebyshev array and achieved the same optimal result. Compressive arrays have been shown to achieve better results than a weighted thinned array designed using the IFT, and improve on the performance of an existing completely overlapped dual-transform subarray system. A compressive array with a soft null obtained better results than a conventional ULA with the same number of beamforming controls. The increased number of elements of a compressive array is thus shown to enable better control over the shape of array patterns than a conventional array with the same number of controls. A hard-null compressive array shows that the subarray patterns can be constrained outside the steering range to suppress interference in hardware before sampling, for example.

CHAPTER 7 A CHEQUERED NETWORK FOR IMPLEMENTING ARBITRARY FEED NETWORKS

7.1 CHAPTER OVERVIEW

Traditionally, subarray pattern synthesis is linked to the chosen feed network layout, where the aim is, for example, to synthesise flat-topped subarray patterns for a limited steering range (Chapter 3) [2,3,39,40]. In Chapter 6, a generalised framework for designing compressive feed networks for beamforming arrays was presented and shown to improve on existing reduced-control techniques. This improvement was achieved by removing all feed-network constraints and allowing an arbitrary complex-valued feed network response $\mathbf{\Phi}$. However, no details have yet been given of how compressive feed networks could be implemented in hardware. Since no constraints are placed on the feed network response, a technique for implementing a compressive feed network in hardware must be able to implement an arbitrary complex-valued response.

In this chapter, the problem of implementing an arbitrary complex-valued feed network response is addressed by presenting a novel overlapped feed network layout consisting of interconnected couplers and fixed phase shifters. The proposed approach separates the tasks of subarray pattern synthesis and feed network implementation, and enables full exploitation of compressive arrays by not placing any constraints on the feed network response.

While no constraints are placed on the feed network response to be implemented, constraints may be placed on the coupling ratios and phase shifts to ensure realisability. An algorithm is proposed below to implement a desired feed network response while minimising the range of coupling ratios and phase shifts required. Constraints may also be placed on the power delivered to terminations to reduce losses.

Section 7.2 introduces the proposed chequered feed network layout and presents a method for deriving its response. An algorithm for designing the feed network to implement a desired response is presented in Section 7.3. Section 7.4 presents results for some test cases, including measured results for a microstrip implementation of a compressive array with four antenna elements and two subarrays. Finally, a summary of the chapter is provided in Section 7.5.

This chapter is a modified and updated version of a published journal paper [24]. Portions of this chapter are ©2019 IEEE; reprinted, with permission, from [24].

The MATLAB code for generating the main results in this chapter is available online at `https://doi.org/10.24433/CO.2625641.v1`.

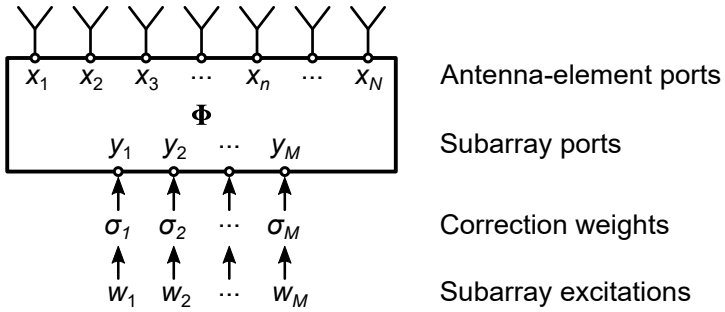


Figure 7.1. A generalised overlapped feed network. Adapted from [24], ©2019 IEEE.

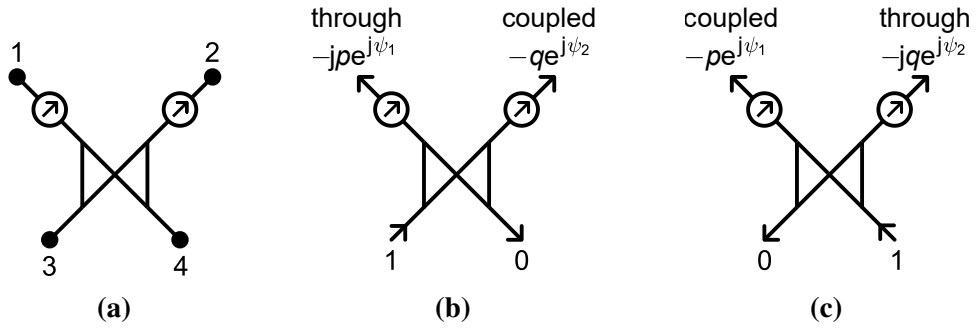


Figure 7.2. Definition of a single tile consisting of a directional coupler and two fixed phase shifters: (a) Tile port definitions. (b) Tile response to a stimulus at port 3. (c) Tile response to a stimulus at port 4. Adapted from [24], ©2019 IEEE.

7.2 THE CHEQUERED NETWORK LAYOUT FOR ARBITRARY FEED NETWORK RESPONSES

7.2.1 Overlapped feed networks

An arbitrary overlapped feed network with M subarrays for N antenna elements is illustrated in Figure 7.1. Following the same notational convention as before, the feed network response is given by the $M \times N$ matrix $\mathbf{\Phi}$, such that the subarray signal vector is related to the antenna-element signal vector by the equation $\mathbf{y} = \mathbf{\Phi x}$.

Ports are labelled according to their signal variables, so that antenna-element ports are labelled x_1 to x_N and subarray ports y_1 to y_M as shown in Figure 7.1. Terminated ports are labelled z_l, where $l \in 1, \ldots, T$, and T is the number of terminated ports which depends on the chosen layout.

Complex-valued subarray excitations w_1 to w_M are applied at the subarrays ports, along with correction weights σ_1 to σ_M. These correction weights are included for more degrees of freedom to realise the desired feed network response. In practice, the weights w_m and σ_m can be combined for a single excitation at each subarray port.

7.2.2 The chequered layout

The proposed network uses a tile consisting of a directional coupler and two fixed phase shifters as its basic element, as illustrated and defined in Figure 7.2, with reciprocity applying. The specific coupler used here is a single-section branchline coupler [60], although the technique is also applicable to other

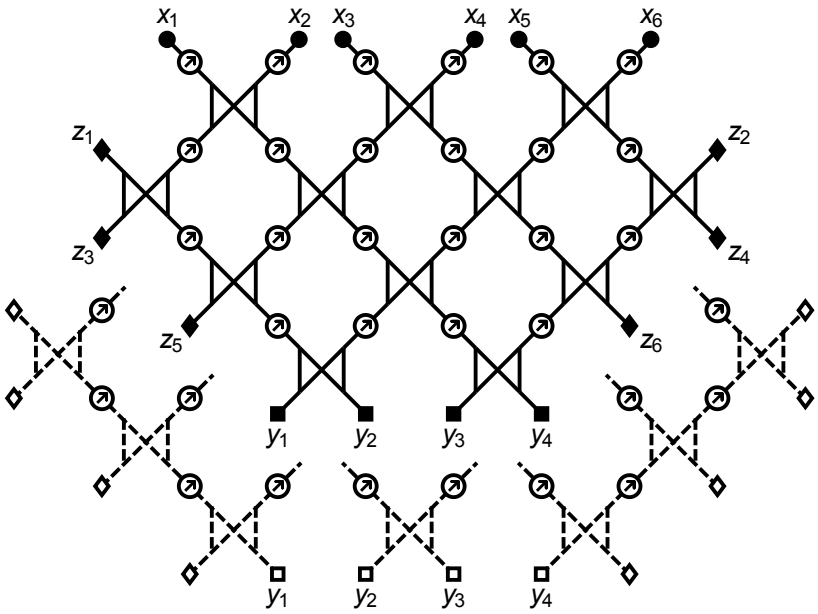


Figure 7.3. An example of the proposed network of couplers and fixed phase shifters for $M = 4$ and $N = 6$ and 4 tile rows, with the additional tiles required for extending the layout to 5 rows shown in dashed lines. Taken from [24], ©2019 IEEE.

couplers with similar port layouts. The phase shifters are fixed and their simplest implementation is varying lengths of transmission line, although other phase shifters with wider bandwidths can also be employed. The relative power delivered to the through and coupled port is p^2 and q^2, respectively, with $p^2 + q^2 = 1$. The coupling ratio is defined as p/q. Coupling ratios above and below 0 dB indicate more power being transferred to the through and coupled ports, respectively. A coupling ratio of 0 dB indicates an equal-split coupler.

The tiles are connected in a chequered pattern to form the feed network, similar to the way couplers are connected in the chess network proposed by Skobelev (Chapter 3) [40, 57]. The layout is planar and requires no physical crossovers. Each row of couplers is identical in Skobelev's chess network, whereas the network proposed here allows the parameters of each tile to be specified independently. The top row of tiles connects to the N antenna-element ports, x_n, and the bottom row connects to the M subarray ports, y_m. All remaining tile ports, z_l, have matched terminations. Sufficient rows are required to ensure that there is at least one signal path from each subarray port to each element port when complete overlap is required. This leads to a diamond-shaped chequered layout as illustrated in Figure 7.3 for $M = 4$ and $N = 6$, with the minimum number of rows to allow complete overlap being 4. Also shown are the additional tiles required to extend the layout to 5 rows.

The minimum number of rows that achieves the desired overlap between subarrays does not guarantee sufficient degrees of freedom to implement the desired response. The number of rows should thus be increased until the desired response can be realised. A procedure for finding the minimum number of rows required to implement a desired feed network response is outlined in Section 7.3.1.

Typically, there will be two or more signal paths between any subarray and element port. For a stimulus at a given subarray port, the various signal paths to a given element port combine via superposition to produce the desired output. The application of both amplitude and phase alterations along the signal paths allows arbitrary complex-valued responses to be realised as long there are enough tile rows to provide the required degrees of freedom.

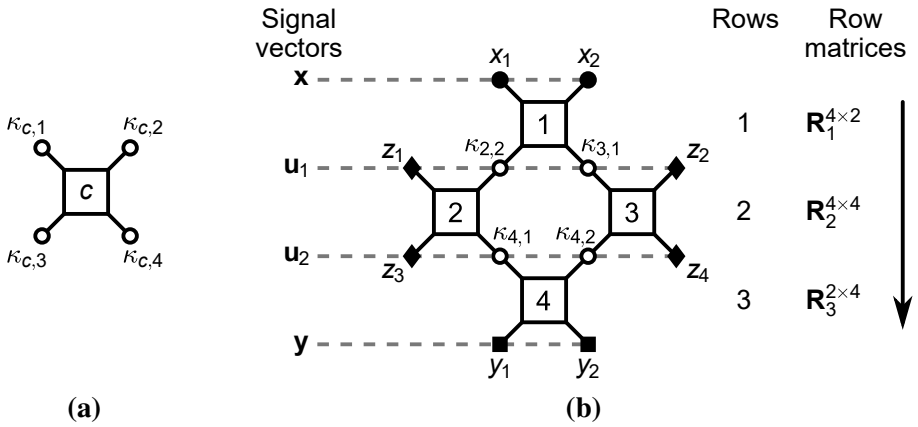


Figure 7.4. Derivation of a 2×2 chequered network: (a) Tile symbol with associated intermediate signal variables. (b) Layout with numbered tiles and intermediate signal nodes. The arrow indicates the signal flow convention used to define the row transform matrices. Taken from [24], ©2019 IEEE.

It is assumed that arbitrary complex-valued excitations, w_1 to w_M, can be applied to the subarray ports. These variable excitations both allow beamforming even though the feed network itself is fixed, and add additional degrees of freedom which can be exploited to help achieve the required response with the smallest number of tiles.

7.2.3 Derivation of the network response

The response of a 2×2 network with three rows is derived below, leading to a procedure for deriving the response of any chequered network.

Figure 7.4(a) presents a symbol for a tile of the form shown in Figure 7.2 along with definitions for the intermediate signals at its ports. The signals flow in a downward direction in Figure 7.4(a) when stimuli are applied at $\kappa_{c,1}$ and $\kappa_{c,2}$. The response of tile c can be expressed in matrix form as

$$\begin{bmatrix} \kappa_{c,3} \\ \kappa_{c,4} \end{bmatrix} = \begin{bmatrix} -\mathrm{j}p_c \mathrm{e}^{\mathrm{j}\psi_{c,1}} & -q_c \mathrm{e}^{\mathrm{j}\psi_{c,2}} \\ -q_c \mathrm{e}^{\mathrm{j}\psi_{c,1}} & -\mathrm{j}p_c \mathrm{e}^{\mathrm{j}\psi_{c,2}} \end{bmatrix} \begin{bmatrix} \kappa_{c,1} \\ \kappa_{c,2} \end{bmatrix} \tag{7.1}$$

$$= \mathbf{K}_c \begin{bmatrix} \kappa_{c,1} \\ \kappa_{c,2} \end{bmatrix}, \tag{7.2}$$

where $\mathbf{K}_c$ is the response matrix for tile c. When a node $\kappa_{c,1}$ or $\kappa_{c,2}$ is connected to a terminated port, the corresponding phase shift $\psi_{c,1}$ or $\psi_{c,2}$ is set to zero since it has no effect on the feed network response.

Figure 7.4(b) shows a 2×2 chequered layout with intermediate signal nodes, inter-row signal vectors, and row transform matrices that will be used in the derivation to follow. The response will first be derived for stimuli applied to the element ports so that signal flow is in the indicated downward direction.

The inter-row signal vector $\mathbf{u}_1$ is defined as the terminated and inter-tile signals between the first and second rows, and is given by

$$\mathbf{u}_1 = \begin{bmatrix} z_1 \\ \kappa_{2,2} \\ \kappa_{3,1} \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\mathrm{j}p_1 \mathrm{e}^{\mathrm{j}\psi_{1,1}} & -q_1 \mathrm{e}^{\mathrm{j}\psi_{1,2}} \\ -q_1 \mathrm{e}^{\mathrm{j}\psi_{1,1}} & -\mathrm{j}p_1 \mathrm{e}^{\mathrm{j}\psi_{1,2}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{7.3}$$

$$= \begin{bmatrix} \mathbf{0}_{1\times 2} \\ \mathbf{K}_1 \\ \mathbf{0}_{1\times 2} \end{bmatrix} \mathbf{x} = \mathbf{R}_1 \mathbf{x}. \tag{7.4}$$

where $\mathbf{0}_{a\times b}$ is an $a \times b$ matrix of zeros, and the row transform matrix for row r is denoted $\mathbf{R}_r$. The next inter-row signal vector $\mathbf{u}_2$ is found to be

$$\mathbf{u}_2 = \begin{bmatrix} z_3 \\ \kappa_{4,1} \\ \kappa_{4,2} \\ z_4 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_2 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{K}_3 \end{bmatrix} \begin{bmatrix} z_1 \\ \kappa_{2,2} \\ \kappa_{3,1} \\ z_2 \end{bmatrix} = \mathbf{R}_2 \mathbf{u}_1. \tag{7.5}$$

The subarray output vector $\mathbf{y}$ is then

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{2\times 1} & \mathbf{K}_4 & \mathbf{0}_{2\times 1} \end{bmatrix} \begin{bmatrix} z_3 \\ \kappa_{4,1} \\ \kappa_{4,2} \\ z_4 \end{bmatrix} = \mathbf{R}_3 \mathbf{u}_2. \tag{7.6}$$

Combining (7.3)–(7.6) gives

$$\mathbf{y} = \mathbf{R}_3 \mathbf{R}_2 \mathbf{R}_1 \mathbf{x}. \tag{7.7}$$

The 2×2 matrix $\mathbf{R}_3\mathbf{R}_2\mathbf{R}_1$ represents the chequered network response from the antenna elements to the subarray ports. Due to reciprocity, reversing the direction of signal flow and finding $\mathbf{x}$ as a function of a stimulus $\mathbf{y}$ results in

$$\mathbf{x} = (\mathbf{R}_3\mathbf{R}_2\mathbf{R}_1)^{\mathrm{T}} \mathbf{y} = \mathbf{R}_1^{\mathrm{T}}\mathbf{R}_2^{\mathrm{T}}\mathbf{R}_3^{\mathrm{T}}\mathbf{y}, \tag{7.8}$$

which shows that the derivation for the reversed signal direction follows the same approach, but uses transposed row transform matrices in reverse order. The scattering parameters between the element and subarray ports are provided by the network response matrix with the scattering parameter S_{y_m,x_n} being element (m, n) of $\mathbf{R}_3\mathbf{R}_2\mathbf{R}_1$.

To find the scattering parameters involving terminated ports, the inter-row signal vector containing the desired terminated port signal must be found, with the signal flow direction depending on whether the scattering parameter is calculated from an element or subarray port. Finding S_{z_3,x_n} and S_{z_4,x_n} in Figure 7.4(b) requires applying stimuli at x_n and calculating $\mathbf{R}_2\mathbf{R}_1$ in that order since the signal flow direction is downward. Conversely, S_{z_1,y_1} will be element (1, 1) in $\mathbf{R}_2^{\mathrm{T}}\mathbf{R}_3^{\mathrm{T}}$ since the stimulus is applied to y_1. The parameters S_{z_1,x_n}, S_{z_2,x_n}, S_{z_3,y_m}, and S_{z_4,y_m} are all zero since the corresponding ports are isolated from each other.

The total aperture illumination $\mathbf{x}$ determines the radiation pattern when the feed network is connected to an antenna array. The radiation amplitude pattern, normalised to the array directivity, remains unchanged when $\mathbf{x}$ is multiplied by an arbitrary complex constant $A\mathrm{e}^{\mathrm{j}\alpha}$ as only the relative amplitudes and phases of the elements are of importance. Applying an arbitrary constant phase shift α to $\mathbf{x}$ gives $\mathbf{y}\mathrm{e}^{\mathrm{j}\alpha} = \mathbf{R}_3\mathbf{R}_2\mathbf{R}_1\mathbf{x}\mathrm{e}^{\mathrm{j}\alpha}$, where $\mathrm{e}^{\mathrm{j}\alpha}$ can be incorporated into any of the row matrices. This means that any

Algorithm 7.1. Determining chequered layouts. Adapted from [24], ©2019 IEEE.

1) The smallest number of rows that ensures the existence of at least one signal path between the ports y_m and x_n for each non-zero entry $\phi_{m,n}$ in $\mathbf{\Phi}$ is determined.
2) Algorithm 7.2 is followed in an attempt to find a feasible solution that satisfies the constraints.
3) If no feasible solution is found, one row is added to the network as illustrated in Figure 7.3.
4) Steps (2) and (3) are repeated until a feasible solution is found.

phase shift can be added to or subtracted from all the phase shifts in a row to, for example, reduce the lengths of the transmission lines used as phase shifters.

From the 2×2 chequered network derivation, a simplified procedure can be deduced for deriving the response of any chequered network.

1. Rows are numbered from top to bottom. The inter-row signal vectors are defined with the left-most signal node corresponding to the first element in each signal vector as shown in Figure 7.4(b).
2. The row transform matrices are numbered according to their rows and their sizes are determined. Using the signal flow convention in Figure 7.4(b), the number of rows in each matrix equals the length of the inter-row signal vector below the row, and the number of columns equals the length of the inter-row signal vector above the row.
3. Each row transform matrix is populated by defining a square block diagonal matrix with the row-specific tile responses $\mathbf{K}_c$ as the block diagonal elements in the order in which they appear in the row. For example, $\mathbf{R}_2$ in (7.5) is such a block diagonal matrix. If the row transform matrix is rectangular, the block diagonal matrix is padded with zeros to reach the required size.

7.3 AN ALGORITHM FOR DESIGNING A CHEQUERED NETWORK FOR A DESIRED RESPONSE

7.3.1 Determining the network layout

The procedure to select a layout for a given problem is outlined in Algorithm 7.1, after which the network can be designed using the approach described below.

7.3.2 Single-stage optimisation for a given set of constraints

A chequered network implements a desired feed network response $\mathbf{\Phi}$ when the constraints

$$\sigma_m S_{y_m,x_n} = \phi_{m,n} \quad \forall m, n \tag{7.9}$$

are satisfied, where S_{y_m,x_n} is the scattering parameter between ports y_m and x_n, and σ_m is a complex-valued correction weight applied at subarray port y_m. Additional constraints are added to ensure that the coupling ratios and phase shifts are realisable. After satisfying all the constraints, any remaining degrees of freedom are utilised to minimise the mean power delivered to the terminated ports. The optimisation problem is constraint-driven rather than goal-function-driven in the sense that the network response and realisability criteria are controlled by constraints rather than by the goal function.

The problem described above can be solved using a general-purpose constrained non-linear solver, such as SQP [72]. A local solver such as SQP is not guaranteed to converge to a feasible solution, so

multiple trial points must be used in an attempt to reach a feasible solution [75]. The MultiStart solver in MATLAB offers a convenient way of starting the SQP algorithm from multiple trial points [76].

The optimisation problem is formulated as

$$\min \left[\sum_l \sum_m \left| S_{z_l, y_m} \right|^2 + \sum_l \sum_n \left| S_{z_l, x_n} \right|^2 \right], \tag{7.10}$$

with the constraints

$$\mathrm{Re}\left\{ \sigma_m S_{y_m, x_n} - \phi_{m,n} \right\} \leq \varepsilon \qquad \forall m, n \tag{7.11}$$

$$\mathrm{Im}\left\{ \sigma_m S_{y_m, x_n} - \phi_{m,n} \right\} \leq \varepsilon \qquad \forall m, n \tag{7.12}$$

$$p_c^2 + q_c^2 = 1 \qquad \forall c \tag{7.13}$$

$$p_{\min} \leq p_c \leq p_{\max} \qquad \forall c \tag{7.14}$$

$$0 \leq q_c \leq 1 \qquad \forall c \tag{7.15}$$

$$\psi_{\min} \leq \psi_c \leq \psi_{\max} \qquad \forall c, \tag{7.16}$$

where c are the tile indices. The goal, defined in (7.10), is to minimise the mean power delivered to the terminated ports from both subarray and element ports. Constraints (7.11) and (7.12) ensure that the real and imaginary components of the network response achieve their desired values to within a tolerance ε. The relationship between p_c and q_c is maintained by (7.13). The fact that p_c and q_c are specified individually instead of expressing q_c as $\sqrt{1 - p_c^2}$ allows the algorithm to pass through infeasible regions which improves convergence. The range of coupling ratios is bounded by (7.14), while (7.15) ensures that (7.13) always only has real-valued solutions. The range of phase shifts is bounded by (7.16).

