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

NRL/5520/FR--2024/1

Optimizing Combined Transmit-and-Receive Coherent Distributed Arrays

Michael V. Lipski

Networks and Communication Systems Branch
Information Technology Division

May 2, 2024

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
May 2, 2024	NRL Formal Report	START DATE 07/07/2020	END DATE 07/07/2022

4. TITLE AND SUBTITLE
Optimizing Combined Transmit-and-Receive Coherent Distributed Arrays

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
		NISE
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
55-N2Y1-01		N2Y1

6. AUTHOR(S)
Michael V. Lipski

7. PERFORMING ORGANIZATION / AFFILIATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
US Naval Research Laboratory 4555 Overlook Ave SW Washington, DC 20375-5320	NRL/5520/FR--2024/1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR / MONITOR'S ACRONYM(S)	11. SPONSOR / MONITOR'S REPORT NUMBER(S)
US Naval Research Laboratory 4555 Overlook Ave SW Washington, DC 20375-5320	NRL-NISE	

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

13. SUPPLEMENTAL NOTES
Karles Fellowship

14. ABSTRACT
In the presented work, a model for a combined transmit-and-receive coherent distributed array system is considered. The system model describes a pair of distributed arrays, a transmit array and a receive array, that are defined as aperiodic volumetric arrays with cubic array volumes. Total gain for a combined transmit-and-receive coherent distributed array is labeled coherent communication gain (CCG), and a general expression for CCG is given. The general expression can accommodate various propagation models as well as directional elements. Various distributed array systems are simulated using different parameters and the corresponding CCG is calculated. Optimization methods are explored for the purpose of improving the CCG of a distributed array system by adjusting the positions of the array nodes and the beam angle of the transmit array. Simulations are conducted to test and compare the improvements in system gain and the typical node displacement resulting from the application of optimization techniques.

15. SUBJECT TERMS
Distributed arrays, collaborative beamforming, coherent arrays, mobile ad hoc networks, optimization

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT Unclassified Unlimited	b. ABSTRACT Unclassified Unlimited	c. THIS PAGE Unclassified Unlimited	SAR	84

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Michael Lipski	(202) 767-5098

This page intentionally left blank

CONTENTS

EXECUTIVE SUMMARY E-1

1. BACKGROUND 1
   1.1 Motivation 1
   1.2 Key Contributions 2

2. COLLABORATIVE BEAMFORMING AND COHERENT DISTRIBUTED ARRAYS 4
   2.1 Overview 4
   2.2 Closed-Loop Versus Open-Loop Beamforming 4
   2.3 Node Synchronization 5

3. SYSTEM MODEL 9
   3.1 Volumetric System Model 9
   3.2 Planar System Model 14

4. TRANSMIT PATTERN PROPERTIES 16
   4.1 Planar Transmit Pattern 16
   4.2 Volumetric Transmit Pattern 20
   4.3 Periodic Arrays 25

5. OPTIMIZATION PROBLEM 31
   5.1 Problem Analysis 31
   5.2 Motivating Gradient Descent 33
   5.3 Steepest Descent Approach 34
   5.4 Finding Initial Beam Angle 35
   5.5 Distributed Array SQSD Algorithm 36
   5.6 Alternative Optimization Algorithms 39

6. CLOSED-FORM GRADIENT EXPRESSIONS 41
   6.1 Gradient Expressions for Planar Model 41
   6.2 Gradient Expressions for Volumetric Model 45

7. DISTRIBUTED ARRAY SIMULATION 51
   7.1 Metrics 52
   7.2 Results 52

8. CONCLUSION 72

REFERENCES 72

FIGURES

1 Curves for $P(G_c \geq X) = 0.9$ for values of internode range and orientation errors, from [1] 6

2 Probability of coherent gain exceeding 0.9 versus clock phase error, from [1] 7

3 Portion of normalized far-field power patterns for distributed transmit arrays using different numbers of nodes, along with level lines showing corresponding average power levels; $L_t = 100, \phi_0 = 0$ 18