Q trial points are passed to the MultiStart solver which are uniformly distributed over the bounds (7.14) to (7.16), $-1 \leq \mathrm{Re}\{\sigma_m\} \leq 1$, and $-1 \leq \mathrm{Im}\{\sigma_m\} \leq 1$. The final solution is the feasible solution with the lowest goal function value.

On transmission, terminated ports that are not isolated from subarray ports (e.g. ports z_1 and z_2 in Figure 7.4) may have to dissipate large amounts of power. Constraints of the form

$$\sum_l \left| S_{z_l, y_m} \right|^2 \leq P_{\max} \qquad \forall m, \tag{7.17}$$

where $P_{\max}$ is the maximum total relative power delivered from a subarray port to the terminated ports, can be placed on the power delivered to terminated ports. However, the scattering parameters assume that only one port is excited at a time, so the actual power delivered to terminated ports will differ when multiple ports are stimulated simultaneously. Furthermore, the power delivered will depend on the signals at the stimulated ports, so the power delivered will vary with steering angle in the case of a beamforming array. Scattering-parameter-based terminated powers therefore do not reflect the actual losses of a beamforming system in operation, but nevertheless provide a useful way of evaluating the efficiency of a chequered network in a manner which is easy to formulate as a constraint.

Constraints are not placed on the powers delivered to terminated ports on reception since non-uniform aperture illuminations may require that significant power be delivered to terminated ports to form the desired illuminations across the elements. The power delivered to terminated ports is a function of the aperture efficiency of the illuminations and is thus not necessarily indicative of the efficiency of the chequered network implementation. However, the aperture illuminations do not prevent the full utilisation of at least one antenna element. The element port with the lowest loss (i.e. with the least

Algorithm 7.2. Optimising chequered networks. Adapted from [24], ©2019 IEEE.

1) An optimisation is performed from Q trial points without constraints on the coupling ratios.
2) Couplers from step (1) with $p_c > p_{\text{short}}$ are shorted.
3) Optimisation is repeated with incrementing $p_{\min}$ and decrementing $p_{\max}$, until no feasible solution is found.
4) Optimisation is repeated with incrementing $\psi_{\min}$, until no feasible solution is found.
5) A final optimisation is performed from Q trial points with the final constraints.

power delivered to the terminated ports) can therefore be used to provide an indication of the efficiency of the chequered network implementation on reception. As with all feed networks, an LNA may be added at each antenna element to improve system performance.

7.3.3 Multi-stage optimisation to minimise the range of coupling ratios and phase shifts

Two observations which arise from the approach outlined above are considered below.

First, the obvious choice when specifying the constraints on the coupling ratios and phase shifts is to specify the worst-case constraints that are tolerable for the chosen hardware implementation. However, it is often possible to find feasible solutions with even more stringent constraints, which is desirable as certain coupling ratios and phase shifts may be easier to implement. Unfortunately, it is difficult to predict the most stringent constraints that will produce a feasible solution for a given layout and $\mathbf{\Phi}$.

Second, some of the couplers may tend toward having $p_c = 1$ when optimising chequered networks without constraints on the coupling ratios, especially for couplers that are connected to terminated ports. This is a result of the algorithm attempting to minimise the power delivered to the terminated ports. Such couplers reduce to 90° phase shifters and will be referred to as shorted couplers. Shorted couplers are desirable as they simplify the implementation and reduce loss by removing terminated ports. Some couplers may also tend toward having $q_c = 1$, resulting in cross-shorted couplers, which comprise two 180° phase shifters that cross one another. Cross-shorted couplers undesirable as crossovers are difficult to implement [62].

These two observations can be used to devise a scheme for finding the most stringent constraints that are able to produce a feasible solution, while allowing the inclusion of shorted couplers to reduce power loss. Instead of formulating a multi-objective optimisation problem, the proposed approach is to sequentially find solutions using the approach in Section 7.3.2 with increasingly stringent constraints, while prioritising those constraints that are deemed most important.

The proposed sequential optimisation procedure is described in Algorithm 7.2. First, the constraints on the coupling ratios are removed and an initial optimisation is performed using Q trial points. The resulting solution is then evaluated and couplers with $p_c > p_{\text{short}}$ are shorted in the following steps by setting $p_c = 1$. For all non-shorted couplers, if the initial solution has more stringent coupling ratios than those corresponding to $p_{\min}$ and $p_{\max}$, the coupling-ratio constraints are altered to match the range of coupling ratios from the initial solution.

Next, the procedure outlined in Section 7.3.2 is followed repeatedly with coupling-ratio constraints that become more stringent with each MultiStart solver run, until no feasible solution can be found within

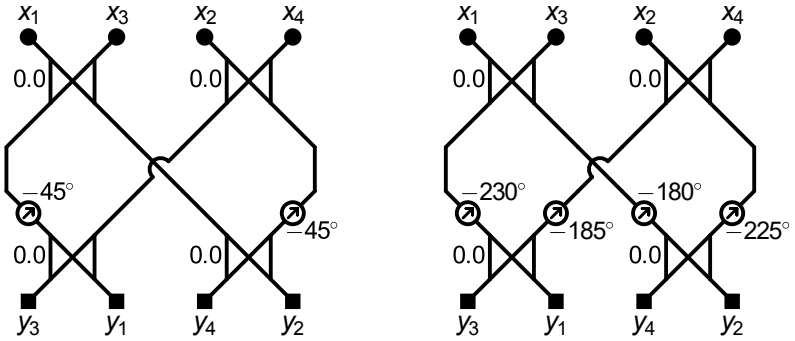


Figure 7.5. (a) A 4×4 Butler matrix implementation, and (b) the 4×4 DFT implementation obtained from the proposed algorithm with adjacent phase shifts combined. Coupling ratios in decibels appear to the left of the couplers. Taken from [24], ©2019 IEEE.

Q trials. At each run, the solver terminates upon finding the first feasible solution. Depending on the chosen priority, the minimum coupling ratio can be incremented first, the maximum coupling ratio can be decremented first, or the furthest of the two from 0 dB at each step can be changed.

A similar procedure is followed for the phase shifts, where the range $[\psi_{\min}, \psi_{\max}]$ is reduced until no feasible solution is found within Q trials. In the sections to follow, $\psi_{\max}$ is fixed and $\psi_{\min}$ is incremented since a less negative phase shift corresponds to a shorter transmission line, although a different scheme could also be employed depending on the type of phase shifter used.

The first trial point of each solver run is the last feasible solution from the previous run, and the remaining $Q - 1$ trial points are randomly distributed as described in Section 7.3.2. After obtaining the final constraints, a final optimisation is performed from Q trial points.

7.4 RESULTS FOR SOME TEST CASES

The results for a number of test cases are presented below, after a description of the system parameters.

The chequered-network optimisation algorithm was implemented in MATLAB R2018b and run on a computer with two 6-core Intel Xeon E5-2630 processors and 32 GB of memory. Couplers from the initial solution with coupling ratios above 20 dB were shorted, and coupling ratios and phase shifts were varied in 0.1 dB and 5° steps, respectively.

7.4.1 4×4 Butler matrix

A Butler matrix is an example of a completely overlapped feed network and is a hardware-based implementation of a fast Fourier transform (FFT) [77]. In this section, a chequered network is designed to implement the DFT and is shown to approach the optimised Butler matrix (FFT) implementation.

Figure 7.5(a) shows a standard implementation of a 4×4 Butler matrix [61]. In practice, an additional crossover may be required to correctly order ports x_2 and x_3.

The desired response for a 4×4 Butler matrix is given by [2]

$$\mathbf{\Phi} = \begin{bmatrix} 1\angle 157.5^\circ & 1\angle -67.5^\circ & 1\angle 67.5^\circ & 1\angle -157.5^\circ \\ 1\angle -67.5^\circ & 1\angle -22.5^\circ & 1\angle 22.5^\circ & 1\angle 67.5^\circ \\ 1\angle 67.5^\circ & 1\angle 22.5^\circ & 1\angle -22.5^\circ & 1\angle -67.5^\circ \\ 1\angle -157.5^\circ & 1\angle 67.5^\circ & 1\angle -67.5^\circ & 1\angle 157.5^\circ \end{bmatrix}. \tag{7.18}$$

The chequered layout was chosen to mirror that of a Butler matrix in anticipation of the network reducing to the standard implementation. The procedure in Algorithm 7.2 was followed with $\psi \in [-180^\circ, 0^\circ]$, $Q = 20$, and an error tolerance of 10^{-6} on the feed network response. In the same way that couplers with large coupling ratios are shorted, couplers with coupling ratios below −20 dB were constrained to have $p_c = 0$ to allow crossovers in the design as they are present in the standard implementation.

The resulting chequered network is shown in Figure 7.5(b), where adjacent phase shifts, including the phase shifts in the shorted and cross-shorted couplers, have been combined. All couplers converged to equal-split couplers.

The design can be further simplified by introducing a phase shift of +185° to ports y_1 and y_3 by changing the phases of the correction weights at these ports, σ_1 and σ_3. Applying the splitting rule [60], the newly introduced phase shifts can be split and shifted up to the two branches containing the −230° and −185° phase shifts, which then become −230° + 185° = −45° and −185° + 185° = 0°, respectively. Applying a 180° phase shift to ports y_2 and y_4 has a similar effect, and the −180° and −225° phase shifts become 0° and −45°, respectively. The resulting simplified design is the same as the standard implementation in Figure 7.5(a), thereby validating the proposed algorithm.

7.4.2 Three independent subarrays spanning eight elements

In order to demonstrate the versatility of the proposed approach, three non-scanning conventional 8-element arrays were designed independently and combined to share the same aperture by means of a 3×8 chequered network. The spacing between elements was half a wavelength.

The specifications of the three arrays were chosen arbitrarily. The first two subarrays were designed using the shaped-beam synthesis algorithm proposed by Orchard *et al.* [78], and five and two pattern roots were placed in the sidelobe and main-beam regions, respectively. The first subarray has a $\mathrm{cosec}^2(\theta)\cos(\theta)$ main-beam shape with its peak at 10° [78] and −20-dB sidelobes. The required beam shape was realised to within ±0.1 dB across $16^\circ < \theta < 48^\circ$. The second array has a flat-topped [78] main beam centred around −15°, with the sidelobes to the left and right of the main beam being −15 dB and −20 dB, respectively. The flat-topped beam was realised to within ±0.01 dB across $-27^\circ < \theta < -4^\circ$. The third subarray is a −30-dB Chebyshev array with the main beam at 30°. The resulting $\mathbf{\Phi}$ is given by

$$|\mathbf{\Phi}| = \begin{bmatrix} 0.24 & 0.24 & 0.43 & 0.48 & 0.68 & 1.00 & 0.96 & 0.63 \\ 0.15 & 0.13 & 0.30 & 0.06 & 0.51 & 0.99 & 1.00 & 0.56 \\ 0.26 & 0.52 & 0.81 & 1.00 & 1.00 & 0.81 & 0.52 & 0.26 \end{bmatrix} \tag{7.19}$$

and

$$\angle\mathbf{\Phi} = \begin{bmatrix} 0.0^\circ & 32.1^\circ & -2.8^\circ & -25.8^\circ & -36.6^\circ & -75.0^\circ & -129.0^\circ & 171.5^\circ \\ 0.0^\circ & -139.9^\circ & -81.2^\circ & -27.0^\circ & -168.1^\circ & -125.6^\circ & -80.2^\circ & -30.5^\circ \\ 0.0^\circ & -90.0^\circ & 180.0^\circ & 90.0^\circ & 0.0^\circ & -90.0^\circ & 180.0^\circ & 90.0^\circ \end{bmatrix}, \tag{7.20}$$

for which the subarray radiation patterns are shown in Figure 7.6.

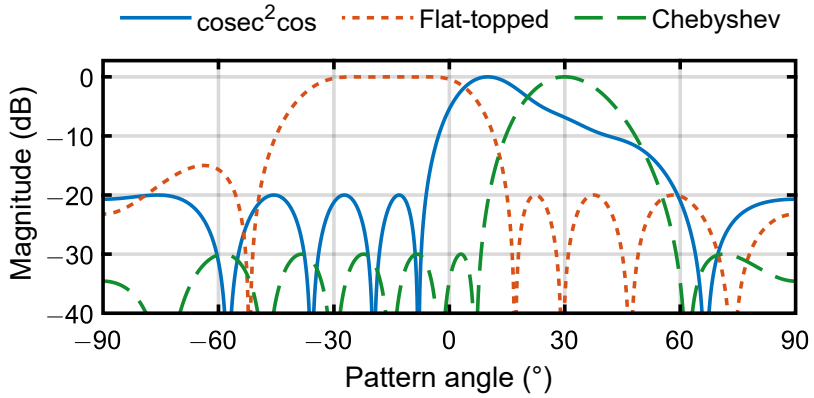


Figure 7.6. Normalised radiation patterns of three independent subarrays spanning eight elements. Adapted from [24], ©2019 IEEE.

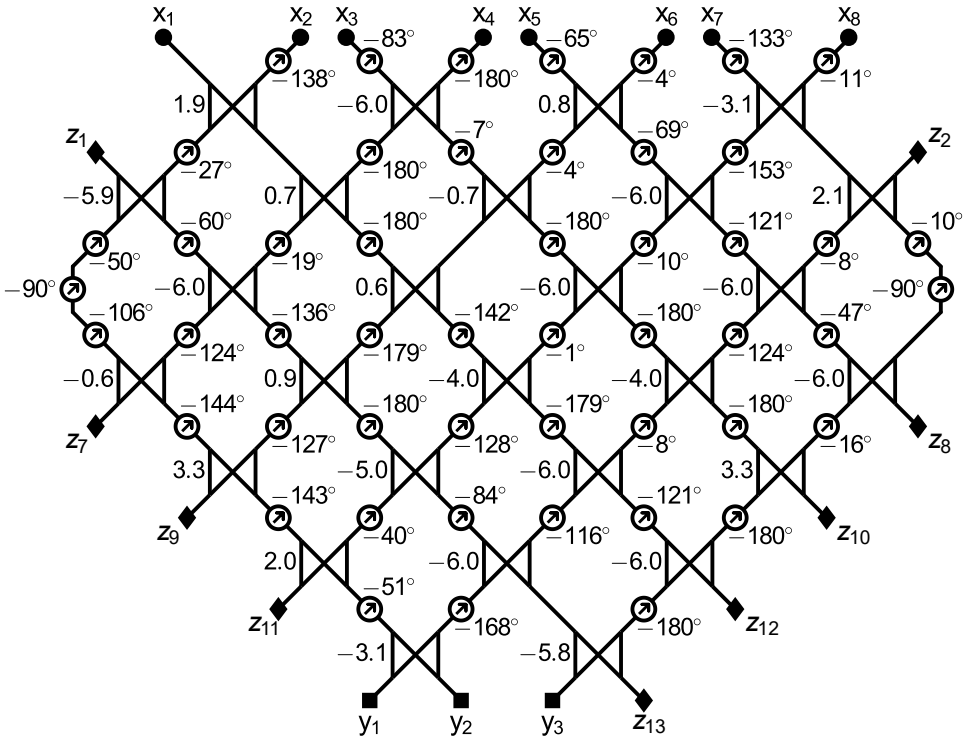


Figure 7.7. The obtained design for the 3×8 independent-subarray problem. Coupling ratios in decibels appear to the left of the couplers. Taken from [24], ©2019 IEEE.

A chequered network was designed to simultaneously implement these three patterns as a fully-overlapped array by using the illuminations of the three subarrays to form the rows of $\mathbf{\Phi}$. Algorithms 7.1 and 7.2 were used with with $\psi \in [-180°, 0°]$, $Q = 20$, and an error tolerance of 10^{-3} on the feed network response. Equal priority was given to increasing $p_{\min}$ and decreasing $p_{\max}$ in (7.14) to minimise the range of coupling ratios as outlined in Section 7.3.3.

Figure 7.7 shows the resulting chequered-network design. The final coupling ratios are between −6 dB and 5.7 dB, and the phase shifts range from −180° to 0°. The antenna element with the lowest loss is at port x_6, from which −3.8 dB of power is delivered to terminated ports z_7 to z_{12}, which gives an indication of the efficiency of the chequered network implementation on reception. On transmission,

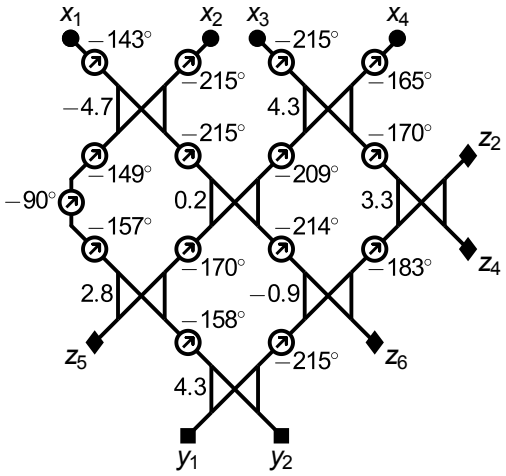


Figure 7.8. The obtained design for the 2×4 compressive-feed-network problem. Coupling ratios in decibels appear to the left of the couplers. Taken from [24], ©2019 IEEE.

the total power delivered to terminated ports z_1 and z_2 relative to the input power at the subarray ports y_1, y_2, and y_3 is −4.6 dB, −5.6 dB, and −6.6 dB, respectively. Adding appropriate constraints changes the above powers to −5.0 dB, −6.0 dB, and −5.5 dB, respectively, when constrained to be less than −5 dB.

7.4.3 2×4 compressive feed network

In this section, measured results for a 2×4 narrowband compressive feed network implemented in microstrip for a ULA are presented. These are the first published results for a successfully manufactured compressive feed network without hardware constraints on $\mathbf{\Phi}$. The steering range was limited to $|\theta_s| < 10°$ and the start of the SLL region was $\sin(\theta_s) = 0.35$. The element spacing was limited to 0.64 wavelengths by manufacturing considerations.

This feed network represents the smallest number of subarrays that constitutes a beamforming array as a single subarray would not allow dynamic control of the array pattern because variations to the single excitation would not affect the array pattern. The limited number of controls thus makes it essential to exploit all the available degrees of freedom if acceptable array performance is to be achieved. Using a compressive feed network configuration is therefore desirable, as it places no constraints on $\mathbf{\Phi}$.

Ten independent runs of the compressive array design algorithm (Chapter 6) all obtained an SLL of −12.15 dB, so all 10 designs were candidates for implementation. Chequered networks were designed using Algorithms 7.1 and 7.2 with $\psi \in [-260°, -140°]$, $Q = 20$, and an error tolerance of 10^{-3} on the feed network response. The maximum phase shift was set to −140° to account for the minimum transmission-line length required to connect the couplers. Decreasing $p_{\max}$ was prioritised to avoid the excessively thin microstrip lines which result from large coupling ratios.

Out of the 10 designs, the network that achieved the most stringent constraints while still satisfying the specifications was chosen, with the resulting response of

$$\mathbf{\Phi} = \begin{bmatrix} 0.67\underline{/261.6°} & 0.35\underline{/180.2°} & 0.26\underline{/126.4°} & 0.52\underline{/84.9°} \\ 0.33\underline{/218.8°} & 0.51\underline{/194.6°} & 0.65\underline{/213.1°} & 0.52\underline{/266.8°} \end{bmatrix}$$

being implemented by the network shown in Figure 7.8. The final coupling ratios are between −4.7 dB and 4.3 dB, and the phase shifts range from −215° to −143°. The phase shifts were not simplified as was

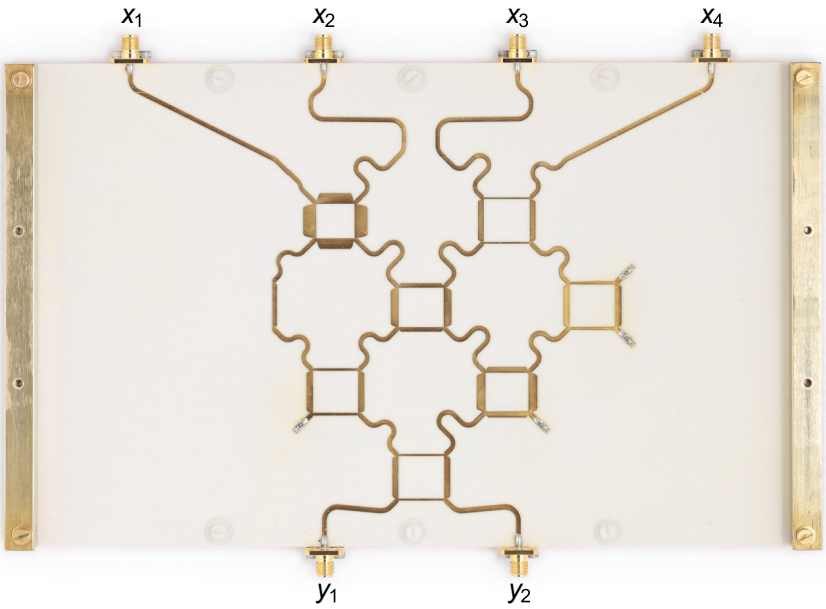


Figure 7.9. Photograph of the manufactured 2×4 compressive feed network. Taken from [24], ©2019 IEEE.

done in Section 7.4.1, showing that Algorithms 7.1 and 7.2 produce designs which can be implemented even without additional tuning. The antenna element with the lowest loss (port x_1) delivers −3.6 dB of power to ports z_4 to z_6. On transmission, the power delivered to terminated port z_2 relative to the input power at the subarray ports y_1 and y_2 is −13.2 dB and −8.9 dB, respectively.

The chequered network was implemented in microstrip using single-section branchline couplers [60], with lengths of transmission line being used as phase shifters. The substrate used was Rogers R4003C with a process dielectric constant of 3.38 and a height of 0.81 mm. The manufactured circuit is shown in Figure 7.9. The implementation of chequered networks in microstrip will be described in more detail in Chapter 8.

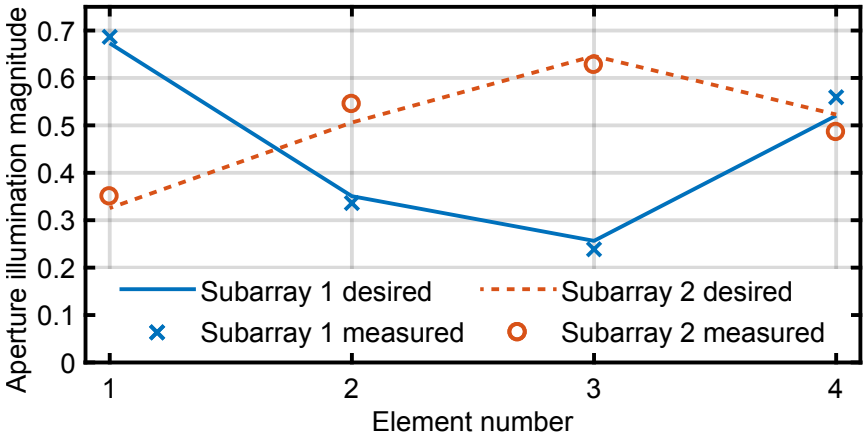
The desired and measured aperture illumination magnitudes and phases at 3.15 GHz are shown in Figure 7.10. The subarray corrections σ_m for the manufactured network were re-calculated to minimise the maximum deviations from the desired aperture illuminations in decibels and degrees. The corrected measured illuminations are within 0.6 dB and 5.2° of the desired response.

The measured frequency-dependent results of the chequered network are summarised in Table 7.1. Figure 7.11 shows the worst measured return loss and isolation across all relevant ports. Specifying the required voltage standing wave ratios (VSWRs) as 1.5 and 2 (return losses of 14 dB and 9.5 dB, respectively) give bandwidths of 4.5% and 14.6%, and worst-case isolations of 16.9 dB and 11.2 dB, respectively. Considering a bandwidth of 10% gives a VSWR of 1.65 (a return loss of 12.2 dB) and a worst-case isolation of 12.6 dB.

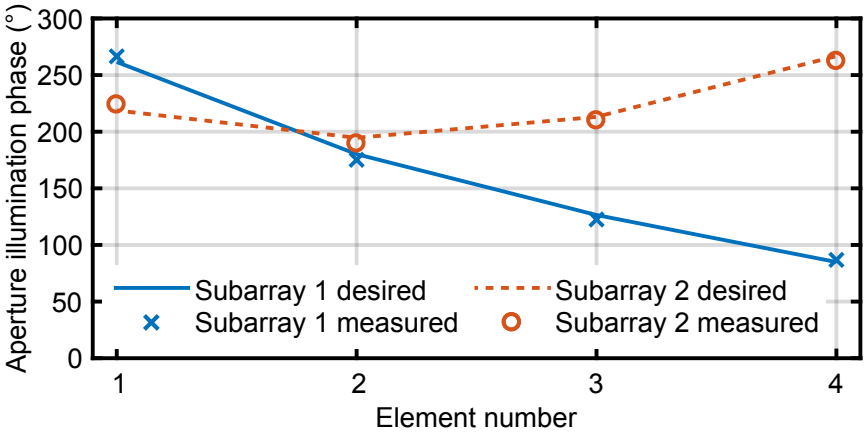
Figure 7.12 shows the accuracy with which the manufactured network realises the desired response as a function of frequency. Errors are found by calculating $\sigma_m S_{y_m,x_n} / \phi_{m,n}$ $\forall m, n$. The maximum error is calculated as the range of errors divided by two, which ensures that only the relative amplitude and phase distributions are evaluated because arbitrary amplitude and phase shifts can be introducted as outlined in Section 7.2.3. The standard deviation of the errors is also presented to quantify the distribution of the errors relative to their mean. The accuracy across various frequency ranges is given in Table 7.1. The fact that the system was designed for a single frequency using narrowband components, is seen

Table 7.1. Measured Worst-Case Frequency Results for the 2×4 Compressive Feed Network. Taken from [24], ©2019 IEEE.

	Return loss (dB)	Isolation (dB)	f_l (GHz)	f_u (GHz)	Bandwidth (%)	Response accuracy Max.	Response accuracy St. dev.	Transmit/receive loss (dB) Subarrays	Transmit/receive loss (dB) Element x_1
3.15 GHz	18.1	21.2	3.15	3.15	–	0.6 dB, 5.2°	0.6 dB, 4.5°	1.9	3.7
VSWR $\leq$ 1.5	14.0	16.9	3.07	3.21	4.5	1.4 dB, 10.3°	0.9 dB, 6.3°	2.0	4.0
VSWR $\leq$ 1.65	12.2	12.6	3.00	3.32	10.0	1.9 dB, 17.2°	1.2 dB, 10.6°	2.6	4.5
VSWR $\leq$ 2	9.5	11.2	2.94	3.40	14.6	3.5 dB, 22.9°	2.1 dB, 14.5°	3.2	5.3



(a)



(b)

Figure 7.10. Aperture illumination (a) magnitudes and (b) phases for the 2×4 compressive feed network. Taken from [24], ©2019 IEEE.

by the fact that the results are excellent at 3.15 GHz, but deteriorate as the bandwidth is increased. Despite this deterioration, it is believed that the performance at 10% and even 14.6% bandwidth will be acceptable in many applications.

Figure 7.13 shows the measured transmit loss from each subarray port to the element ports, and the measured receive loss from the element port with the least loss (x_1) to the subarray ports as a function of frequency. These measurements include the effects of return loss, isolation, power delivered to terminations, and component losses (e.g. due to dielectric loss and finite conductivity). Key results are again summarised in Table 7.1. These results show that the ability to implement an arbitrary feed network comes at the expense of losses, which include component losses, which are dependent on the size of the network, and the power delivered to terminations, which depends on the chequered network design.

The results above show that a practical chequered network can be designed and realised despite the many variations inherent in manufacturing. The achieved return loss, isolation and feed network accuracy were reasonable despite manufacturing tolerances, and no manual tuning of the circuit was required. The measured results were achieved without de-embedding the connectors [79], and therefore include the effect of mismatching at the interfaces between the connectors and the microstrip circuit. This

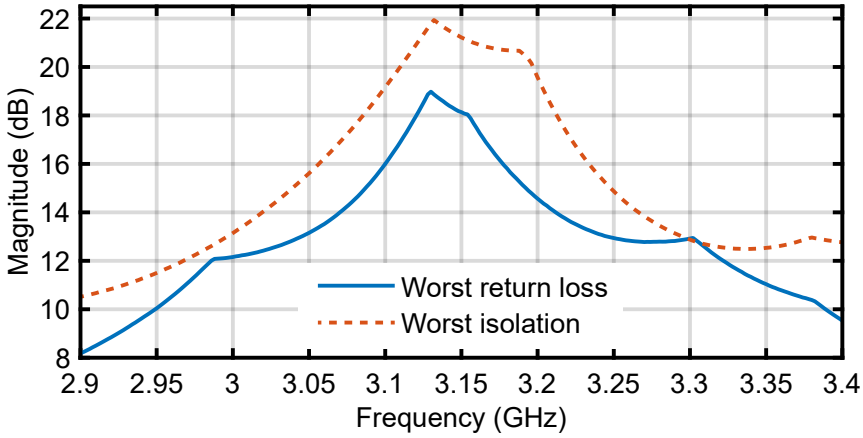


Figure 7.11. Measured worst return loss and isolation across all relevant ports for the 2×4 compressive feed network. Taken from [24], ©2019 IEEE.

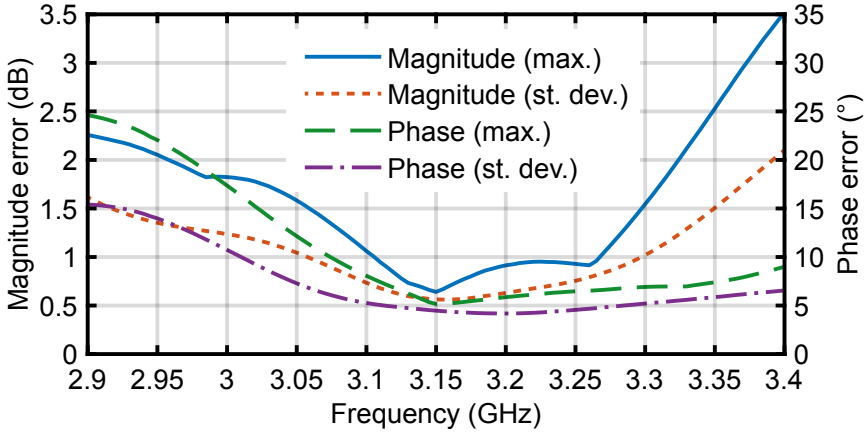


Figure 7.12. Measured accuracy of the 2×4 compressive feed network as a function of frequency. Taken from [24], ©2019 IEEE.

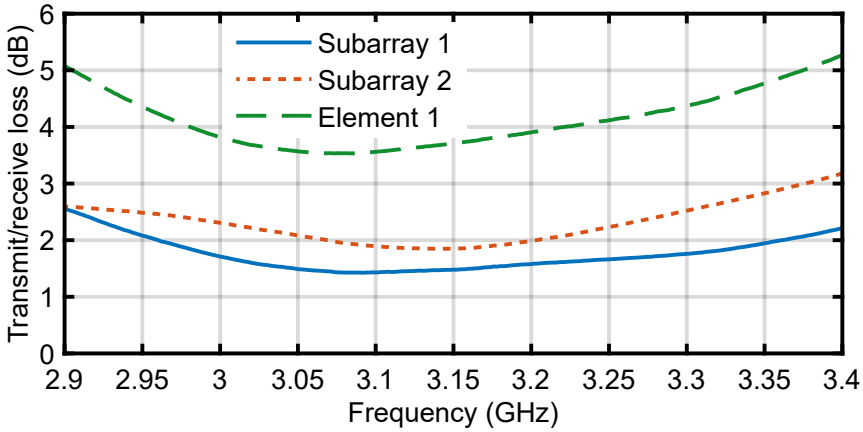


Figure 7.13. Transmit loss from subarray ports and receive loss from element port x_1 of the 2×4 compressive network as a function of frequency. Taken from [24], ©2019 IEEE.

shows that chequered networks perform well even when connected to external circuitry via non-ideal interfaces. A chequered network is also able to achieve reasonable results across a range of frequencies despite both using narrowband components and being designed at a single frequency. The presence of terminations in a chequered network can increase losses, but such losses will depend on the network structure and response.

7.5 CHAPTER SUMMARY

A network of couplers and phase shifters has been proposed for implementing arbitrary complex-valued feed networks. The responses of such chequered networks have been derived using a process based on a series of matrix multiplications. An optimisation approach to determine the coupling ratios and phase shifts required to implement an arbitrary feed network while satisfying constraints necessary to allow the network to be physically implemented has been outlined.

The proposed chequered network and algorithm were validated by demonstrating that a synthesised 4×4 DFT chequered network approaches the manually-designed Butler matrix (FFT) implementation. Three independently designed subarrays were overlapped to share an 8-element aperture (a 3×8 array), illustrating the versatility of the proposed approach. A 2×4 compressive feed network for a ULA was implemented as a microstrip chequered network to demonstrate that complicated unconventional network responses can be realised. The measured aperture illuminations of the microstrip chequered network were within 0.6 dB and 5.2° of the desired aperture illuminations at 3.15 GHz, and within 1.4 dB and 10.3° across the impedance bandwidth of 4.5%. The ability to implement arbitrary feed network responses comes at the expense of losses on transmission and reception, though these losses depend on the network response and structure.

CHAPTER 8 DESIGN AND IMPLEMENTATION OF A PROTOTYPE COMPRESSIVE ARRAY

8.1 CHAPTER OVERVIEW

Thus far, an algorithm for designing compressive array feed network responses (Chapter 6), and an algorithm for implementing a desired feed network response as a chequered network (Chapter 7) has been proposed. A chequered network has successfully been implemented in microstrip in Chapter 7, although the details of such an implementation have not yet been given. All the required components are in place to develop and implement a prototype compressive array. In this chapter, a set of specifications are chosen and a compressive array with two subarrays and four integrated antenna elements is designed, manufactured using a single sheet of substrate, and tested. The manufactured array achieves an SLL within 1.1 dB of the theoretical SLL at 3 GHz, and has a 3-dB SLL bandwidth of 7.9%. By describing the practical implementation of a compressive array with integrated antenna elements, this chapter shows how the various components of a compressive array are designed and integrated to form a complete working system. This chapter also highlights some of the practical issues associated with designing and manufacturing a compressive array.

The specifications for the prototype compressive array are given in Section 8.2. Section 8.3 describes the design of the antenna elements and presents the simulated embedded element patterns on which the feed network design is based. Section 8.4 describes the design of the compressive feed network response and its implementation as a theoretical chequered network. The resulting chequered network is implemented in microstrip as described in Section 8.5, which also serves as a general framework for implementing chequered networks in microstrip. Simulated and measured results are presented in Section 8.6 and compared to the theoretical results. Finally, the chapter is summarised in Section 8.7.

8.2 DESIGN SPECIFICATIONS

The chosen application for the prototype compressive array is a beamforming array with a limited steering range. Four antenna elements and two subarrays were implemented since this allowed the entire compressive array to be implemented on a single sheet of 9” by 12” substrate. Two subarrays represents the smallest number of subarrays that constitutes a beamforming array. Such a small reduced-control array requires all the available degrees of freedom to be exploited to obtain reasonable performance, making a compressive array a good choice over other reduced-control techniques. For example, it would be impossible to design a thinned array that selects two elements out of four without introducing ambiguities in the array factor. The only reasonable option would be to select two adjacent elements, which is equivalent to a conventional two-element array and thus not a reduced-control array.

The chosen specifications for the prototype compressive array are listed below.

1. The compressive array should operate at 3 GHz. This frequency allows for a reasonably compact distributed circuit implementation while still maintaining dimensions that are large enough to be easily manufactured using standard techniques.
2. The steering range should be $|\theta_s| < 10°$, a typical range for limited-steering applications [39], rounded to $|\sin(\theta_s)| < 0.175$.
3. The start of the SLL region should be at $\sin(\theta) = 0.35$. This results in an SLL similar to that of a uniform-excitation four-element ULA with isotropic elements.
4. The VSWRs of the subarrays should be better than 2 and the isolation between the subarray ports better than 10 dB at 3 GHz.
5. Due to cost limitations, the feed network and antenna array should fit on a single sheet of double-sided substrate of dimensions 272 mm by 196 mm (the usable space on a 9" by 12" sheet).

8.3 DIPOLE ARRAY DESIGN

The design of the antenna array was performed in simulation using Computer Simulation Technology (CST) Microwave Studio 2018. The chosen type of antenna element was a printed dipole with an integrated balun since it is easy to implement on double-sided subtrate [80, 81].

A single printed dipole was first designed in isolation. The dipole arms were optimised to resonate at 3 GHz, after which the balun parameters were optimised for impedance matching at 3 GHz. Next, four of these dipoles were placed in an array along with the sheet of substrate on which the feed network would be implemented and the mounting brackets for holding the substrate and subminiature version A (SMA) connectors. The embedded dipole elements were then optimised to account for mutual coupling. The dipole arm length for each element was optimised to move the resonant frequency back to 3 GHz, after which the feed point of each balun was optimised for impedance matching at 3 GHz.

The chosen element spacing was 0.64 wavelengths. This spacing ensures that the start of the SLL region of an ambiguity does not enter the visible region up to a steering angle of $\sin(\theta_s) = 0.2125$ or $\theta_s = 12.3°$, slightly wider than the steering range of interest. An element spacing of 0.64 wavelengths allowed the antenna array to fit on the substrate with some space for the ground-plane reflector to extend beyond the outer elements.

Figure 8.1 shows the simulated embedded element gain patterns, and Figure 8.2 shows the worst-case return loss and isolation across all elements. The elements have peak gains ranging from 4.3 to 4.8 dBi. The VSWRs are equal to 2 (a return loss of 9.5 dB) or better from 2.82 GHz to 3.36 GHz, an impedance bandwidth of 17%. The isolation between all ports is better than 17.2 dB across this range. The return losses at 3 GHz are 46.6 dB or better and the isolations at 3 GHz 19.5 dB or better.

The simulated embedded element patterns were used to design the feed network response as described in Section 8.4. This shows that the compressive array design algorithm (Chapter 6) is able to utilise embedded element patterns which are not identical.

The alternative to using simulated element patterns is to first manufacture the antenna array, measure its response, and then use the measured response to design the feed network. The disadvantage of this approach is that interfaces (e.g. SMA connectors) would be required to connect the array to the measurement equipment, and additional interfaces to connect to the subsequently manufactured feed network (e.g. barrel and SMA connectors). Cascading such interfaces can lead to potentially significant and unpredictable discontinuities, minimising the benefit of using measured element patterns.

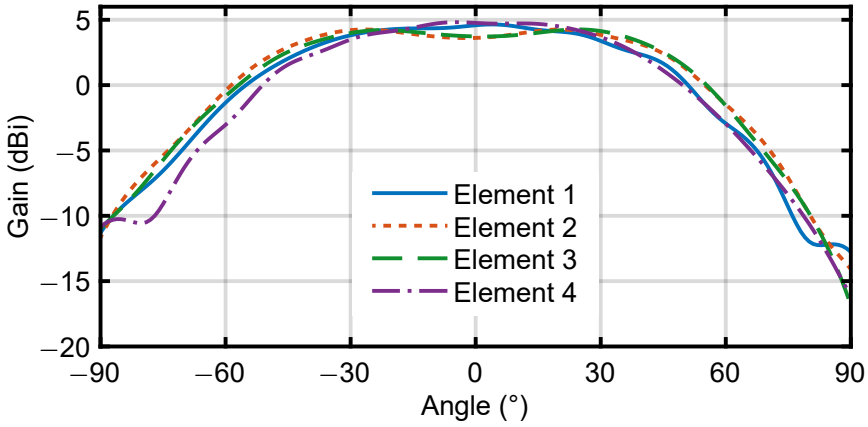


Figure 8.1. Simulated dipole element gain patterns.

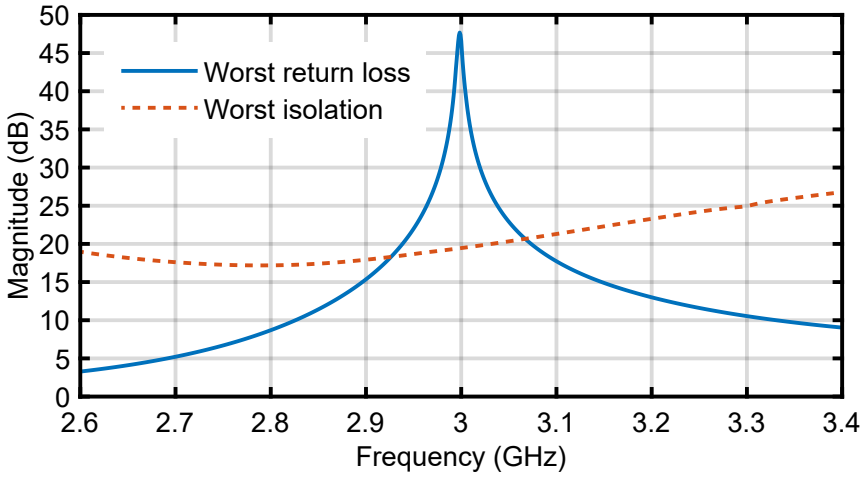


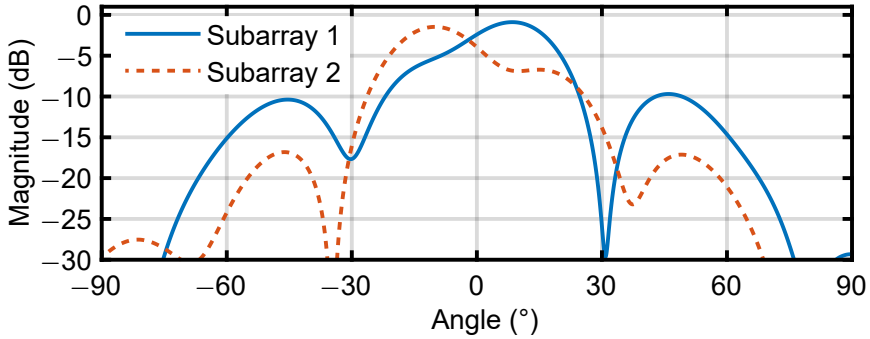
Figure 8.2. Worst-case return loss and isolation across all dipole elements.

Using simulated element patterns allows the entire compressive array to be manufactured as a single component, but assumes good agreement between simulation and measurement.

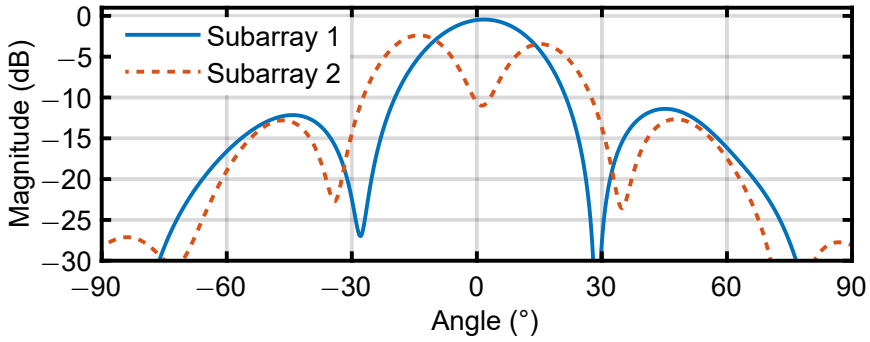
8.4 DESIGN OF THE FEED NETWORK RESPONSE AND CHEQUERED NETWORK

The feed network response was designed using the compressive array design algorithm from Chapter 6 for the specifications listed in Section 8.2 with the simulated embedded element patterns in Figure 8.1. The element patterns were only specified and the SLL only calculated across $\theta \in [-90°, 90°]$ since the patterns outside $\theta \in [-90°, 90°]$ were assumed to be sufficiently suppressed due to the ground-plane reflector behind the printed dipole elements.

Since the SLL of a compressive array typically deteriorates rapidly beyond the steering range (see e.g. Figure 6.7), a slightly wider steering range of $|\sin(\theta_s)| < 0.2$ or $|\theta_s| < 11.5°$ was passed to the algorithm. The sum of the subarray pattern powers was constrained to be within ±0.1 dB across the steering range (constraints (6.9) and (6.10)) as the SLL deteriorated significantly when the constraint was reduced below this value. The algorithm was run 20 times and all 20 designs achieved an SLL of −11.3 dB which is equal to the SLL of a uniform-excitation four-element ULA with isotropic



(a)



(b)

Figure 8.3. Resulting subarray patterns from the compressive array design algorithm for (a) design 10 and (b) design 15.

elements. Therefore, all 20 designs were considered as candidates for implementation in a chequered feed network.