4 Portion of normalized far-field power pattern of distributed transmit array; $N = 16, L_t = 100, \phi_0 = 0$. The peaks and nulls are indicated. 19

5 Distributions of angular width between successive nulls, or FNBW, for planar distributed arrays with different numbers of nodes; $L_t = 100, \phi_0 = 0$ 20

6 Distributions of FNBW values for cubic distributed arrays found by slicing transmit pattern along azimuth and elevation directions from the main beam; $N = 16, L_t = 1k, \phi_0 = 0, \theta_0 = \pi/2$ 22

7 Distributions of s_b values for different array sizes, calculated using FNBW values generated from cubic distributed arrays; $N = 16$ 23

8 Histogram of s_b values with a fitted gamma distribution 24

9 Node arrangement for a periodic volumetric distributed array with $N = 27$ nodes arranged in a cubic grid with dimension $L_t = 10$ 25

10 Azimuth cut of the transmit pattern of periodic volumetric distributed array with $N = 27$ nodes arranged in a cubic grid with dimension $L_t = 10$ and main beam at $(\phi_0, \theta_0) = (0, \pi/2)$. 26

11 Three-dimensional transmit pattern of periodic volumetric distributed array with $N = 27$ nodes arranged in a cubic grid with dimension $L_t = 10$ and main beam at $(\phi_0, \theta_0) = (0, \pi/2)$. 27

12 Histogram showing relative incidence of peak widths in degrees for periodic vs. aperiodic distributed arrays with $N = 27$ nodes and L_t around 100. 28

13 Histogram showing relative incidence of normalized intensity values for periodic vs. aperiodic distributed arrays with $N = 27$ nodes and L_t around 100. 29

14 Histogram showing relative incidence of peak widths in degrees for periodic vs. aperiodic distributed arrays with $N = 64$ nodes and L_t around 100. 30

15 Histogram showing relative incidence of normalized intensity values for periodic vs. aperiodic distributed arrays with $N = 64$ nodes and L_t around 100. 30

16 Plot of coherent communication gain as a function of the y and z positions of a single Tx node, with all other system nodes' positions fixed. For example system, $N = 16, M = 16, L_t = L_r = 100, D = 1,000$. 32

17 Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 16, M = 16, L_t = L_r = 100, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$ 53

18 Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 16, M = 16, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$ 54

19 Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 16, M = 16, L_t = L_r = 10k, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$ 54

20 Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 8, M = 8, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$ 55

21 Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 4, M = 32, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$ 56

22 Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 32, M = 4, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$ 56

23 Distributions of the improvement in coherent communication gain after optimizing node locations using SQSD versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 57

24 Distributions of the improvement in coherent communication gain after optimizing node locations using PS versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 58

25 Distributions of the improvement in coherent communication gain after optimizing node locations using PSO versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 58

26 Distributions of the average node displacement after optimizing node locations using SQSD versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 59

27 Distributions of the average node displacement after optimizing node locations using PS versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 60

28 Distributions of the average node displacement after optimizing node locations using PSO versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 60

29 Distributions of algorithm running time of SQSD used to optimize system gain versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 61

30 Distributions of algorithm running time of PS used to optimize system gain versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 62

31 Distributions of algorithm running time of PSO used to optimize system gain versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$ 62

32 Lower quartile values of improvement in coherent communication gain versus interarray distance compared across different optimization algorithms and array volumes; $N = 16, M = 16, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 64

33 Upper quartile values of average transmitter node displacement versus interarray distance compared across different optimization algorithms and array volumes; $N = 16, M = 16, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 65

34 Upper quartile values of average receiver node displacement versus interarray distance compared across different optimization algorithms and array volumes; $N = 16, M = 16, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 66

35 Upper quartile values of algorithm running time versus interarray distance compared across different optimization algorithms and array volumes; $N = 16, M = 16, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 67