The feed network responses were implemented as chequered networks using the chequered network design algorithm from Chapter 7. The range of phase shifts was set to $\psi \in [-260°, -140°]$. The upper bound $\psi_{\max}$ corresponds to the approximate minimum length of transmission line required to connect two couplers. The lower bound $\psi_{\min}$ is determined by the approximate longest transmission line that will fit on the chosen chequered tile size, which will be described in Section 8.5. The chequered network was required to implement the desired response with a magnitude and phase tolerance of 0.1 dB and 1°, roughly an order of magnitude less than the expected accuracy of the hardware implementation (Chapter 7). The minimum and maximum coupling ratio constraints were varied with equal priority (Section 7.3.3). Couplers from the initial solution with coupling ratios above 20 dB were shorted, and $Q = 20$ trial points were used.

All 20 compressive feed networks were successfully implemented as chequered networks. The smallest coupling ratio range (the ratio between the largest and smallest coupling ratio for a given design) was 3.2 dB (design 15), the median coupling ratio range 9.5 dB, and the largest coupling ratio range 13.1 dB. The smallest phase shift range for any one design was 54° (design 15), the median phase shift range 87°, and the largest phase shift range 120°.

Figure 8.3 shows the subarray patterns for the two designs with the smallest coupling ratio ranges, namely design 10 (7.2 dB) and design 15 (3.2 dB). Design 10 has two beams pointing in opposite directions, while design 15 has one subarray with a single main lobe near broadside, and one subarray with two lobes to the left and right of broadside with low gain near broadside. These two designs

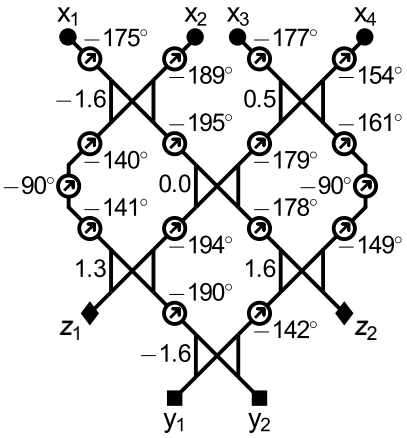


Figure 8.4. Chequered network obtained for design 15. Coupling ratios in decibels appear to the left of the couplers.

show striking similarities to two common types of monopulse array patterns, namely squinted-beam monopulse patterns and sum-and-difference monopulse patterns [82]. The algorithm is thus shown to approach known manually designed systems for limited-scan DoA estimation. The compressive array design algorithm can therefore also be expected to obtain good results for larger systems and systems with unconventional requirements for which manual designs do not yet exist.

Design 15 was chosen since it had the smallest coupling ratio and phase shift range. The resulting chequered network is shown in Figure 8.4. The coupling ratios range from -1.6 dB to 1.6 dB and the phase shifts from $-195°$ to $-140°$. This design has the added benefit of being lossless on transmit.

8.5 MICROSTRIP IMPLEMENTATION OF THE CHEQUERED NETWORK

Microstrip was chosen to implement the chequered network since it can easily be manufactured on double-sided substrate. The chosen substrate was Rogers R4003C with a process dielectric constant of 3.38 and a height of 0.81 mm. In simulation, the recommended design value for the dielectric constant of 3.55 was used.

The proposed approach is to draw a parametrised tile of a chosen size in a full-wave electromagnetic solver such as CST Microwave Studio and then optimise the parameters for each tile in the chequered layout. The proposed microstrip circuit with associated parameters for a single tile is shown in Figure 8.5, where the tile is square with sides 40 mm long. The circuitry consists of a branchline coupler and two meanders of transmission lines with variable heights. The branchline coupler moves up or down as the length of the cross arms vary, while the positions of the microstrip lines at the ports remain constant. This enables the tiles to be connected simply by placing them next to one another.

The parameters of the tile are the lengths and widths of the through and cross arms and the heights of the meanders. Chequered networks may require large coupling ratios, leading to very wide microstrip lines with potentially significant discontinuities at the coupler junctions. Discontinuities are therefore compensated for using constant-impedance tapers [83].

Sufficiently wide curved bends offer less significant discontinuities than chamfered corners [83]. Defining the curved microstrip lines by blending edges in CST Microwave Studio allows for meanders with small heights, almost down to straight lines, whereas the use of chamfered corners would require a minimum height for the meandered lines. These considerations led to the choice of curved bends for implementing the meandered lines.

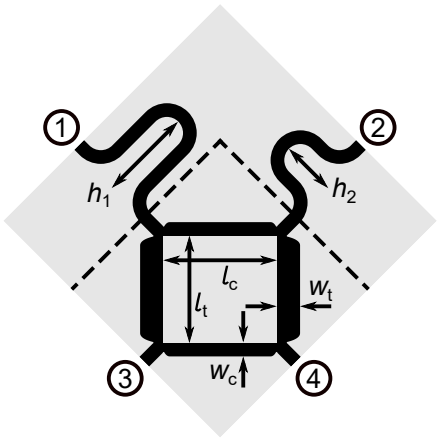


Figure 8.5. Microstrip implementation of a single chequered network tile with optimisation parameters indicated. Each side is 40 mm long. Tile port numbers are encircled. Dashed lines show where the tile is cropped for the optimisation of the coupler.

The first step in optimising a tile is to crop the tile as illustrated by the dashed lines in Figure 8.5, leading to a coupler circuit with horizontal and vertical symmetry. The lengths and widths of the coupler arms are optimised for the following goals:

1. The coupling ratio, given by $|S_{3,1}|/|S_{4,1}|$, must equal the desired coupling ratio. The coupling ratio is used instead of the desired power to the through and coupled ports (p^2 and q^2, respectively) since the coupling ratio can be realised exactly. The values of p^2 and q^2 are specified assuming that $p^2 + q^2 = 1$ which will not hold true due to losses and mismatching, so that the specified p^2 and q^2 cannot be realised exactly.
2. $S_{1,1}$ is minimised at 3 GHz to ensure matched transmission-line junctions.
3. The phase difference $\angle S_{4,1} - \angle S_{3,1}$ must equal $-90°$ as assumed in the analytical tile model. This enables accurate phases to be realised when the meandered lines are included.
4. $S_{2,1}$ is minimised at 3 GHz to maintain isolation between ports 1 and 2, and between ports 3 and 4.

After optimising the coupler in isolation, the meandered lines are included and their heights are optimised so that $\angle S_{3,1} = \psi_1 - 90°$ and $\angle S_{4,2} = \psi_2 - 90°$.

Where tiles are shorted, only the through arms and meandered lines are kept. The tapers at the junctions are kept, since constant-impedance tapers are also useful to compensate for bend discontinuities [83]. This simplifies the design procedure as the same tile template can be used for shorted tiles with only minor modifications.

Where port 1 or 2 is terminated, the corresponding meandered line is removed and replaced with a matched termination. Matched terminations were implemented as 50 Ω surface-mount resistors that connect to the ground plane by means of vias.

Figure 8.6 shows the complete microstrip circuit with all the tiles connected. The transmission lines that connect the upper tiles to antenna elements 2 and 3 were meandered to have the same phase length as the transmission lines connecting to elements 1 and 4. The thickest microstrip line is the through arm of coupler 1 which is 3.6 mm wide, corresponding to a characteristic impedance of about 31 Ω. The thinnest line is the cross arm of coupler 7 which is 1.3 mm wide, corresponding to a characteristic impedance of about 61 Ω.

8.6 RESULTS

Figure 8.7 shows a photograph of the manufactured and assembled compressive array. The substrate and SMA connectors were held together using 10-mm wide square brass rods. In the following sections, results will be presented for the scattering parameters (Section 8.6.1), subarray patterns (Section 8.6.2), broadband SLL (Section 8.6.3), and 3-GHz steered patterns (Section 8.6.4) of the compressive array.

8.6.1 Scattering parameter results

Figure 8.8 shows the simulated and measured scattering parameters of the assembled compressive array. The return losses and isolation of the simulated array are better than 17.1 dB at 3 GHz. The simulated array has VSWRs of 2 (a return loss of 9.5 dB) or better from 2.77 GHz to beyond 3.4 GHz, but the isolation is better than 10 dB only up to 3.35 GHz. This gives a combined impedance and isolation bandwidth of 19%. The return losses and isolation of the manufactured array are better than 17.7 dB at 3 GHz. The manufactured array has VSWRs of 2 or better from 2.84 GHz to 3.28 GHz, and the isolation across this range is better than 12.3 dB. This gives an impedance bandwidth of 14%. Considering the resonant frequencies of $S_{1,2}$ and $S_{2,2}$, there appears to be an upward frequency shift of about 70 MHz from the simulated to the manufactured results.

8.6.2 Subarray pattern results

Figure 8.9 shows the desired, simulated and measured subarray patterns at 3 GHz. The desired subarray patterns are those resulting from the compressive array design algorithm. Between $\theta \in [-20°, 20°]$, the simulated patterns equal the desired patterns within 0.9 dB, and the manufactured patterns equal the desired patterns within 1.5 dB. The simulated subarray sidelobes have peak values within 0.9 dB of the desired subarray sidelobe levels, and the measured patterns have subarray sidelobe levels within 2.6 dB of the desired levels. These results show that there is good agreement with the desired patterns at both the simulation and manufacturing stages of the design process, especially in the main-lobe regions where the gains are the highest. The main differences between the measured and desired patterns are the depth of the nulls and shape of the sidelobes in subarray 1, and the height of the right sidelobe in subarray 2.

Simulated subarrays 1 and 2 had peak gains of 6.9 dB and 6.1 dB, respectively. Both manufactured subarrays had peak gains of 7.3 dBi. The patterns in the region outside $\theta \in [-90°, 90°]$ were suppressed by at least 12.1 dB and 14.5 dB relative to the peak gains for the simulated and manufactured arrays, respectively. This validates the assumption that the patterns outside $\theta \in [-90°, 90°]$ are sufficiently suppressed to have a negligible effect on the SLL.

8.6.3 Broadband SLL results

The SLL of the simulated and manufactured compressive arrays as a function of frequency is shown in Figure 8.10. The correction weights (Section 7.3.2) were calculated to minimise the SLL at 3 GHz. The SLL of the simulated array reaches a minimum of −11.2 dB at 3 GHz, or within 0.1 dB of the theoretical SLL. The simulated SLL is within 3 dB of its minimum value from 2.754 GHz to 3.288 GHz for a 3-dB SLL bandwidth of 17.7%. The SLL of the manufactured array is −10.2 dB at 3 GHz, or within 1.1 dB of the theoretical SLL. A minimum SLL of −11.2 dB occurs at 3.05 GHz. The SLL for the manufactured array is within 3 dB of this value from 2.899 GHz to 3.137 GHz for a 3-dB SLL bandwidth of 7.9%.

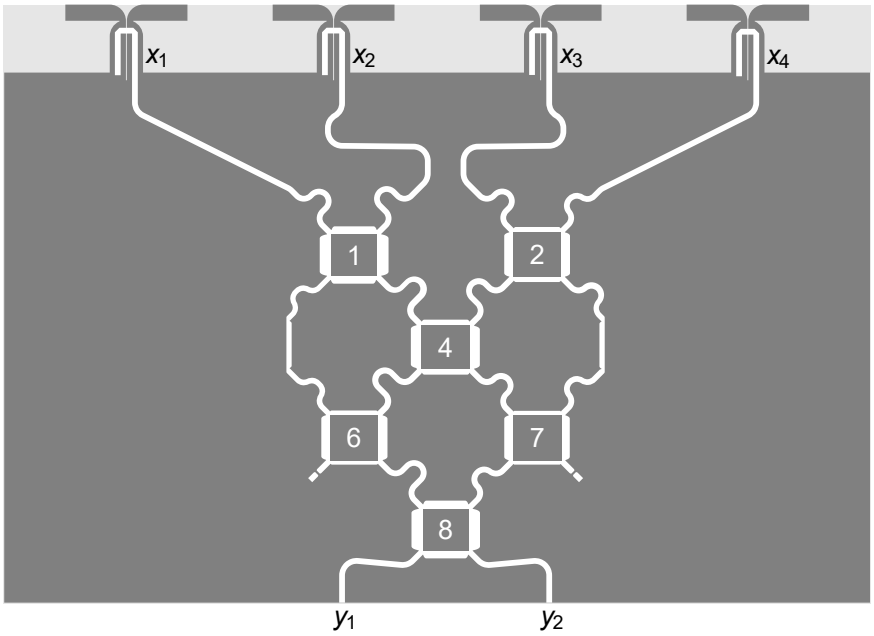


Figure 8.6. Two-layer microwave circuit for the complete compressive array with tile and port numbers indicated. Top-layer copper and coupler numbers are shown in white and bottom-layer copper in dark grey. The substrate measures 268 mm by 185 mm.

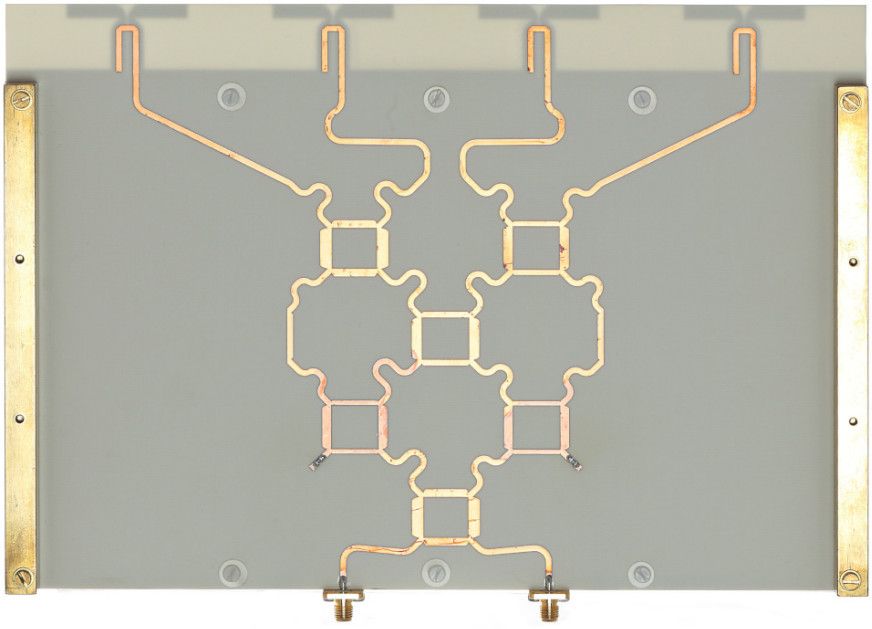


Figure 8.7. Photograph of the manufactured compressive array.

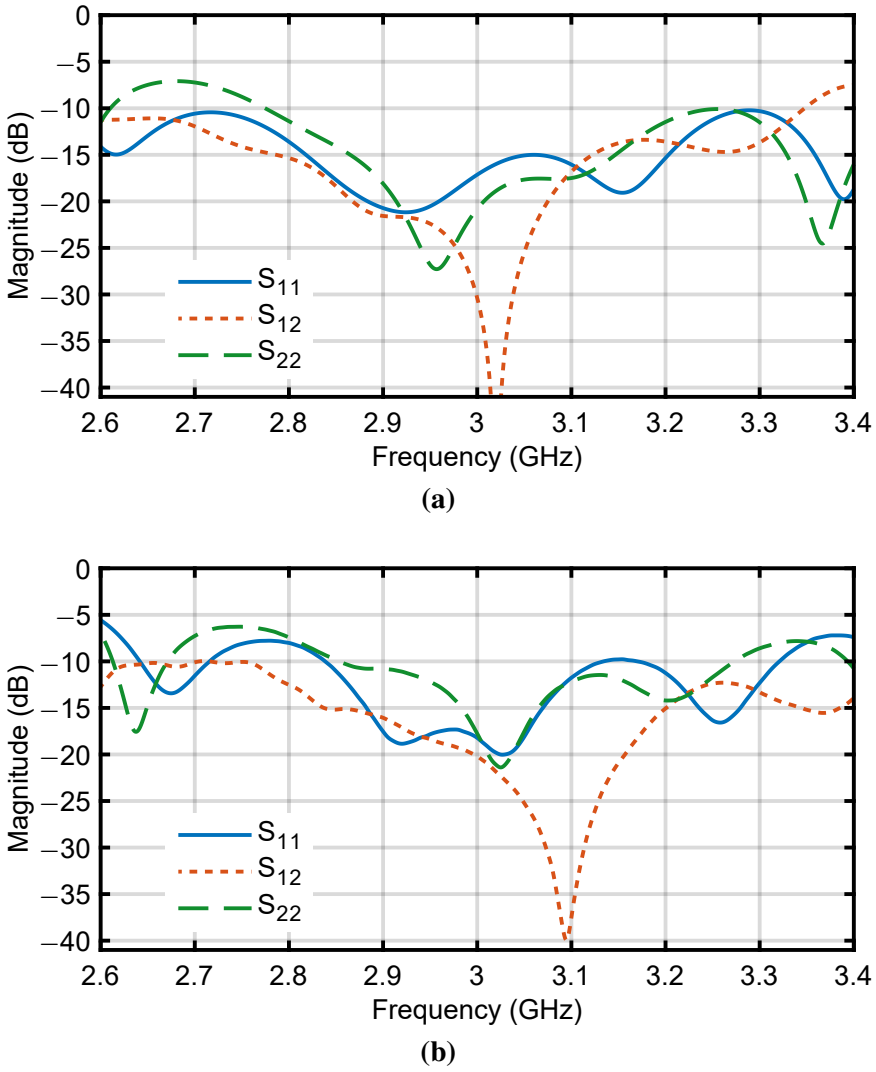
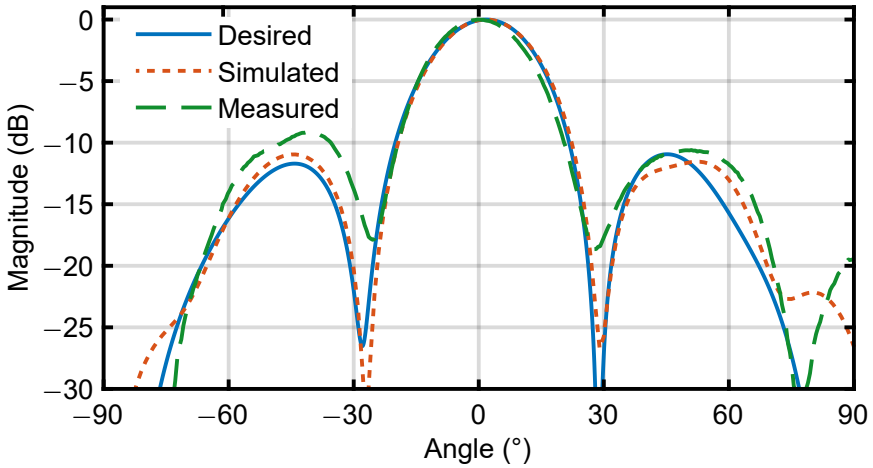
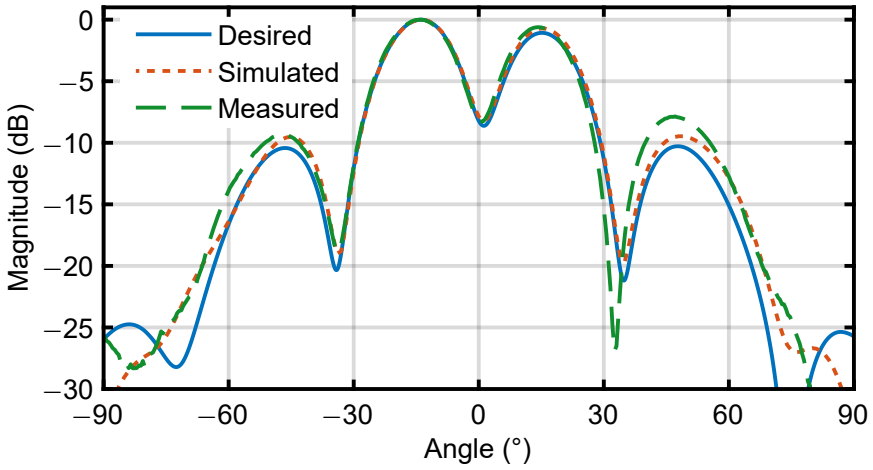


Figure 8.8. (a) Simulated and (b) measured scattering parameters of the compressive array.

The broadband results show that a manufactured compressive array is able to achieve the same minimum SLL as a simulated compressive array, although a frequency shift of 50 MHz in the minimum-SLL frequency was observed. This frequency shift could be due to manufacturing tolerances or an inaccurate estimation of the dielectric constant of the substrate. The simulated compressive array has significantly better bandwidth results than the manufactured compressive array, which is most likely a result of manufacturing tolerances. Even so, a bandwidth of 7.9% is still considered reasonable considering that the network is made up of narrowband single-section branchline couplers, which are generally limited to 10-20% bandwidth each [60]. Should the compressive array be used to steer multiple narrowband signals within a larger bandwidth, the correction weights could be re-calculated for each signal's centre frequency to allow the compressive array to be used over a wider frequency range.



(a)



(b)

Figure 8.9. Normalised desired, simulated and measured radiation patterns at 3 GHz for subarray (a) 1 and (b) 2.

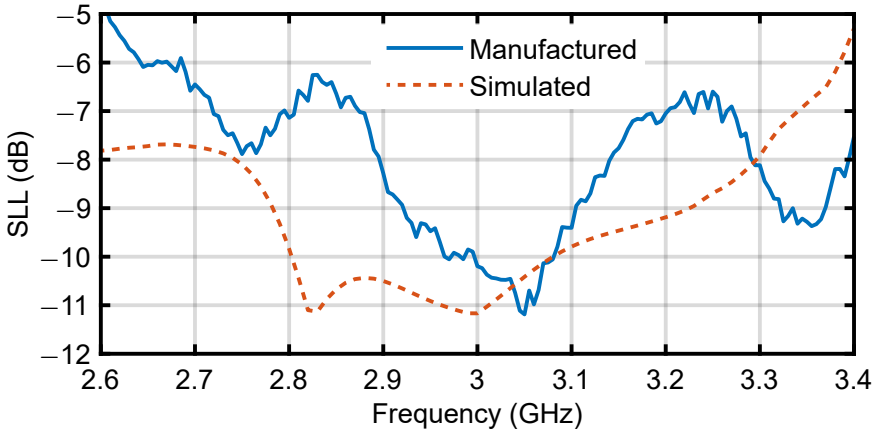


Figure 8.10. SLL of the simulated and manufactured compressive arrays as a function of frequency.