36 Lower quartile values of improvement in coherent communication gain versus interarray distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 68

37 Upper quartile values of average transmitter node displacement versus interarray distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 69

38 Upper quartile values of average receiver node displacement versus interarray distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 70

39 Upper quartile values of algorithm running time versus interarray distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$ 71

EXECUTIVE SUMMARY

Mobile ad hoc networks (MANETs), drone-based wireless networks, or wireless sensor networks (WSNs) consisting of nodes with single omnidirectional antennas and limited transmit power can support wireless connectivity over long distances through coherent operation. Although designed for short-range networked communication, the individual radio nodes in a wireless network can be coordinated to form a sparse aperiodic array with a high-power directional gain. This action is called collaborative beamforming using coherent distributed arrays. By targeting a distant receiver with the array beam synthesized by a synchronized transmission, the network can ensure greater signal quality at the receiver. This function can be exploited to obtain increased transmission range, faster data rates, and improved power efficiency.

In the presented work, a model for a combined transmit-and-receive coherent distributed array system is considered. The system model describes a pair of distributed arrays, a transmit array and a receive array, that are defined as aperiodic volumetric arrays with cubic array volumes. First, the transmit-array side is explored by itself, and the resulting transmit patterns are examined using analytical expressions and their derivatives. Then, Monte Carlo simulations are used to validate average behavior in comparison with the analytical expressions.

Next, the entire signal path is considered by examining the combined transmit-and-receive distributed array system. Total gain for a combined transmit-and-receive coherent distributed array is labeled coherent communication gain (CCG), and a general expression for CCG is given. The general expression can accommodate various propagation models as well as directional elements. Various distributed array systems are simulated using different parameters and the corresponding CCG is calculated. Optimization methods are explored for the purpose of improving the CCG of a distributed array system by adjusting the positions of the array nodes and the beam angle of the transmit array. Simulations are conducted to test and compare the improvements in system gain and the typical node displacement resulting from the application of optimization techniques.

This page intentionally left blank

OPTIMIZING COMBINED TRANSMIT-AND-RECEIVE COHERENT DISTRIBUTED ARRAYS

1. BACKGROUND

There is a huge number of wireless devices in operation across the world, and that number is growing daily. Tunable transceiver ICs in the UHF/SHF bands, such as the AD 9363 and the AD 9375, contain an entire radio on a chip. These single-chip radios have found numerous applications in industrial, academic, and consumer environments. Coupling a transceiver IC with an FPGA or a microcontroller along with flash memory yields a programmable and adaptable compact wireless node. The ubiquity of small, low-power wireless devices has inspired a huge amount of research on mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), and cognitive radio (CR). The ready availability of mobile, low-power wireless network nodes enables unique opportunities for distributed networking techniques and protocols.

These devices are typically outfitted with a single, omnidirectional antenna and have low transmit range due to limited transmit power capability. Nodes within the ad hoc network may be able to broadcast to all other nodes, but it is assumed that they also need to communicate with entities outside of the network. The broadcast mechanism coupled with an omnidirectional antenna allows reliable, efficient intranetwork communication but presents limitations apropos the network communicating with a specific external entity, such as a sensor fusion center, a control nexus, or a relay point. The external target could perhaps be out of direct transmit range of any single node in the MANET in the absence of some sort of backhaul infrastructure.

This shortcoming can be remedied by employing some form of directional transmit/receive technology. One solution involves equipping nodes with directional antennas, but this could pose size/weight challenges, could introduce new limitations, and still may not enable individual radios to communicate across the needed distances. The other solution, which is the focus of this work, is collaborative beamforming using coherent distributed array formation.

Nodes in the ad hoc network can coordinate their wireless operations so that transmitted signals add up coherently at the location of the intended receiver. The network essentially becomes a distributed phased array, whereby multiple nodes are synchronized to send or receive an individual signal imbued with a directional array gain. A coherent distributed array (CDA) is synthesized by coordinating coherent distributed operations across wireless network nodes. This synthesis is also referred to as collaborative beamforming.

1.1 Motivation

Collaborative beamforming using distributed arrays has been an active area of research because of the potential payoff for both commercial and military wireless communications. Beamforming is included in the specifications for 5G and has been identified as a key technology for increasing data rates and reducing interference. Additionally, beamforming is crucial for the use of higher-frequency RF communications (e.g., millimeter wave) since the ability to concentrate the transmitted and received energy helps overcome range

Manuscript approved April 12, 2024.

and energy constraints. However, beamforming in 5G is accomplished primarily via massive multiple-input, multiple-output (MIMO) from base stations outfitted with dozens or hundreds of antennas. The Army has also demonstrated interest in this area with their Highly Directional RF program; their program encompasses a broad spectrum of beamforming topics including novel antenna engineering/materials/manufacturing, fixed arrays, and MIMO, in addition to distributed arrays.

Extensive prior work on collaborative beamforming has developed theory and has cultivated simulation results in an effort to demonstrate the feasible performance expectations and synchronization requirements at a fundamental level. Collaborative beamforming by distributed arrays has been a topic of research for the past two decades, but it has not truly progressed past the occasional small-scale experiments involving a single transmitter array and a single data stream. Most of the theoretical work is focused on transmit-side beamforming to a single receiver. In recent years, published work has revealed gaps and the need for additional research into "truly network" collaborative beamforming involving multiple transmitters, receivers, and data streams. Furthermore, much of the existing foundational work establishes that multiple different distributed-beamforming techniques are viable and that the synchronization and coordination requirements are within reach.

Distributed, collaborative beamforming and beam-nulling can be leveraged to improve communication range, throughput, reliability, and low observability. The ability to form dynamic distributed arrays as an added capability would endow distributed networks, particularly those deployed in a tactical context, with greater flexibility and resiliency. Distributed array operations would be hardware-independent and, in addition to increasing maximum communications range, would grant network users increased ability to harden network communications against external interference, to limit emissions toward hostile receivers, and to reduce interference to friendly receivers. In many situations, directional RF solutions are needed to ensure reliable communications over great distances, and collaborative beamforming is the most effective solution for certain disadvantaged platforms with size, weight, and power (SWaP) constraints, such as UAVs and USVs forming a MANET.

1.2 Key Contributions

This work examines system models of coherent distributed arrays and provides a generalized model for a coherent communication system consisting of a distributed transmit array beamforming messages to a distributed receive array. The distributed arrays considered are aperiodic volumetric arrays of homogeneous elements. Included with the system model are expressions for coherent communication gain, which is a measure of the cumulative gain of a combined transmit-and-receive distributed array system.

Initially, the transmit side of the communication system is considered, and the properties of the transmit patterns for several different types of distributed array geometries are described. Simulations are used to provide empirical comparisons for distributions of peak and beam width and off-beam intensity, and expressions are provided for finding the average first-null beamwidth as a function of the dimensions of the array volume.

Next, both the transmit and receive sides of the coherent communication system are considered together. Properties of coherent communication gain are explored. A formal definition of the optimization problem is given, followed by a discussion of the problem's complexity. The coherent communication gain optimization uses the positions of the nodes as optimization variables, but also includes the azimuth and elevation angles of the transmit array's beam. Attempting to find the global maximum of a nonlinear, highly nonconvex,

high-dimensional objective function is generally difficult, and there would be no way to guarantee the optimality of the solution without discretizing the search space and exhaustively evaluating every point. In the context of collaborative beamforming in wireless networks, particularly MANETs or WSNs, the computational cost of running an optimization algorithm on a battery-powered mobile node can be prohibitive. Additionally, it is important to consider that the locations of the network nodes are likely not arbitrary, that their placement is deliberate and perhaps essential to the purpose of the network. With this is mind, local optimization is conducted using gradient descent. Square quadratic steepest descent (SQSD) is a first-order gradient descent algorithm with an auto-adjusting step size and is adapted to the combined transmit-and-receive distributed array optimization problem. Closed-form expressions for the gradient terms of the objective function are derived and used to evaluate the gradient during simulation.