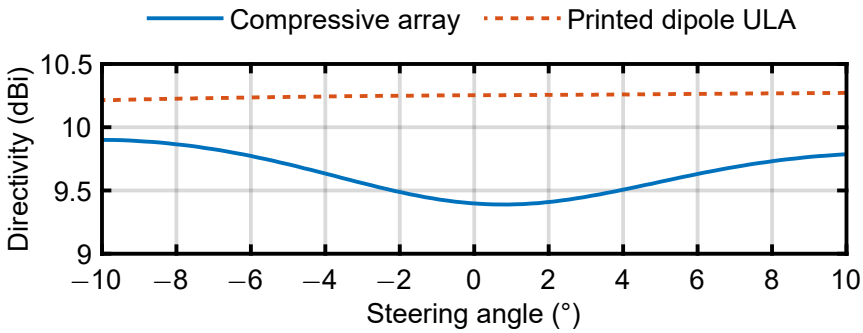


Figure 8.11. Directivity of the steered patterns for simulated conventional and compressive arrays as a function of steering angle at 3 GHz.

Table 8.1. SLL at 3 GHz at various stages of the design process.

Design aspect	SLL (dB)	Deviation from theoretical SLL (dB)
Theoretical feed network response	−11.3	–
Numerical chequered network	−10.8	0.5
Simulated compressive array	−11.2	0.1
Manufactured compressive array	−10.2	1.1

8.6.4 Steered-pattern results at 3 GHz

This section focuses on the properties of the steered patterns at the design frequency of 3 GHz.

Figure 8.11 shows the directivity of the steered patterns for the simulated compressive array at 3 GHz as a function of steering angle. Also shown is the directivity of the steered patterns for a four-element, uniform-excitation ULA which is the same array that is attached to the compressive feed network. Simulated patterns are used because the antenna elements were not manufactured in isolation to be measured separately, and the measurement set-up did not allow for measurement of the full three-dimensional radiation patterns that are required for calculating directivity numerically. Nevertheless, the good agreement observed between the simulated and measured subarray azimuth patterns suggests that the simulated three-dimensional subarray patterns are also accurate.

The directivity of the compressive array ranges from 9.4 dBi to 9.9 dBi and is the lowest near broadside. The directivity of the printed dipole ULA varies less and ranges from 10.2 dB to 10.3 dB across the steering range. The SLL of the ULA across the steering range is −11.4 dB which is only 0.2 dB better than that of the simulated compressive array. The compressive array is therefore able to halve the number of beamforming controls while sacrificing only 0.8 dB in worse-case directivity and maintaining a similar SLL across the steering range. A limitation of this result is that directivity does not take losses into account. The simulated realised gains (which includes the effect of losses) of the elements in the ULA were within 0.1 dB of their simulated directivities. The simulated realised gains of the subarrays were up to 2 dB worse than the simulated directivities of the subarrays.

Table 8.1 gives the 3-GHz SLL at various stages of the design. The theoretical SLL of the feed network response obtained from the compressive array design algorithm is −11.3 dB. After implementing the desired response in a chequered network, the SLL worsens by 0.5 dB to −10.8 dB. Interestingly, subsequently implementing the chequered network in microstrip and simulating the complete compressive

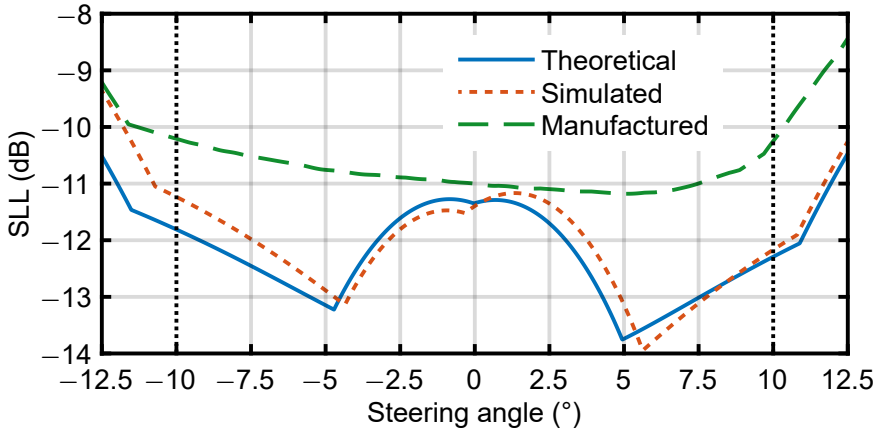


Figure 8.12. SLL as a function of steering angle at 3 GHz.

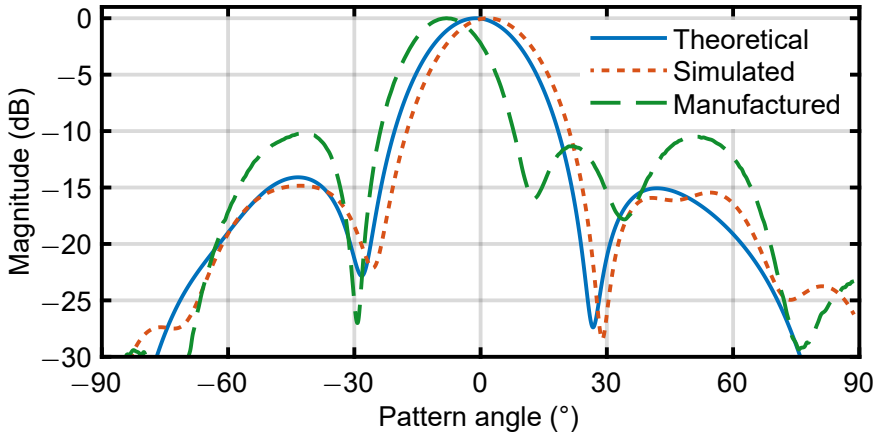


Figure 8.13. Steered patterns with the worst SLL at 3 GHz.

array results in an SLL of −11.2 dB which is within 0.1 dB of the theoretical SLL. This does not necessarily mean that the response of the simulated network moved closer to the desired response, but it does mean that the response deviated in a direction that also happened to give good SLL. The manufactured compressive array had an SLL within 1.1 dB of the theoretical SLL at 3 GHz.

Figure 8.12 shows the SLL of the steered patterns as a function of steering angle. The theoretical and simulated arrays have the worst SLL near broadside, whereas the manufactured array has the worst SLL at the extreme steering angles. As expected, the SLL of the arrays start to deteriorate rapidly near the extremes of the steering range. The SLL of the manufactured array starts deteriorating rapidly near about +9.5°, which shows the importance of specifying a slightly wider steering range than required. Figure 8.13 shows the steered patterns with the worst SLL for the theoretical, simulated, and manufactured arrays which occur at steering angles of −0.8°, 1.3°, and −10.1°, respectively. The SLL near broadside steering for the theoretical and simulated arrays is dominated by the shoulders of the main beam and not the sidelobes themselves. The beamwidth of the worst steered pattern for the manufactured array is narrower than that of the theoretical and simulated array so that the SLL is dominated by the amplitude of the sidelobes.

Figure 8.14 shows the beamwidths in $u = \sin(\theta)$ at the various stages of the design process. At all stages, the beamwidth peaks near broadside steering and is lowest at the extreme steering angles. The worst-case beamwidths for the theoretical, simulated, and manufactured arrays are 0.394, 0.388, and

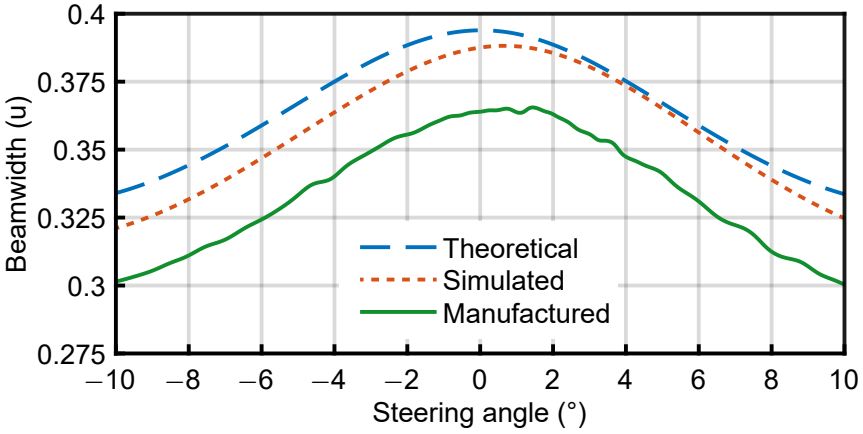


Figure 8.14. Beamwidths in $u = \sin(\theta)$ as a function of steering angle at 3 GHz.

0.366, respectively. The simulated array has marginally better beamwidths than the theoretical array, and the manufactured array has a worst-case beamwidth 7.1% better than the theoretical array.

Although the steering angle corresponding to the worst steered pattern for the manufactured array is $-10.1°$, the peak of the steered pattern is in fact at $-8.1°$. Figure 8.15 shows the peak angles of the steered patterns as a function of the intended steering angles. To accurately determine the locations of the pattern peaks, measurement noise was removed by filtering the steered patterns with second-order Savitzky-Golay filters with frame lengths of 30° [84]. Instead of the expected linear relationship between the intended and realised peak angles, the relationship is non-linear and the realised peak angles tend toward saturation at extreme steering angles. This means, for example, that in order to realise a steered-beam peak at $-8.1°$, the steering weights corresponding to a steering angle of $-10.1°$ should be applied.

Figure 8.16 shows the squint of the steered beams as a function of steering angle. The theoretical, simulated, and manufacted arrays have a maximum absolute squint of 1.1°, 1.2°, and 2° across the steering range, respectively. This means that the useful steering range is reduced by up to 1.1°, 1.2°, and 2° on each side, respectively.

The peaks of the steered patterns are maintained at the desired steering angles by means of the constant-subarray-power-sum constraints (6.9) and (6.10). The normalised sums of the subarray pattern powers at various stages of the design are shown in Figure 8.17. The sum of subarray pattern powers for the theoretical array has a range of 0.2 dB as specified, which leads to a maximum absolute squint of 1.1°. The sum of subarray pattern powers for the simulated array has a range of 0.28 dB, which leads to a slightly larger maximum absolute squint of 1.2°. The sum of subarray pattern powers for the manufactured array has the largest range of 0.53 dB, leading to a maximum absolute squint of 2°. By comparing Figures 8.16 and 8.17, it can be seen that the steered beams have zero squint where the gradients of the subarray pattern power sums are zero, e.g. near broadside and ±7.5° for the theoretical array and near −5° for the manufactured array. This validates the assumption that steered patterns have zero squint where the subarray pattern power sums are locally constant (Section 6.3.1).

The constant-subarray-power-sum constraint is only maintained in the steering range to allow the suppression of the subarray patterns in the out-of-scan regions. This out-of-scan suppression occurs at all stages of the design as shown by the suppression of the subarray pattern power sums outside $|\theta| < 10°$ in Figure 8.17. At the leftmost steering angle, the left side of the steered beam is suppressed by the subarray patterns, thereby squinting the beam towards the right (a positive squint). At the rightmost steering angle, the right side of the steered beam is suppressed by the subarray patterns,

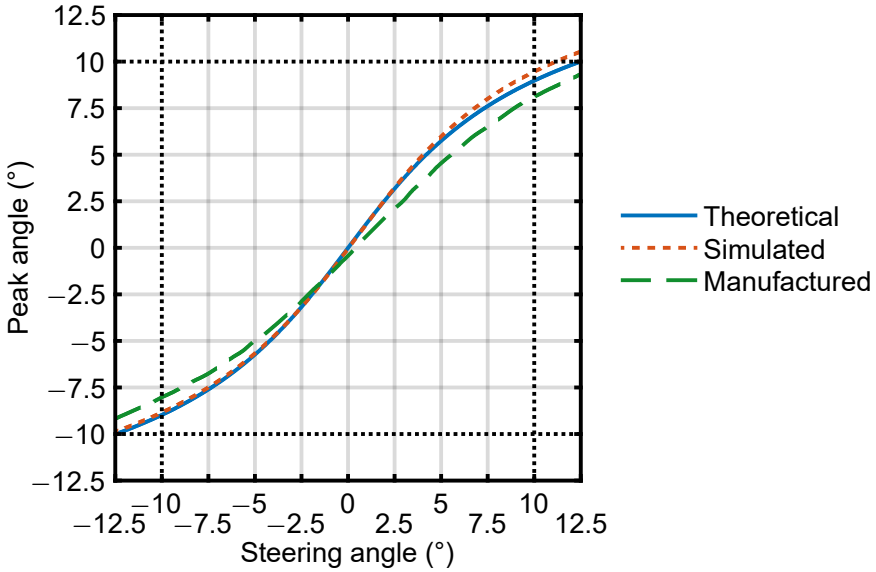


Figure 8.15. Peak pattern angle as a function of steering angle at 3 GHz.

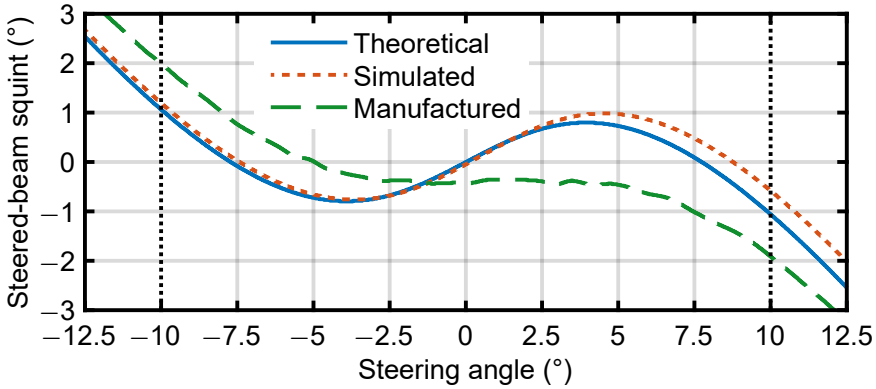


Figure 8.16. Steered-pattern squint at 3 GHz.

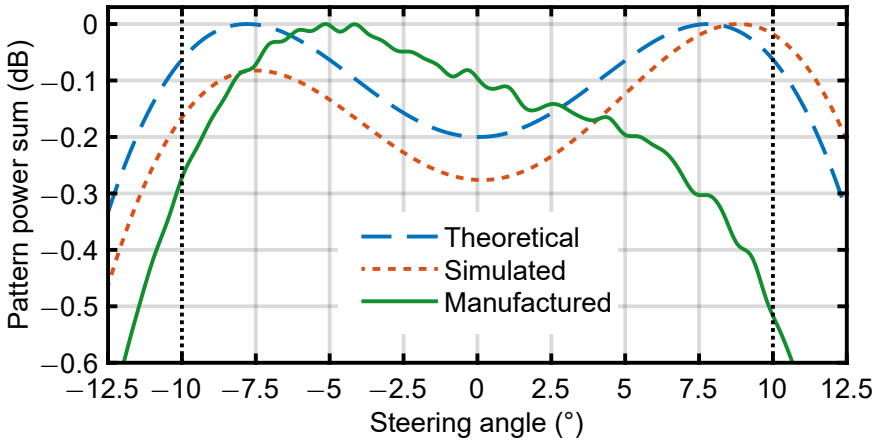


Figure 8.17. Normalised sums of subarray pattern powers at 3 GHz.

squinting the beam towards the left (a negative squint). This also explains the decreased beamwidths at extreme steering angles as seen in Figure 8.14. In the case of the manufactured array, the suppression of the subarray patterns occurs prematurely within the steering range, leading to worse squint than expected. To account for this deviation as a result of manufacturing tolerances, it may be helpful to constrain the sum of subarray pattern powers to be constant across a range slightly wider than the steering range.

8.7 CHAPTER SUMMARY

In this chapter, results have been presented for the first successfully manufactured prototype compressive array with integrated antenna elements. The use of embedded element patterns in the design confirms that the compressive array design algorithm does not require the assumption of identical element patterns. Good agreement was observed between the theoretical, simulated and measured results for the assembled compressive array, showing that compressive arrays are feasible to implement in practice. The use of simulated element patterns to design the feed network resulted in acceptable measured results, which simplifies the design process and allows a compressive array to be manufactured as a single component. The manufactured compressive array achieved an SLL within 1.1 dB of the theoretical value at 3 GHz, and the return losses and isolation were better than -17.7 dB at 3 GHz. The simulated and manufactured compressive arrays achieved the same minimum SLL of -11.2 dB, but the minimum SLL for the manufactured array occurred at 50 MHz above the design frequency. The manufactured compressive array had a 3-dB SLL bandwidth of 7.9% even though it was only designed to operate at 3 GHz, showing that compressive arrays have the potential to operate at a range of frequencies. The manufactured array exhibited a maximum absolute squint in the steered patterns of 2° compared to a theoretical maximum squint of 1.1°, thereby reducing the useful steering range. This could be improved by constraining the sum of subarray pattern powers across a wider angular range, or investigating alternative ways of constraining the pattern peaks to occur at the desired angles.

CHAPTER 9 CONCLUSION

9.1 CONCLUSIONS

In this thesis, a thorough investigation into the design and feasibility of compressive arrays has been presented. The problem has been formulated to consider conventional array characteristics while exploiting the benefits of the unconventional concept of CS.

This work represents the first investigation into compressive arrays that considers their feasibility in terms of traditional antenna array characteristics such as SLL, directivity, and beamwidth. A framework has been proposed for the optimisation of compressive arrays as general-purpose beamforming arrays with arbitrary sidelobe specifications, along with an algorithm for performing such optimisation.

A major challenge in the optimisation of compressive arrays is the fact that the patterns must be optimised for explicitly at all steering angles, leading to a complex problem which is combinatorial in nature. In order to develop a computationally feasible compressive array optimisation algorithm, it was first necessary to address the issue of computational complexity in the state-of-the-art BCASC coherence optimisation algorithm. This was done by proposing a novel coherence optimisation algorithm to directly minimise coherence, which in turn inspired the proposed compressive array optimisation algorithm.

Compressive arrays were shown to improve on existing reduced-control array designs, which was achieved by removing all constraints on the feed network response. However, this opened up the question of whether it was possible to implement such unconstrained responses in hardware. This question was addressed by proposing a novel overlapped feed network layout consisting of interconnected couplers and phase shifters. Such chequered networks were shown to be able to implement arbitrary feed network responses, making them suitable for implementing compressive arrays. An advantage of this approach is that the tasks of pattern synthesis and feed network implementation are separated, thereby simplifying the design process.

The study was concluded by designing and manufacturing a prototype compressive array, which highlighted a number of practical challenges associated with compressive arrays. The prototype was successfully implemented and obtained satisfactory results despite manufacturing tolerances, showing that compressive arrays are indeed feasible to implement in practice.

The main contributions of the work are summarised below.

Suitability of random compression weights and compressive arrays based on optimised codebooks: A compressive array designed using an optimised codebook was shown to improve on random real- and complex-valued compressive arrays in terms of mean DF accuracy. A compressive array optimised for low SLL showed a significant improvement over the random compressive arrays in both mean and worst-case DF accuracy, converging with SNR to one-fifth of the RMS error of a conventional M-element array. Neither real- nor complex-valued compression weights can, therefore, be assumed to be optimal for compressive arrays of practical sizes.

Addressing the computational complexity in codebook optimisation algorithms: A novel algorithm has been proposed for optimising codebooks which directly minimises coherence instead of maximising the Euclidean distance between codewords. The proposed codebook optimisation algorithm resulted in almost an order-of-magnitude reduction in median run time compared to the BCASC algorithm while maintaining similar coherence results. Optimising directly for coherence is thus a better approach towards designing optimal codebooks.

Developing a compressive array SLL minimisation algorithm with arbitrary sidelobe shaping: Inspired by the proposed coherence-based codebook optimisation algorithm, the optimisation of discrete codewords in a codebook was extended to the optimisation of SLL in a compressive array. Coherence and SLL become equivalent when the beamforming weight vectors are defined as the complex conjugates of the corresponding steering vectors, and when the steering vectors are constrained to have unit length. The only difference is that SLL is taken across continuous steering and pattern angles such that a start of the SLL region must be defined. Compressive arrays are a generalisation of reduced-control arrays and are able to improve on existing reduced-control array designs by removing all constraints on the feed network response. Increasing the number of antenna elements increases the control over the array patterns for the same number of beamforming controls, so that a compressive array is able to improve on the performance of a conventional ULA for the same number of beamforming controls. Non-uniform sidelobes may be realised by applying a sidelobe-shaping mask which varies with both steering and pattern angle. Compressive arrays may also have subarray patterns that are constrained outside the steering range, allowing an interferer to suppressed in hardware before sampling, for example.

Hardware implementation of arbitrary compression weights: A novel network of interconnected couplers and fixed phase shifters has been proposed for implementing arbitrary complex-valued feed network responses. An algorithm has been proposed for optimising chequered networks to implement a desired response while minimising the range of coupling ratios and phase shifts required, which facilitates the practical realisation of compressive arrays in hardware. The claim that chequered networks are able to implement arbitrary responses was substantiated by combining three independently designed arrays to share the same 8-element aperture by means of a 3×8 chequered feed network. A 2×4 microstrip chequered network for a ULA was manufactured, achieving the desired response within 0.6 dB and 5.2°. These results show that chequered networks are able to implement arbitrary feed network responses and are feasible to implement in hardware.

Practical feasibility of compressive arrays: The successful design and manufacturing of a 3-GHz prototype compressive array with integrated antenna elements showed that compressive arrays are feasible to implement in microwave hardware. The compressive array design algorithm is able to use simulated embedded element patterns which are not identical. Although only designed to operate at 3 GHz, the compressive array had a 3-dB SLL bandwidth of 7.9%, showing that compressive arrays have the potential to perform well across a range of frequencies. Comparing the steered patterns of a simulated compressive array against the steered patterns of the ULA without the compressive feed network, it was found that the compressive array was able to halve the number of beamforming controls for only a 0.8 dB reduction in directivity, although this excludes the effect of losses. A number of practical design considerations have been highlighted. Where multiple feed network designs exist with equally good SLL, candidate designs should be evaluated in terms of the range of coupling ratios and phase shifts that each design requires when implemented as a chequered network. The steering range should be specified slightly wider than required to avoid rapid deterioration of the SLL at extreme steering angles due to manufacturing tolerances. Steered-beam squint has been shown to be related to the gradient of the sum of subarray pattern powers (or steering vector lengths). Squint in the steered beams may limit the useful steering range due to the non-linear relationship between the intended and realised steering angles.

9.2 FUTURE WORK

Being the first investigation into the design of compressive arrays as a pattern-synthesis problem, many questions have arisen that fall outside the scope of the current study. These questions are listed below as a hypothesis and suggestion as to the direction future work could take.

For what size arrays is the compressive array design algorithm and chequered network design algorithm feasible in terms of computational complexity? The proposed algorithms for designing and implementing compressive feed networks are computationally expensive due to their combinatorial nature. The compressive array design algorithm must consider all relevant combinations of steering and pattern angles, and the chequered network design algorithm must consider all the signal paths that determine the total feed network response. A crucial question that remains unanswered is for what size arrays the proposed algorithms will remain useful before their computational complexity becomes prohibitive. However, even if the proposed methods are restricted to the smallest of arrays, this remains useful as the smallest arrays require the most degrees of freedom to be exploited in order to obtain reasonable performance (Sections 7.4.3 and 8.2).

Can compressive arrays be designed for predictable broadband performance? The results for the prototype compressive array presented in Chapter 8 shows that a compressive array is able to obtain reasonable performance across a range of frequencies. However, this was coincidental as the array was only optimised to perform well at 3 GHz. Where a compressive array is intended to work at a range of frequencies, broadband performance must be explicitly incorporated into the design process. This would entail designing a feed network response that takes into account frequency-dependent antenna element patterns.

Is it possible and computationally feasible to optimise beamforming weights separately? Currently, beamforming weights are assumed to be the complex conjugates of the steering vectors at the corresponding steering angles. However, it was suggested in Chapter 8 that the beamforming weights should be adjusted to account for squint, which does not hold to this assumption. Although it would increase the computational complexity of the problem, optimising beamforming weights separately may result in better performance, especially as the number of beamforming controls increases.

Is there a better way to account for steered-beam squint? Currently, steered-beam squint is constrained indirectly by the constant-subarray-pattern-power-sum constraint. This is problematic as it is not always possible to reduce the range of subarray pattern power sums to an arbitrarily small value, especially where simulated or measured element patterns are used. It would be preferable to explicitly account for steered-beam squint in the optimisation, which would probably include the optimisation of beamforming weights as mentioned above.

Is it possible and useful to perform sensitivity analyses on theoretical and chequered-network feed network responses? Currently, feed network responses and their chequered-network implementations are accepted based on their predicted performance derived from analytical models or simulation, an approach that has thus far produced acceptable results. However, manufacturing tolerances do contribute towards reducing performance. It is expected that some designs, both in terms of their theoretical feed network responses and chequered-network implementations, may be more sensitive to variations in the design parameters (compression weights, coupling ratios, phase shifts, and lengths and widths of transmission lines) than others. A sensitivity analysis would be able to predict in a statistical manner how the performance may be expected to vary along with the parameters of a manufactured circuit, allowing designs to be chosen based on their sensitivity to manufacturing tolerances.

REFERENCES

[1] H. E. A. Laue and W. P. du Plessis, "Numerical optimization of compressive array feed networks," *IEEE Trans. Antennas Propag.*, vol. 66, no. 7, pp. 3432–3440, July 2018.

[2] R. J. Mailloux, *Phased Array Antenna Handbook*. Norwood, MA, USA: Artech House, 2005.

[3] J. S. Herd and M. D. Conway, "The evolution to modern phased array architectures," *Proc. IEEE*, vol. 104, no. 3, pp. 519–529, Mar. 2016.

[4] A. van Ardenne, J. D. Bregman, W. A. van Cappellen, G. W. Kant, and J. G. B. de Vaate, "Extending the field of view with phased array techniques: Results of European SKA research," *Proc. IEEE*, vol. 97, no. 8, pp. 1531–1542, Aug. 2009.

[5] B. Friedlander, "Wireless direction-finding fundamentals," in *Classical and Modern Direction-of-Arrival Estimation*, T. E. Tuncer and B. Friedlander, Eds. Burlingston, MA: Elsevier, 2009, ch. 1, pp. 1–51.

[6] C. Fulton, M. Yeary, D. Thompson, J. Lake, and A. Mitchell, "Digital phased arrays: Challenges and opportunities," *Proc. IEEE*, vol. 104, no. 3, pp. 487–503, Mar. 2016.

[7] T. Azar, "Overlapped subarrays: Review and update [Education column]," *IEEE Antennas Propag. Mag.*, vol. 55, no. 2, pp. 228–234, Apr. 2013.

[8] Y. T. Lo, "Aperiodic Arrays," in *Antenna Handbook*, Y. T. Lo and S. W. Lee, Eds. New York, NY: Van Nostrand Reinhold Company Inc., 1988, ch. 14, pp. 14-1–14-37.

[9] W. P. M. N. Keizer, "Linear array thinning using iterative FFT techniques," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2757–2760, Aug. 2008.

[10] Y. Lo, "A mathematical theory of antenna arrays with randomly spaced elements," *IEEE Trans. Antennas Propag.*, vol. 12, no. 3, pp. 257–268, May 1964.

[11] Y. Wang, G. Leus, and A. Pandharipande, "Direction estimation using compressive sampling array processing," in *IEEE/SP 15th Work. Stat. Signal Process.*, Aug. 2009, pp. 626–629.

[12] R. G. Baraniuk, "Compressive sensing [Lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, July 2007.

[13] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[14] J. Steckel, D. Laurijssen, A. Schenck, N. BniLam, and M. Weyn, "Low-cost hardware platform for angle of arrival estimation using compressive sensing," in *Eur. Conf. Antennas Propag.*, Apr. 2018.

[15] C. A. Olen and R. T. Compton, "A numerical pattern synthesis algorithm for arrays," *IEEE Trans. Antennas Propag.*, vol. 38, no. 10, pp. 1666–1676, Oct. 1990.

[16] H. Zörlein and M. Bossert, "Coherence optimization and best complex antipodal spherical codes," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6606–6615, Dec. 2015.

[17] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[18] H. Zörlein, D. E. Lazich, and M. Bossert, "On the noise-resilience of OMP with BASC-based low coherence sensing matrices," in *Proc. 10th Int. Conf. Sampling Theory Appl. (SampTA)*, July 2013, pp. 468–471.

[19] H. A. Zörlein, "Channel Coding Inspired Contributions to Compressed Sensing," Ph.D. dissertation, Inst. Commun. Eng., Ulm Univ., Ulm, Germany, 2015.

[20] O. M. Bucci, G. D'Elia, G. Mazzarella, and G. Panariello, "Antenna pattern synthesis: a new general approach," *Proc. IEEE*, vol. 82, no. 3, pp. 358–371, Mar. 1994.

[21] S. P. Skobelev, "Methods of constructing optimum phased-array antennas for limited field of view," *IEEE Antennas Propag. Mag.*, vol. 40, no. 2, pp. 39–50, Apr. 1998.

[22] H. E. A. Laue and W. P. du Plessis, "Compressive direction-finding antenna array," in *IEEE-APS Topical Conf. Antennas Propag. Wireless Commun. (APWC)*, Sep. 2016, pp. 158–161.

[23] H. E. A. Laue and W. P. du Plessis, "A coherence-based algorithm for optimizing rank-1 Grassmannian codebooks," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 823–827, June 2017.

[24] H. E. A. Laue and W. P. du Plessis, "A checkered network for implementing arbitrary overlapped feed networks," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 11, pp. 4632–4640, Nov 2019.

[25] H. E. A. Laue, "Demystifying compressive sensing [Lecture notes]," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 171–176, July 2017.

[26] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. E. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[27] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: From coding to source separation," *Proc. IEEE*, vol. 98, no. 6, pp. 995–1005, June 2010.

[28] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.

[29] A. Massa, P. Rocca, and G. Oliveri, "Compressive sensing in electromagnetics—A review," *IEEE Antennas Propag. Mag.*, vol. 57, no. 1, pp. 224–238, Feb. 2015.

[30] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proc. Nat. Acad. of Sci.*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.

[31] S. Foucart, "A note on guaranteed sparse recovery via ℓ_1-minimization," *Appl. and Computational Harmonic Anal.*, vol. 29, no. 1, pp. 97–103, July 2010.

[32] M. Fornasier and H. Rauhut, "Compressive sensing," in *Handbook of Mathematical Methods in Imaging*, O. Scherzer, Ed. New York, NY: Springer New York, 2011, pp. 187–228.

[33] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, May 2015.

[34] C. L. Dolph, "A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level," *Proc. IRE*, vol. 34, no. 6, pp. 335–348, June 1946.

[35] M. I. Skolnik, J. W. Sherman, III, and F. C. Ogg, Jr, "Statistically designed density-tapered arrays," *IEEE Trans. Antennas Propag.*, vol. 12, no. 4, pp. 408–417, July 1964.

[36] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Trans. Antennas Propag.*, vol. 42, no. 7, pp. 993–999, July 1994.

[37] V. Murino, A. Trucco, and A. Tesei, "Beam pattern formulation and analysis for wide-band beamforming systems using sparse arrays," *Signal Process.*, vol. 56, no. 2, pp. 177–183, Jan. 1997.

[38] W. P. du Plessis, "Weighted thinned linear array design with the iterative FFT technique," *IEEE Trans. Antennas Propag.*, vol. 59, no. 9, pp. 3473–3477, Sep. 2011.

[39] R. J. Mailloux, *Electronically Scanned Arrays*. San Rafael, USA: Morgan and Claypool Publishers, 2007.

[40] S. P. Skobelev, *Phased Array Antennas with Optimized Element Patterns*. Norwood, USA: Artech House, 2011.

[41] P. Rocca, G. Oliveri, R. J. Mailloux, and A. Massa, "Unconventional phased array architectures and design methodologies—a review," *Proc. IEEE*, vol. 104, no. 3, pp. 544–560, Mar. 2016.

[42] D. G. Leeper, "Isophoric arrays-massively thinned phased arrays with well-controlled sidelobes," *IEEE Trans. Antennas Propag.*, vol. 47, no. 12, pp. 1825–1835, Dec 1999.

[43] G. Oliveri, M. Donelli, and A. Massa, "Linear array thinning exploiting almost difference sets," *IEEE Trans. Antennas Propag.*, vol. 57, no. 12, pp. 3800–3812, Dec 2009.

[44] G. Oliveri and A. Massa, "Genetic algorithm (GA)-enhanced almost difference set (ADS)-based approach for array thinning," *IET Microw., Antennas Propag.*, vol. 5, no. 3, pp. 305–315, Feb 2011.

[45] Y. Liu, Z. Nie, and Q. H. Liu, "Reducing the number of elements in a linear antenna array by the matrix pencil method," *IEEE Trans. Antennas Propag.*, vol. 56, no. 9, pp. 2955–2962, Sep 2008.

[46] G. Prisco and M. D'Urso, "Maximally sparse arrays via sequential convex optimizations," *IEEE Antennas Wireless Propag. Lett.*, vol. 11, pp. 192–195, Feb 2012.

[47] B. Fuchs, "Synthesis of sparse arrays with focused or shaped beampattern via sequential convex optimizations," *IEEE Trans. Antennas Propag.*, vol. 60, no. 7, pp. 3499–3503, July 2012.

[48] R. J. Mailloux, S. G. Santarelli, and T. M. Roberts, "Wideband arrays using irregular (polyomino) shaped subarrays," *Electron. Lett.*, vol. 42, no. 18, pp. 1019–1020, Aug. 2006.

[49] A. P. Goffer, M. Kam, and P. R. Herczfeld, "Design of phased arrays in terms of random subarrays," *IEEE Trans. Antennas Propag.*, vol. 42, no. 6, pp. 820–826, June 1994.

[50] Z. Kachwalla, "A limited-scan linear array using overlapping subarrays," *J. Elect. and Electron. Eng., Australia*, pp. 126–131, June 1983.

[51] J. S. Herd, S. M. Duffy, and H. Steyskal, "Design considerations and results for an overlapped subarray radar antenna," in *2005 IEEE Aerospace Conf.*, Mar. 2005, pp. 1087–1092.

[52] H. Southall and D. McGrath, "An experimental completely overlapped subarray antenna," *IEEE Trans. Antennas Propag.*, vol. 34, no. 4, pp. 465–474, Apr. 1986.

[53] S. M. Duffy, D. D. Santiago, and J. S. Herd, "Design of overlapped subarrays using an RFIC beamformer," in *2007 IEEE Antennas Propag. Soc. Int. Symp.*, June 2007, pp. 1949–1952.

[54] R. Mailloux, "Array grating lobes due to periodic phase, amplitude, and time delay quantization," *IEEE Trans. Antennas Propag.*, vol. 32, no. 12, pp. 1364–1368, Dec. 1984.

[55] R. Haupt, "Reducing grating lobes due to subarray amplitude tapering," *IEEE Trans. Antennas Propag.*, vol. 33, no. 8, pp. 846–850, Aug. 1985.

[56] P. Rocca, M. D'Urso, and L. Poli, "Advanced strategy for large antenna array design with subarray-only amplitude and phase control," *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 91–94, Jan. 2014.

[57] S. P. Skobelev, "Analysis and synthesis of an antenna array with sectoral partial radiation patterns," *Telecommun. and Radio Eng.*, vol. 45, no. 11, pp. 116–119, Nov. 1990.

[58] D. Petrolati, P. Angeletti, and G. Toso, "A novel lossless BFN for linear arrays based on overlapped sub-arrays," in *8th European Conf. Antennas Propag. (EuCAP 2014)*, Apr. 2014, pp. 2247–2250.

[59] S. P. Skobelev, "Some features of the overlapped subarrays built up of beam-forming matrices for shaping flat-topped radiation patterns," *IEEE Trans. Antennas Propag.*, vol. 63, no. 12, pp. 5529–5535, Dec. 2015.

[60] D. M. Pozar, *Microwave Engineering*. Hoboken, NJ, USA: John Wiley and Sons, Inc., 2012.

[61] W. P. Delaney, "An RF multiple beam-forming technique," *IRE Trans. Military Electron.*, vol. MIL-6, no. 2, pp. 179–186, Apr. 1962.

[62] T. N. Kaifas and J. N. Sahalos, "On the design of a single-layer wideband Butler matrix for switched-beam UMTS system applications [Wireless corner]," *IEEE Antennas Propag. Mag.*, vol. 48, no. 6, pp. 193–204, Dec. 2006.

[63] B. Piovano, L. Accatino, F. Muoio, G. Caille, and M. Mongiardo, "CAD and mechanical realization of planar, Ka-band, 8x8 Butler matrices," in *2002 32nd Eur. Microw. Conf.*, Sep. 2002, pp. 1–4.

[64] N. BniLam, J. Steckel, and M. Weyn, "Synchronization of multiple independent subarray antennas: An application for angle of arrival estimation," *IEEE Trans. Antennas Propag.*, vol. 67, no. 2, pp. 1223–1232, Feb. 2019.

[65] J. H. Conway, R. H. Hardin, and N. J. Sloane, "Packing lines, planes, etc.: Packings in Grassmannian spaces," *Experiment. Math.*, vol. 5, no. 2, pp. 139–159, Jan. 1996.

[66] A. Medra and T. N. Davidson, "Flexible codebook design for limited feedback systems via sequential smooth optimization on the Grassmannian manifold," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1305–1318, Mar. 2014.

[67] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2735–2747, Oct. 2003.

[68] P. Xia, S. Zhou, and G. B. Giannakis, "Achieving the Welch bound with difference sets," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1900–1907, May 2005.

[69] I. S. Dhillon, J. R. Heath, T. Strohmer, and J. A. Tropp, "Constructing packings in Grassmannian manifolds via alternating projection," *Experiment. Math.*, vol. 17, no. 1, pp. 9–35, Jan. 2008.

[70] B. Dumitrescu, "Designing incoherent frames with only matrix-vector multiplications," *IEEE Signal Process. Lett.*, vol. 24, no. 9, pp. 1265–1269, Sep. 2017.

[71] A. Xenaki, P. Gerstoft, and K. Mosegaard, "Compressive beamforming," *J. Acoust. Soc. Am.*, vol. 136, no. 1, pp. 260–271, July 2014.

[72] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer-Verlag, 1999.

[73] W. P. du Plessis, "Efficient computation of array factor and sidelobe level of linear arrays [EM programmer's notebook]," *IEEE Antennas Propag. Mag.*, vol. 58, no. 6, pp. 102–114, Dec. 2016.

[74] P. E. Howland, D. Maksimiuk, and G. Reitsma, "FM radio based bistatic radar," *IEE Proc.—Radar Sonar Navig.*, vol. 152, no. 3, pp. 107–115, June 2005.

[75] *Nonlinear Systems with Constraints*, MathWorks Inc., Natick, USA, 2018. [Online]. Available: https://www.mathworks.com/help/optim/ug/nonlinear-systems-with-constraints.html

[76] *MultiStart*, MathWorks Inc., Natick, USA, 2018. [Online]. Available: https://www.mathworks.com/help/gads/multistart.html

[77] J. P. Shelton, "Fast Fourier transforms and Butler matrices," *Proc. IEEE*, vol. 56, no. 3, pp. 350–350, Mar. 1968.

[78] H. J. Orchard, R. S. Elliott, and G. J. Stern, "Optimising the synthesis of shaped beam antenna patterns," *IEE Proc. H—Microw., Antennas Propag.*, vol. 132, no. 1, pp. 63–68, Feb. 1985.

[79] G. F. Engen and C. A. Hoer, "Thru-reflect-line: An improved technique for calibrating the dual six-port automatic network analyzer," *IEEE Trans. Microw. Theory Techn.*, vol. 27, no. 12, pp. 987–993, Dec. 1979.

[80] D. Jaisson, "Fast design of a printed dipole antenna with an integrated balun," *IEE Proc.—Microw., Antennas Propag.*, vol. 153, no. 4, pp. 389–394, Aug. 2006.

[81] R. Li, T. Wu, B. Pan, K. Lim, J. Laskar, and M. M. Tentzeris, "Equivalent-circuit analysis of a broadband printed dipole with adjusted integrated balun and an array for base station applications," *IEEE Trans. Antennas Propag.*, vol. 57, no. 7, pp. 2180–2184, July 2009.

[82] S. M. Sherman and D. K. Barton, *Monopulse Principles and Techniques*. Norwood, MA, USA: Artech House, 2011.

[83] P. L. D. Abrie, *The Design of Impedance-Matching Networks for Radio-Frequency and Microwave Amplifiers*. Dedham, MA, USA: Artech House, 1985.

[84] S. J. Orfanidis, *Introduction to Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.

GLOSSARY

Antenna array: A group of antennas that are combined on reception or simultaneously excited on transmission, resulting in a *radiation pattern* for the array as a whole.

Antenna element: A single antenna in an *antenna array*.

Antenna pattern: See *radiation pattern*.

Antipodal: Any phase shift of a *codeword*, applied to all the elements in the *codeword*.

Aperture illumination: The complex-valued current distribution that exists across the elements of an *antenna array*.

Array aperture: The physical surface area of an *antenna array* that radiates or receives electromagnetic energy.

Array pattern: See *radiation pattern*.

Beamforming: The process of using an *antenna array* to form a *radiation pattern* with a peak in a desired direction by either combining weighted *antenna-element* signals on reception, or applying excitations to *antenna elements* on transmission.

Beamforming control: Any device used to apply the weights or excitations required for *beamforming*, including phase shifters and/or amplifiers for conventional *phased arrays*, and transmitters and/or receivers for *digital arrays*.

Beamforming weight: The weight or excitation applied by a *beamforming control*.

Beamwidth: The width of the main beam in a *radiation pattern* as measured between the angles where the gain is 3 dB lower than the peak gain.

Codebook: A collection of *codewords*. Rank-1 codebooks can be expressed as matrices, where the columns of the matrix form the *codewords*.

Codeword: A vector or matrix that forms part of a *codebook*, where the number of columns in a codeword is equal to the rank of the *codebook*. Rank-1 codewords are column vectors.

Coherence: A measure of the similarity between *codewords* in a rank-1 *codebook*, measured as the maximum absolute dot product between any two *codewords*.

Completely overlapped subarray: An *overlapped subarray* where each *subarray* is a function of all the *antenna elements*. Represented by a fully populated *sensing matrix*.

Compressed samples: The samples/measurements taken after applying the *sensing matrix* to the *conventional samples*, expressed as $\mathbf{y} = \mathbf{\Phi x}$.

Compressed steering matrix: The matrix $\mathbf{B}$ with the *compressed steering vectors* at the sampled angles as columns.

Compressed steering vector: The received signal vector $\mathbf{b}(\theta)$ at the *subarrays* for a reference wave impinging on the *compressive array* from the direction θ.

Compressed transform matrix: The matrix $\mathbf{B}$ in a *CS* system that relates the *compressed samples* to the *sparse domain* via $\mathbf{y} = \mathbf{Bs}$. Equivalent to the *compressed steering matrix* in antenna arrays.

Compression weights: The elements in the *sensing matrix* $\mathbf{\Phi}$.

Compressive array: A generalisation of *reduced-control arrays* which allows for *complete overlap* and does not necessarily place constraints on the *feed network response*.

Compressive feed network: The *feed network* associated with a *compressive array*, i.e. a *compressive array* excluding the *antenna elements*.

Compressive sensing (CS): A novel sub-Nyquist sampling framework based on the assumption of *sparsity*.

Conventional samples: Uniformly distributed samples that are taken in accordance with the Nyquist theorem.

Coupling ratio: The the ratio between the square-root power delivered to the through and coupled ports in a directional coupler.

Digital array: An *antenna array* with a transmitter and/or receiver at each *antenna element*.

Digital beamforming: The process of combining *antenna-element* or *subarray* signals in software. Requires a receiver at each *antenna element* or *subarray*.

Embedded element pattern: The *radiation pattern* of an *antenna element* with the effects of mutual coupling with neighbouring *antenna elements* included.

Feed network: The hardware that connects the *antenna elements* to the transmitter(s) and/or receiver(s) in an *antenna array* system.

Feed network response: The numerical relationship between the *antenna-element* signals and the signals at the *subarray* ports in a *reduced-control array*, given by the matrix $\mathbf{\Phi}$. Equivalent to the *sensing matrix* in the context of a *compressive array*.

Grassmannian manifold $\mathbb{G}(M, R)$*:* The set of all R-dimensional subspaces in $\mathbb{C}^M$. Where the rank $R = 1$, the manifold is the set of all lines in complex M-dimensional space.

Hard null: A null placed in a *subarray* pattern and realised in hardware.

Overlapped subarray: A *subarray* that shares some *antenna elements* with one or more adjacent *subarrays*.

Partially overlapped subarray: An *overlapped subarray* where only some of the *antenna elements* are shared between adjacent *subarrays*.

Phase centre: The point on an *antenna array* from which the *wavefront* spreads out in a sphere.

Phased array: A traditional *beamforming* array that makes use of hardware phase shifters to perform *beamforming*.

Radiation pattern: The relative gain of an antenna or *antenna array* as a function of angle. *Antenna-element* or *subarray* patterns are found by stimulating only the element or subarray in question and terminating the others in matched loads.

Reduced-control array: Any array that uses less *beamforming controls* for a given *aperture* than a *ULA* that fills the same *aperture*.

Resolution: The smallest angular distance between two sources that still allows them to be resolved as separate sources for a given *antenna array*.

Sensing matrix: The matrix $\mathbf{\Phi}$ which describes the process of combining a large number of *conventional samples* $\mathbf{x}$ into fewer measurements $\mathbf{y} = \mathbf{\Phi x}$. Equivalent to *feed network response* in the context of an *antenna array*.

Shorted coupler: A coupler with an infinite *coupling ratio*, which reduces to two $-90°$ phase shifters in the case of a branchline coupler.

Sidelobe level (SLL): Where a *start of the sidelobe region* is specified, the largest *radiation pattern* amplitude in the pre-defined sidelobe region relative to the main-beam peak. Otherwise, the largest relative amplitude of the lobes other than the main lobe in a *radiation pattern*. When quoted as a single value for a *reduced-control* array, it refers to the worst sidelobe level across all *steering angles*.

Soft null: A null formed by the application of *beamforming weights*, that is present at all *steering angles* except those near the null direction itself.

Sparse domain: A domain in which a signal has only few non-zero components. The basis for a sparse domain is the *sparsifying basis*.

Sparsifying basis: The basis associated with a *sparse domain*.

Sparsity: The concept of a signal having few non-zero coefficients. Also used in a general sense to convey the concept of redundancy, as quantified by the number of significant coefficients a signal has in a *sparsifying basis*.

Standard basis: The Euclidean basis. In three dimensions, the standard basis vectors are $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$.

Start of the sidelobe region: The angular distance from the main-beam peak to the start of the region where the *radiation pattern* is minimised in an *SLL*-optimisation algorithm.

Steering angle: The intended angular direction in which a beam is pointed during *beamforming*.

Steering matrix: The matrix $\mathbf{A}$ with the *steering vectors* at the sampled angles as columns. The *transform matrix* for an *antenna array*.

Steering vector: The received signal vector $\mathbf{a}(\theta)$ at the *antenna elements* for a reference wave impinging on the *antenna array* from the direction θ.

Subarray: A single *antenna-array* input or output that is a function of multiple *antenna elements*.

Subarray pattern: See *radiation pattern*.

Transform matrix: The matrix $\mathbf{A}$ in a *CS* system that relates the *conventional samples* to the *sparse domain*. Equivalent to the *steering matrix* in *antenna arrays*.

Uniform linear array (ULA): An *antenna array* with *antenna elements* that are uniformly spaced on a line.

Wavefront: The surface on which an electromagnetic wave has constant phase. Assumed to be planar in the far field.

ADDENDUM A DERIVATION OF THE ALGORITHMS

A.1 CBGC DERIVATION

The CBGC algorithm (Chapter 5) is derived below.

Define the functions

$$g_{n,l}(\mathbf{B}) = \left(\left|\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right|^2 - \mu_t^2\right)^p \text{ and} \tag{A.1}$$

$$g(\lambda_n) = \lambda_n\left(\|\mathbf{b}_n\|^2 - 1\right). \tag{A.2}$$

The equilibrium conditions in (5.8) for a given value of m and n now become (all terms not dependent on n fall away)

$$\sum_{n\neq l}\frac{\partial g_{n,l}(\mathbf{B})}{\partial \mathrm{Re}\{b_{m,n}\}} + \frac{\partial g(\lambda_n)}{\partial \mathrm{Re}\{b_{m,n}\}} = 0, \tag{A.3}$$

$$\sum_{n\neq l}\frac{\partial g_{n,l}(\mathbf{B})}{\partial \mathrm{Im}\{b_{m,n}\}} + \frac{\partial g(\lambda_n)}{\partial \mathrm{Im}\{b_{m,n}\}} = 0 \text{ and} \tag{A.4}$$

$$\frac{\partial g(\lambda_n)}{\partial \lambda_n} = 0\,, \tag{A.5}$$

where m and n are fixed, but $l \in (1,\ldots,N)$.

Starting with the real case in (A.3), consider

$$\frac{\partial g_{n,l}(\mathbf{B})}{\partial \mathrm{Re}\{b_{m,n}\}} = p\left(\left|\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right|^2 - \mu_t^2\right)^{p-1} \times \frac{\partial}{\partial \mathrm{Re}\{b_{m,n}\}}\left|\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right|^2 \tag{A.6}$$

$$= p\cdot h_{n,l} \times \frac{\partial}{\partial \mathrm{Re}\{b_{m,n}\}}\left|\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right|^2. \tag{A.7}$$

Then calculate

$$\frac{\partial}{\partial \mathrm{Re}\{b_{m,n}\}}|\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l|^2$$

$$= \frac{\partial}{\partial \mathrm{Re}\{b_{m,n}\}}\left(\left[\mathrm{Re}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\right]^2 + \left[\mathrm{Im}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\right]^2\right) \tag{A.8}$$

$$= 2\mathrm{Re}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \mathrm{Re}\{b_{m,n}\}}\mathrm{Re}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\right) + 2\mathrm{Im}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \mathrm{Re}\{b_{m,n}\}}\mathrm{Im}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\right) \tag{A.9}$$

$$= 2\mathrm{Re}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \mathrm{Re}\{b_{m,n}\}}\mathrm{Re}\left\{\sum_{m'=1}^{M} b_{m',n}^{*}b_{m',l}\right\}\right)$$

$$+ 2\text{Im}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\text{Im}\left\{\sum_{m'=1}^{M} b_{m',n}^{*} b_{m',l}\right\}\right), \tag{A.10}$$

where the sums describe the dot products which perform piece-wise multiplication of the vector elements with indices $m' = \{1, \ldots, m, \ldots, M\}$. However, the derivatives are only functions of the current $m' = m$, and so (A.10) reduces to

$$\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}|\mathbf{b}_n^{\text{H}}\mathbf{b}_l|^2$$

$$= 2\text{Re}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\text{Re}\left\{b_{m,n}^{*} b_{m,l}\right\}\right) + 2\text{Im}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\text{Im}\left\{b_{m,n}^{*} b_{m,l}\right\}\right) \tag{A.11}$$

$$= 2\text{Re}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\text{Re}\Big[\left(\text{Re}\left\{b_{m,n}\right\} - \text{j}\text{Im}\left\{b_{m,n}\right\}\right)\left(\text{Re}\left\{b_{m,l}\right\} + \text{j}\text{Im}\left\{b_{m,l}\right\}\right)\Big]\right)$$

$$+ 2\text{Im}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\text{Im}\Big[\left(\text{Re}\left\{b_{m,n}\right\} - \text{j}\text{Im}\left\{b_{m,n}\right\}\right)\left(\text{Re}\left\{b_{m,l}\right\} + \text{j}\text{Im}\left\{b_{m,l}\right\}\right)\Big]\right) \tag{A.12}$$

$$= 2\text{Re}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\text{Re}\Big[\text{Re}\left\{b_{m,n}\right\}\text{Re}\left\{b_{m,l}\right\} + \text{Im}\left\{b_{m,n}\right\}\text{Im}\left\{b_{m,l}\right\}$$

$$+ \text{j}\left(\text{Re}\left\{b_{m,n}\right\}\text{Im}\left\{b_{m,l}\right\} - \text{Im}\left\{b_{m,n}\right\}\text{Re}\left\{b_{m,l}\right\}\right)\Big]\right)$$

$$+ 2\text{Im}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\text{Im}\Big[\text{Re}\left\{b_{m,n}\right\}\text{Re}\left\{b_{m,l}\right\} + \text{Im}\left\{b_{m,n}\right\}\text{Im}\left\{b_{m,l}\right\}$$

$$+ \text{j}\left(\text{Re}\left\{b_{m,n}\right\}\text{Im}\left\{b_{m,l}\right\} - \text{Im}\left\{b_{m,n}\right\}\text{Re}\left\{b_{m,l}\right\}\right)\Big]\right) \tag{A.13}$$

$$= 2\text{Re}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\Big[\text{Re}\left\{b_{m,n}\right\}\text{Re}\left\{b_{m,l}\right\} + \text{Im}\left\{b_{m,n}\right\}\text{Im}\left\{b_{m,l}\right\}\Big]\right)$$

$$+ 2\text{Im}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\left(\frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\Big[\text{Re}\left\{b_{m,n}\right\}\text{Im}\left\{b_{m,l}\right\} - \text{Im}\left\{b_{m,n}\right\}\text{Re}\left\{b_{m,l}\right\}\Big]\right) \tag{A.14}$$

$$= 2\Big(\text{Re}\left\{b_{m,l}\right\}\text{Re}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\} + \text{Im}\left\{b_{m,l}\right\}\text{Im}\left\{\mathbf{b}_n^{\text{H}}\mathbf{b}_l\right\}\Big) \tag{A.15}$$

$$= r_{m,n,l}\,. \tag{A.16}$$

Next, consider

$$\frac{\partial g(\lambda_n)}{\partial \text{Re}\{b_{m,n}\}} = \frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\lambda_n\left(\|\mathbf{b}_n\|^2 - 1\right) \tag{A.17}$$

$$= \frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\lambda_n\left(\sum_{m'=1}^{M}\text{Re}^2\left\{b_{m',n}\right\} + \text{Im}^2\left\{b_{m',n}\right\}\right) \tag{A.18}$$

$$= \frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\lambda_n\left(\text{Re}^2\left\{b_{m,n}\right\} + \text{Im}^2\left\{b_{m,n}\right\}\right) \tag{A.19}$$

$$= \frac{\partial}{\partial \text{Re}\{b_{m,n}\}}\lambda_n\text{Re}^2\left\{b_{m,n}\right\} \tag{A.20}$$

$$= 2\lambda_n\text{Re}\{b_{m,n}\}\,. \tag{A.21}$$

Combining (A.3), (A.7), (A.16) and (A.21) gives

$$0 = \sum_{n \neq l} \frac{\partial g_{n,l}(\mathbf{B})}{\partial \mathrm{Re}\{b_{m,n}\}} + \frac{\partial g(\lambda_n)}{\partial \mathrm{Re}\{b_{m,n}\}} \tag{A.22}$$

$$= \sum_{n \neq l} p \cdot h_{n,l} \cdot r_{m,n,l} + 2\lambda_n \mathrm{Re}\{b_{m,n}\} \tag{A.23}$$

$$\therefore \mathrm{Re}\{b_{m,n}\} = \frac{p}{2\lambda_n} \left[- \sum_{n \neq l} h_{n,l} \cdot r_{m,n,l} \right] = \frac{p}{2\lambda_n} u_{m,n} \,. \tag{A.24}$$

The imaginary case in (A.4) is derived using similar logic. Consider

$$\frac{\partial g_{n,l}(\mathbf{B})}{\partial \mathrm{Im}\{b_{m,n}\}} = p \left(\left| \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right|^2 - \mu_t^2 \right)^{p-1} \times \frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \left| \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right|^2 \tag{A.25}$$

$$= p \cdot h_{n,l} \times \frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \left| \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right|^2 . \tag{A.26}$$

Then calculate

$$\frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} |\mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l|^2$$

$$= \frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \left(\left[\mathrm{Re}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \right]^2 + \left[\mathrm{Im}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \right]^2 \right) \tag{A.27}$$

$$= 2\mathrm{Re}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \left(\frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \mathrm{Re}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \right) + 2\mathrm{Im}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \left(\frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \mathrm{Im}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \right) \tag{A.28}$$

$$= 2\mathrm{Re}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \left(\frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \left[\mathrm{Re}\left\{ b_{m,n} \right\} \mathrm{Re}\left\{ b_{m,l} \right\} + \mathrm{Im}\left\{ b_{m,n} \right\} \mathrm{Im}\left\{ b_{m,l} \right\} \right] \right)$$

$$+ 2\mathrm{Im}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \left(\frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \left[\mathrm{Re}\left\{ b_{m,n} \right\} \mathrm{Im}\left\{ b_{m,l} \right\} - \mathrm{Im}\left\{ b_{m,n} \right\} \mathrm{Re}\left\{ b_{m,l} \right\} \right] \right) \tag{A.29}$$

$$= 2\left(\mathrm{Im}\left\{ b_{m,l} \right\} \mathrm{Re}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} - \mathrm{Re}\left\{ b_{m,l} \right\} \mathrm{Im}\left\{ \mathbf{b}_n^{\mathrm{H}} \mathbf{b}_l \right\} \right) \tag{A.30}$$

$$= s_{m,n,l} \,. \tag{A.31}$$

Next, consider

$$\frac{\partial g(\lambda_n)}{\partial \mathrm{Im}\{b_{m,n}\}} = \frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \lambda_n \left(\|\mathbf{b}_n\|^2 - 1 \right) \tag{A.32}$$

$$= \frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \lambda_n \left(\mathrm{Re}^2\left\{ b_{m,n} \right\} + \mathrm{Im}^2\left\{ b_{m,n} \right\} \right) \tag{A.33}$$

$$= \frac{\partial}{\partial \mathrm{Im}\{b_{m,n}\}} \lambda_n \mathrm{Im}^2\left\{ b_{m,n} \right\} \tag{A.34}$$

$$= 2\lambda_n \mathrm{Im}\{b_{m,n}\} \,. \tag{A.35}$$

Combining (A.4), (A.26), (A.31) and (A.35) gives

$$0 = \sum_{n \neq l} \frac{\partial g_{n,l}(\mathbf{B})}{\partial \mathrm{Im}\{b_{m,n}\}} + \frac{\partial g(\lambda_n)}{\partial \mathrm{Im}\{b_{m,n}\}} \tag{A.36}$$

$$= \sum_{n \neq l} p \cdot h_{n,l} \cdot s_{m,n,l} + 2\lambda_n \mathrm{Im}\{b_{m,n}\} \tag{A.37}$$

$$\therefore \mathrm{Im}\{b_{m,n}\} = \frac{p}{2\lambda_n} \left[- \sum_{n \neq l} h_{n,l} \cdot s_{m,n,l} \right] = \frac{p}{2\lambda_n} v_{m,n} \,. \tag{A.38}$$

Combining (A.5), (A.24) and (A.38) gives

$$0 = \frac{\partial g(\lambda_n)}{\partial \lambda_n} = \frac{\partial}{\partial \lambda_n} \lambda_n \left(\|\mathbf{b}_n\|^2 - 1 \right) = \|\mathbf{b}_n\|^2 - 1 \tag{A.39}$$

$$\therefore 1 = \|\mathbf{b}_n\|^2 \tag{A.40}$$

$$= \sum_{m'=1}^{M} \mathrm{Re}^2\{b_{m',n}\} + \mathrm{Im}^2\{b_{m',n}\} \tag{A.41}$$

$$= \sum_{m'=1}^{M} \left(\frac{p}{2\lambda_n} u_{m',n} \right)^2 + \left(\frac{p}{2\lambda_n} v_{m',n} \right)^2 \tag{A.42}$$

$$= \left(\frac{p}{2\lambda_n} \right)^2 \sum_{m'=1}^{M} u_{m',n}^2 + v_{m',n}^2 \tag{A.43}$$

$$\therefore \frac{2\lambda_n}{p} = \sqrt{\sum_{m'=1}^{M} u_{m',n}^2 + v_{m',n}^2} \tag{A.44}$$

$$\therefore 2\lambda_n = p \sqrt{\sum_{m'=1}^{M} u_{m',n}^2 + v_{m',n}^2} \tag{A.45}$$

$$= p \left\| \mathbf{u}_n + \mathrm{j}\mathbf{v}_n \right\| \,. \tag{A.46}$$

The elements $b_{m,n}$ of the codeword $\mathbf{b}_n$, taken as the force vector, is found by combining (A.24), (A.38) and (A.46) to give

$$b_{m,n} = \mathrm{Re}\{b_{m,n}\} + \mathrm{j}\mathrm{Im}\{b_{m,n}\} \tag{A.47}$$

$$= \frac{p}{2\lambda_n} u_{m,n} + \mathrm{j} \frac{p}{2\lambda_n} v_{m,n} \tag{A.48}$$

$$= \frac{p}{2\lambda_n} \left(u_{m,n} + \mathrm{j} v_{m,n} \right) \tag{A.49}$$

$$= \frac{p}{p \left\| \mathbf{u}_n + \mathrm{j}\mathbf{v}_n \right\|} \left(u_{m,n} + \mathrm{j} v_{m,n} \right) \tag{A.50}$$

$$= \frac{u_{m,n} + \mathrm{j} v_{m,n}}{\left\| \mathbf{u}_n + \mathrm{j}\mathbf{v}_n \right\|} . \tag{A.51}$$

Combining the elements $b_{m',n}$ for $m' \in \{1, \ldots, M\}$ gives

$$\mathbf{f}_n \equiv \mathbf{b}_n = \frac{\mathbf{u}_n + \mathrm{j}\mathbf{v}_n}{\left\| \mathbf{u}_n + \mathrm{j}\mathbf{v}_n \right\|} = \underline{\mathbf{u}_n + \mathrm{j}\mathbf{v}_n}, \tag{A.52}$$

where underlining denotes normalisation (Section 5.2). The relationship $\mathbf{f}_n = \mathbf{b}_n$ only holds true at equilibrium, since it was derived using the equilibrium conditions in (A.3)–(A.5). However, when equilibrium is not met, this force vector will provide an estimate for the direction in which the nearest stationary point lies.

The next step is to combine the real and imaginary parts of (A.52), which can be expanded into

$$\mathbf{f}_n = \underline{\mathbf{u}_n + \mathrm{j}\mathbf{v}_n} \tag{A.53}$$

$$= \underline{-\sum_{n\neq l} h_{n,l} \cdot \mathbf{r}_{n,l} - \mathrm{j}\sum_{n\neq l} h_{n,l} \cdot \mathbf{s}_{n,l}} \tag{A.54}$$

$$= \underline{-\sum_{n\neq l} h_{n,l} \cdot \left(\mathbf{r}_{n,l} + \mathrm{j}\mathbf{s}_{n,l}\right)} \tag{A.55}$$

$$= \underline{-2\sum_{n\neq l} h_{n,l} \cdot \Big[\mathrm{Re}\left\{\mathbf{b}_l\right\}\mathrm{Re}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\} + \mathrm{Im}\left\{\mathbf{b}_l\right\}\mathrm{Im}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}} \\ \underline{+\mathrm{j}\Big(\mathrm{Im}\left\{\mathbf{b}_l\right\}\mathrm{Re}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\} - \mathrm{Re}\left\{\mathbf{b}_l\right\}\mathrm{Im}\left\{\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right\}\Big)\Big]}. \tag{A.56}$$

In order to combine the real and imaginary parts of (A.56), consider the identity

$$ab^* = \Big(\mathrm{Re}\{a\} + \mathrm{jIm}\{a\}\Big)\Big(\mathrm{Re}\{b\} + \mathrm{jIm}\{b\}\Big)^* \tag{A.57}$$

$$= \Big(\mathrm{Re}\{a\} + \mathrm{jIm}\{a\}\Big)\Big(\mathrm{Re}\{b\} - \mathrm{jIm}\{b\}\Big) \tag{A.58}$$

$$= \mathrm{Re}\{a\}\mathrm{Re}\{b\} + \mathrm{Im}\{a\}\mathrm{Im}\{b\} + \mathrm{j}\Big[\mathrm{Im}\{a\}\mathrm{Re}\{b\} - \mathrm{Re}\{a\}\mathrm{Im}\{b\}\Big]. \tag{A.59}$$

Comparing (A.56) to (A.59) allows (A.56) to be re-written as

$$\mathbf{f}_n = \underline{-2\sum_{n\neq l} h_{n,l} \cdot \mathbf{b}_l \left(\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right)^*} \tag{A.60}$$

$$= \underline{-2\sum_{n\neq l} h_{n,l} \cdot \mathbf{b}_l \left(\mathbf{b}_l^{\mathrm{H}}\mathbf{b}_n\right)} \tag{A.61}$$

$$= \underline{-2\sum_{n\neq l} \left(\left|\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right|^2 - \mu_t^2\right)^{p-1} \cdot \mathbf{b}_l \left(\mathbf{b}_l^{\mathrm{H}}\mathbf{b}_n\right)} \tag{A.62}$$

$$= \underline{-2\sum_{n\neq l} \left(\left|\mathbf{b}_n^{\mathrm{H}}\mathbf{b}_l\right|^2 - \mu_t^2\right)^{p-1} \left(\mathbf{b}_l^{\mathrm{H}}\mathbf{b}_n\right) \cdot \mathbf{b}_l}. \tag{A.63}$$

A.2 COMPRESSIVE ARRAY DESIGN ALGORITHM GRADIENT DERIVATIONS

The gradients of the goal and constraint functions in the compressive array design algorithm (Chapter 6) are derived below.

The parameters of the optimisation problem are the elements in $\mathbf{\Phi}$, and therefore the derivatives of the goal and constraint functions (6.8)–(6.10) and (6.14) need to be calculated with respect to $\mathbf{\Phi}$.

Start by considering the derivative of (6.8) with respect to each element in $\mathbf{\Phi}$ as

$$\frac{\partial g(\mathbf{\Phi})}{\partial \phi_{m,n}} = \frac{\partial}{\partial \phi_{m,n}} \alpha \left[\frac{1}{C} \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p} \tag{A.64}$$

$$= \frac{\alpha}{C^{1/p}} \frac{\partial}{\partial \phi_{m,n}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p} \tag{A.65}$$

$$= \frac{\alpha}{pC^{1/p}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p-1} \times \frac{\partial}{\partial \phi_{m,n}} \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \tag{A.66}$$

$$= \frac{\alpha}{pC^{1/p}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p-1} \times \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \frac{\partial}{\partial \phi_{m,n}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \tag{A.67}$$

$$= \frac{\alpha}{pC^{1/p}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p-1} \times \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \frac{\partial}{\partial \phi_{m,n}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|^2}{\alpha^2 R_{s,a}^2} \right)^{p/2} \tag{A.68}$$

$$= \frac{p\alpha}{2pC^{1/p}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p-1} \times \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left[\left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|^2}{\alpha^2 R_{s,a}^2} \right)^{p/2-1} \frac{\partial}{\partial \phi_{m,n}} \frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|^2}{\alpha^2 R_{s,a}^2} \right] \tag{A.69}$$

$$= \frac{\alpha}{2\alpha^2 C^{1/p}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p-1} \times \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left[\frac{1}{R_{s,a}^2} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|^2}{\alpha^2 R_{s,a}^2} \right)^{\frac{p-2}{2}} \frac{\partial}{\partial \phi_{m,n}} \left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|^2 \right] \tag{A.70}$$

$$= \frac{1}{2\alpha C^{1/p}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p-1} \times \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left[\frac{1}{R_{s,a}^2} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^{p-2} \frac{\partial}{\partial \phi_{m,n}} \left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|^2 \right]. \tag{A.71}$$

Wherever $\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right| / (\alpha R_{s,a})$ is raised to some function of p, α is not factored out since its function is to prevent the power from underflowing (Section 6.3.2).

Next, find

$$\frac{\partial}{\partial \phi_{m,n}} \left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|^2 = \frac{\partial}{\partial \phi_{m,n}} \left(\text{Re}^2 \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} + \text{Im}^2 \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} \right) \tag{A.72}$$

$$= 2\text{Re} \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} \left(\frac{\partial}{\partial \phi_{m,n}} \text{Re} \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} \right) + 2\text{Im} \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} \left(\frac{\partial}{\partial \phi_{m,n}} \text{Im} \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} \right) \tag{A.73}$$

$$= 2\text{Re} \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} \left(\frac{\partial}{\partial \text{Re} \left\{ \phi_{m,n} \right\}} \text{Re} \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} + \text{j} \frac{\partial}{\partial \text{Im} \left\{ \phi_{m,n} \right\}} \text{Re} \left\{ \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right\} \right)$$

$$+ 2\text{Im}\left\{\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\}\left(\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\text{Im}\left\{\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\} + \text{j}\frac{\partial}{\partial \text{Im}\left\{\phi_{m,n}\right\}}\text{Im}\left\{\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\}\right) \tag{A.74}$$

$$= 2\text{Re}\left\{\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\}\left(\text{Re}\left\{\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\} + \text{jRe}\left\{\frac{\partial}{\partial \text{Im}\left\{\phi_{m,n}\right\}}\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\}\right)$$

$$+ 2\text{Im}\left\{\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\}\left(\text{Im}\left\{\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\} + \text{jIm}\left\{\frac{\partial}{\partial \text{Im}\left\{\phi_{m,n}\right\}}\mathbf{b}_s^{\text{H}}\mathbf{b}_a\right\}\right). \tag{A.75}$$

Then, calculate

$$\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\mathbf{b}_s^{\text{H}}\mathbf{b}_a = \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\sum_{m'=1}^{M} b_{m',s}^{*} b_{m',a}, \tag{A.76}$$

where $b_{m,s}$ and $b_{m,a}$ refer to the mth elements in $\mathbf{b}_s$ and $\mathbf{b}_a$, respectively, and $m' = \{1, \ldots, m, \ldots, M\}$. However, the derivative is only a function of the current $m' = m$, leading to

$$\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\mathbf{b}_s^{\text{H}}\mathbf{b}_a = \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}} b_{m,s}^{*} b_{m,a}. \tag{A.77}$$

Applying the product rule gives

$$\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\mathbf{b}_s^{\text{H}}\mathbf{b}_a = \left(\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}} b_{m,s}^{*}\right) b_{m,a} + b_{m,s}^{*}\left(\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}} b_{m,a}\right). \tag{A.78}$$

The differentiated elements in $\mathbf{b}$ now need to be expanded using $\mathbf{b} = \boldsymbol{\Phi}\mathbf{a}$, starting with

$$\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}} b_{m,s}^{*} = \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\left(\boldsymbol{\phi}_m \mathbf{a}_s\right)^{*} \tag{A.79}$$

$$= \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\sum_{n'=1}^{N} \phi_{m,n'}^{*} a_{n',s}^{*} \tag{A.80}$$

$$= \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\phi_{m,n}^{*} a_{n,s}^{*} \tag{A.81}$$

$$= \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\left(\text{Re}\left\{\phi_{m,n}\right\} - \text{jIm}\left\{\phi_{m,n}\right\}\right) a_{n,s}^{*} \tag{A.82}$$

$$= a_{n,s}^{*}, \tag{A.83}$$

followed by

$$\frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}} b_{m,a} = \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\left(\boldsymbol{\phi}_m \mathbf{a}_a\right) \tag{A.84}$$

$$= \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\sum_{n'=1}^{N} \phi_{m,n'} a_{n',a} \tag{A.85}$$

$$= \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\phi_{m,n} a_{n,a} \tag{A.86}$$

$$= \frac{\partial}{\partial \text{Re}\left\{\phi_{m,n}\right\}}\left(\text{Re}\left\{\phi_{m,n}\right\} + \text{jIm}\left\{\phi_{m,n}\right\}\right) a_{n,a} \tag{A.87}$$

$$= a_{n,a}. \tag{A.88}$$

Substituting (A.83) and (A.88) into (A.78) gives

$$\frac{\partial}{\partial \mathrm{Re}\left\{\phi_{m,n}\right\}} \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a = a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a}. \tag{A.89}$$

Now, consider the derivative of $\mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a$ with respect to the imaginary part of $\phi_{m,n}$ as

$$\frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a = \frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \sum_{m'=1}^{M} b_{m',s}^* b_{m',a} \tag{A.90}$$

$$= \frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} b_{m,s}^* b_{m,a}. \tag{A.91}$$

Applying the product rule as in the real case gives

$$\frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a = \left(\frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} b_{m,s}^* \right) b_{m,a} + b_{m,s}^* \left(\frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} b_{m,a} \right). \tag{A.92}$$

Next, consider the differentiated element in $\mathbf{b}_s$:

$$\frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} b_{m,s}^* = \frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \phi_{m,n}^* a_{n,s}^* \tag{A.93}$$

$$= \frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \left(\mathrm{Re}\left\{\phi_{m,n}\right\} - \mathrm{j}\mathrm{Im}\left\{\phi_{m,n}\right\} \right) a_{n,s}^* \tag{A.94}$$

$$= -\mathrm{j} a_{n,s}^*, \tag{A.95}$$

followed by the element in $\mathbf{b}_a$:

$$\frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} b_{m,a} = \frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \phi_{m,n} a_{n,a} \tag{A.96}$$

$$= \frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \left(\mathrm{Re}\left\{\phi_{m,n}\right\} + \mathrm{j}\mathrm{Im}\left\{\phi_{m,n}\right\} \right) a_{n,a} \tag{A.97}$$

$$= \mathrm{j} a_{n,a}. \tag{A.98}$$

Substituting (A.95) and (A.98) into (A.92) gives

$$\frac{\partial}{\partial \mathrm{Im}\left\{\phi_{m,n}\right\}} \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a = -\mathrm{j} a_{n,s}^* b_{m,a} + \mathrm{j} b_{m,s}^* a_{n,a}. \tag{A.99}$$

Substituting (A.89) and (A.99) into (A.75) gives

$$\frac{\partial}{\partial \phi_{m,n}} \left| \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a \right|^2 = 2\mathrm{Re}\left\{ \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a \right\} \left(\mathrm{Re}\left\{ a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a} \right\} + \mathrm{j}\mathrm{Re}\left\{ -\mathrm{j} a_{n,s}^* b_{m,a} + \mathrm{j} b_{m,s}^* a_{n,a} \right\} \right)$$
$$+ 2\mathrm{Im}\left\{ \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a \right\} \left(\mathrm{Im}\left\{ a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a} \right\} + \mathrm{j}\mathrm{Im}\left\{ -\mathrm{j} a_{n,s}^* b_{m,a} + \mathrm{j} b_{m,s}^* a_{n,a} \right\} \right) \tag{A.100}$$

$$= 2\mathrm{Re}\left\{ \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a \right\} \left(\mathrm{Re}\left\{ a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a} \right\} + \mathrm{j}\mathrm{Im}\left\{ a_{n,s}^* b_{m,a} - b_{m,s}^* a_{n,a} \right\} \right)$$
$$+ 2\mathrm{Im}\left\{ \mathbf{b}_s^{\mathrm{H}} \mathbf{b}_a \right\} \left(\mathrm{Im}\left\{ a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a} \right\} + \mathrm{j}\mathrm{Re}\left\{ -a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a} \right\} \right). \tag{A.101}$$

In order to recombine the real and imaginary parts, start by rearranging (A.101) as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \phi_{m,n}} \left|\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right|^2 \\
&= 2\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a}\right\} + 2\mathrm{jRe}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a} - b_{m,s}^* a_{n,a}\right\} \\
&\quad + 2\mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a}\right\} + 2\mathrm{jIm}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{-a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a}\right\} && \text{(A.102)} \\
&= 2\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a}\right\} + 2\mathrm{jRe}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a} - b_{m,s}^* a_{n,a}\right\} \\
&\quad + 2\mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a}\right\} + 2\mathrm{jIm}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{-a_{n,s}^* b_{m,a} + b_{m,s}^* a_{n,a}\right\} && \text{(A.103)} \\
&= 2\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{a_{n,s}^* b_{m,a}\right\} + 2\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{b_{m,s}^* a_{n,a}\right\} \\
&\quad + 2\mathrm{jRe}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a}\right\} - 2\mathrm{jRe}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{b_{m,s}^* a_{n,a}\right\} \\
&\quad + 2\mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a}\right\} + 2\mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{b_{m,s}^* a_{n,a}\right\} \\
&\quad - 2\mathrm{jIm}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{a_{n,s}^* b_{m,a}\right\} + 2\mathrm{jIm}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{b_{m,s}^* a_{n,a}\right\} && \text{(A.104)} \\
&= 2\Big[\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{a_{n,s}^* b_{m,a}\right\} + \mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a}\right\} \\
&\qquad\qquad + \mathrm{j}\left(\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{a_{n,s}^* b_{m,a}\right\} - \mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{a_{n,s}^* b_{m,a}\right\}\right)\Big] \\
&\quad + 2\Big[\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{b_{m,s}^* a_{n,a}\right\} + \mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{b_{m,s}^* a_{n,a}\right\} \\
&\qquad\qquad + \mathrm{j}\left(-\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Im}\left\{b_{m,s}^* a_{n,a}\right\} + \mathrm{Im}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\mathrm{Re}\left\{b_{m,s}^* a_{n,a}\right\}\right)\Big] && \text{(A.105)} \\
&= 2\left[(ac + bd) + \mathrm{j}(ad - bc)\right] + 2\left[(eg + fh) + \mathrm{j}(-eh + fg)\right]. && \text{(A.106)}
\end{aligned}$$

Consider the identities

$$(a + \mathrm{j}b)^*(c + \mathrm{j}d) = (ac + bd) + \mathrm{j}(ad - bc) \tag{A.107}$$

and

$$(e + \mathrm{j}f)(g + \mathrm{j}h)^* = (eg + fh) + \mathrm{j}(-eh + fg), \tag{A.108}$$

which allow (A.106) to be rewritten as

$$\begin{aligned}
\frac{\partial}{\partial \phi_{m,n}} \left|\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right|^2 &= 2\left(\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\} + \mathrm{jIm}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\right)^* \left(\mathrm{Re}\left\{a_{n,s}^* b_{m,a}\right\} + \mathrm{jIm}\left\{a_{n,s}^* b_{m,a}\right\}\right) \\
&\quad + 2\left(\mathrm{Re}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\} + \mathrm{jIm}\left\{\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right\}\right)\left(\mathrm{Re}\left\{b_{m,s}^* a_{n,a}\right\} + \mathrm{jIm}\left\{b_{m,s}^* a_{n,a}\right\}\right)^* && \text{(A.109)} \\
&= 2\left(\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right)^*\left(a_{n,s}^* b_{m,a}\right) + 2\left(\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right)\left(b_{m,s}^* a_{n,a}\right)^* && \text{(A.110)} \\
&= 2\left(\mathbf{b}_a^{\mathrm{H}}\mathbf{b}_s\right)\left(b_{m,a} a_{n,s}^*\right) + 2\left(\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right)\left(b_{m,s} a_{n,a}^*\right). && \text{(A.111)}
\end{aligned}$$

Now, the derivative of $\left|\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right|^2$ with respect to the matrix $\mathbf{\Phi}$ can be written as

$$\frac{\partial}{\partial \mathbf{\Phi}} \left|\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right|^2 = 2\left[\left(\mathbf{b}_a^{\mathrm{H}}\mathbf{b}_s\right)\mathbf{b}_a\mathbf{a}_s^{\mathrm{H}} + \left(\mathbf{b}_s^{\mathrm{H}}\mathbf{b}_a\right)\mathbf{b}_s\mathbf{a}_a^{\mathrm{H}}\right], \tag{A.112}$$

which, together with (A.71), gives the derivative of the goal function with respect to $\boldsymbol{\Phi}$ as

$$\frac{\partial g(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}} = \frac{1}{\alpha C^{1/p}} \left[\sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^p \right]^{1/p-1} \times \sum_{\substack{\theta_a - \theta_s \geq \\ \theta_{\text{SLL}}}} \left(\frac{1}{R_{s,a}^2} \left(\frac{\left| \mathbf{b}_s^{\text{H}} \mathbf{b}_a \right|}{\alpha R_{s,a}} \right)^{p-2} \left[\left(\mathbf{b}_a^{\text{H}} \mathbf{b}_s \right) \mathbf{b}_a \mathbf{a}_s^{\text{H}} + \left(\mathbf{b}_s^{\text{H}} \mathbf{b}_a \right) \mathbf{b}_s \mathbf{a}_a^{\text{H}} \right] \right). \tag{A.113}$$

To find the gradients of the unit-norm constraint function (6.9), start by calculating

$$\frac{\partial}{\partial \phi_{m,n}} c_1(\boldsymbol{\Phi}) = \frac{\partial}{\partial \phi_{m,n}} \|\mathbf{b}_s\|^2 \tag{A.114}$$

$$= \frac{\partial}{\partial \phi_{m,n}} \sum_{m'=1}^{M} \text{Re}^2\{b_{m',s}\} + \text{Im}^2\{b_{m',s}\} \tag{A.115}$$

$$= \frac{\partial}{\partial \phi_{m,n}} \text{Re}^2\{b_{m,s}\} + \text{Im}^2\{b_{m,s}\} \tag{A.116}$$

$$= 2\text{Re}\{b_{m,s}\} \frac{\partial}{\partial \phi_{m,n}} \text{Re}\{b_{m,s}\} + 2\text{Im}\{b_{m,s}\} \frac{\partial}{\partial \phi_{m,n}} \text{Im}\{b_{m,s}\} \tag{A.117}$$

$$= 2\text{Re}\{b_{m,s}\} \left(\frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} \text{Re}\{b_{m,s}\} + \text{j} \frac{\partial}{\partial \text{Im}\{\phi_{m,n}\}} \text{Re}\{b_{m,s}\} \right) + 2\text{Im}\{b_{m,s}\} \left(\frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} \text{Im}\{b_{m,s}\} + \text{j} \frac{\partial}{\partial \text{Im}\{\phi_{m,n}\}} \text{Im}\{b_{m,s}\} \right) \tag{A.118}$$

$$= 2\text{Re}\{b_{m,s}\} \left(\text{Re}\left\{ \frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} b_{m,s} \right\} + \text{jRe}\left\{ \frac{\partial}{\partial \text{Im}\{\phi_{m,n}\}} b_{m,s} \right\} \right) + 2\text{Im}\{b_{m,s}\} \left(\text{Im}\left\{ \frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} b_{m,s} \right\} + \text{jIm}\left\{ \frac{\partial}{\partial \text{Im}\{\phi_{m,n}\}} b_{m,s} \right\} \right). \tag{A.119}$$

Next, find

$$\frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} b_{m,s} = \frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} \boldsymbol{\phi}_m \mathbf{a}_s \tag{A.120}$$

$$= \frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} \sum_{n'=1}^{N} \phi_{m,n'} a_{n',s} \tag{A.121}$$

$$= \frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} \phi_{m,n} a_{n,s} \tag{A.122}$$

$$= \frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} \left(\text{Re}\{\phi_{m,n}\} + \text{jIm}\{\phi_{m,n}\} \right) \left(\text{Re}\{a_{n,s}\} + \text{jIm}\{a_{n,s}\} \right) \tag{A.123}$$

$$= \frac{\partial}{\partial \text{Re}\{\phi_{m,n}\}} \text{Re}\{\phi_{m,n}\} \text{Re}\{a_{n,s}\} - \text{Im}\{\phi_{m,n}\} \text{Im}\{a_{n,s}\} + \text{j} \left(\text{Re}\{\phi_{m,n}\} \text{Im}\{a_{n,s}\} + \text{Im}\{\phi_{m,n}\} \text{Re}\{a_{n,s}\} \right) \tag{A.124}$$

$$= \mathrm{Re}\{a_{n,s}\} + \mathrm{jIm}\{a_{n,s}\} \tag{A.125}$$
$$= a_{n,s}, \tag{A.126}$$

followed by

$$\frac{\partial}{\partial \mathrm{Im}\{\phi_{m,n}\}} b_{m,s} = \frac{\partial}{\partial \mathrm{Im}\{\phi_{m,n}\}} \boldsymbol{\phi}_m \mathbf{a}_s \tag{A.127}$$
$$= \frac{\partial}{\partial \mathrm{Im}\{\phi_{m,n}\}} \sum_{n'=1}^{N} \phi_{m,n'} a_{n',s} \tag{A.128}$$
$$= \frac{\partial}{\partial \mathrm{Im}\{\phi_{m,n}\}} \phi_{m,n} a_{n,s} \tag{A.129}$$
$$= \frac{\partial}{\partial \mathrm{Im}\{\phi_{m,n}\}} \left(\mathrm{Re}\{\phi_{m,n}\} + \mathrm{jIm}\{\phi_{m,n}\}\right) \left(\mathrm{Re}\{a_{n,s}\} + \mathrm{jIm}\{a_{n,s}\}\right) \tag{A.130}$$
$$= \frac{\partial}{\partial \mathrm{Im}\{\phi_{m,n}\}} \mathrm{Re}\{\phi_{m,n}\}\,\mathrm{Re}\{a_{n,s}\} - \mathrm{Im}\{\phi_{m,n}\}\,\mathrm{Im}\{a_{n,s}\} + \mathrm{j}\left(\mathrm{Re}\{\phi_{m,n}\}\,\mathrm{Im}\{a_{n,s}\} + \mathrm{Im}\{\phi_{m,n}\}\,\mathrm{Re}\{a_{n,s}\}\right) \tag{A.131}$$
$$= -\mathrm{Im}\{a_{n,s}\} + \mathrm{jRe}\{a_{n,s}\} \tag{A.132}$$
$$= \mathrm{j}a_{n,s}. \tag{A.133}$$

Substituting (A.126) and (A.133) into (A.119) gives

$$\frac{\partial}{\partial \phi_{m,n}} c_1(\boldsymbol{\Phi}) = 2\mathrm{Re}\{b_{m,s}\} \left(\mathrm{Re}\{a_{n,s}\} + \mathrm{jRe}\{\mathrm{j}a_{n,s}\}\right) + 2\mathrm{Im}\{b_{m,s}\} \left(\mathrm{Im}\{a_{n,s}\} + \mathrm{jIm}\{\mathrm{j}a_{n,s}\}\right) \tag{A.134}$$
$$= 2\mathrm{Re}\{b_{m,s}\} \left(\mathrm{Re}\{a_{n,s}\} - \mathrm{jIm}\{a_{n,s}\}\right) + 2\mathrm{Im}\{b_{m,s}\} \left(\mathrm{Im}\{a_{n,s}\} + \mathrm{jRe}\{a_{n,s}\}\right) \tag{A.135}$$
$$= 2\left[\mathrm{Re}\{b_{m,s}\}\,\mathrm{Re}\{a_{n,s}\} + \mathrm{Im}\{b_{m,s}\}\,\mathrm{Im}\{a_{n,s}\}\right] + 2\mathrm{j}\left[-\mathrm{Re}\{b_{m,s}\}\,\mathrm{Im}\{a_{n,s}\} + \mathrm{Im}\{b_{m,s}\}\,\mathrm{Re}\{a_{n,s}\}\right] \tag{A.136}$$
$$= 2\left[\left(\mathrm{Re}\{b_{m,s}\} + \mathrm{jIm}\{b_{m,s}\}\right) \left(\mathrm{Re}\{a_{n,s}\} + \mathrm{jIm}\{a_{n,s}\}\right)^*\right] \tag{A.137}$$
$$= 2b_{m,s}a^*_{n,s}. \tag{A.138}$$

From (A.138), the gradient of the constraint function (6.9) with respect to the matrix $\boldsymbol{\Phi}$ is found as

$$\frac{\partial}{\partial \boldsymbol{\Phi}} c_2(\boldsymbol{\Phi}) = 2\mathbf{b}_s \mathbf{a}_s^{\mathrm{H}}. \tag{A.139}$$

Similarly, the gradient of (6.9) is found as

$$\frac{\partial}{\partial \boldsymbol{\Phi}} c_2(\boldsymbol{\Phi}) = -2\mathbf{b}_s \mathbf{a}_s^{\mathrm{H}}. \tag{A.140}$$

The gradient of the hard null constraint function (6.14) is found using a similar approach to be

$$\frac{\partial c_{3,m}(\boldsymbol{\phi}_m)}{\partial \boldsymbol{\phi}_m} = 2b_m(\theta_w)\mathbf{a}(\theta_w)^{\mathrm{H}} \tag{A.141}$$

for row m in $\boldsymbol{\Phi}$; the gradients at all other rows are zero.