MATLAB simulations generate aperiodic distributed array systems consisting of a transmit array and a receive array and then attempt to maximize system gain using one of three possible algorithms. SQSD is compared against two global heuristic optimization algorithms, pattern search and particle swarm optimization. Distributed array scenarios vary based on distance between transmit and receive arrays, volume of transmit and receive arrays, and number of transmit or receive nodes. Total system gain is calculated before and after the optimization is performed and the difference constitutes the gain improvement resulting from optimization. In addition, transmitter and receiver node average displacement and algorithm running time are recorded. SQSD and particle swarm optimization were similar in their ability to improve system gain and could reliably deliver more than 3 dB of improvement with similar average node displacement distances. However, SQSD is considerably faster than particle swarm optimization, running at least 25 times faster on average.

2. COLLABORATIVE BEAMFORMING AND COHERENT DISTRIBUTED ARRAYS

2.1 Overview

The spatial diversity of mobile wireless devices in an ad hoc network can be exploited to form sparse aperiodic arrays. The most widely studied application is collaborative transmit beamforming: the typical scenario is a network of identical, randomly spaced, wireless nodes with omnidirectional transmit patterns synchronizing their phase offsets to send a common message to a distant receiver with enhanced signal strength [1–3]. These systems are known as coherent distributed arrays (CDAs) and they offer the potential of higher data rates, greater range, and lower energy consumption. Additionally, the directional nature of the communications allows for receiver selectivity and reduced interference in the off-beam directions. Assuming N collaborating nodes transmitting with identical fixed transmit power, received power at the intended destination is increased by a factor of N^2 under ideal circumstances [4].

Most of the work on distributed arrays focuses on coherent transmit arrays, or distributed transmit beamforming, but distributed array reception is also possible by aggregating the digitized received signals from all participating nodes at a designated node for processing. Techniques for distributed receive beamforming are essentially short-link distributed transmit beamforming amongst the receivers, and the signal-to-noise ratio (SNR) gain scales linearly with the number of receivers [5, 6]. It was also shown that by exchanging hard decisions rather than the entire received signal capture, the receiver array could operate more efficiently while only sacrificing at most a few dB of gain [7].

2.2 Closed-Loop Versus Open-Loop Beamforming

Perhaps some of the greatest practical challenges to implementing a coherent distributed transmit array using an ad hoc network involve coordinating the nodes so that the individual transmit signals add up coherently in the desired direction. Collaborative transmit beamforming approaches can be roughly broken down into two types based upon whether or not the transmit side is using feedback from the receive side to assist in coordinating the emissions from the distributed array. These two approach categories are described as closed-loop and open-loop [1].

Closed-loop coherent distributed arrays use external inputs, typically feedback from the receiver, to synchronize node transmissions. Feedback indicating signal strength or change in signal strength from the receiver allows the array nodes to adjust their phase offsets to optimize array gain. The advantage of closed-loop systems is that the internode coordination requirements are relaxed, or sometimes are absent altogether if the transmit nodes have continuous feedback from the receive location. The disadvantage of using a closed-loop approach is that the transmitter array can only achieve effective beamforming gain toward a target that is sending it RSS feedback. This feedback can be as simple as a one-bit message indicating whether RSS is increasing or decreasing as the transmitters adjust their phase. A number of closed-loop beamforming approaches are discussed in [1, 4, 8–13].

An open-loop coherent transmit array takes the receiver direction with respect to the array as input and adjusts the phase offset of every transmitting node depending on the node's relative position and the direction of the main beam. Because there is no receiver feedback, the phase offsets need to be calculated analytically with no mechanism for optimizing performance. Acceptable performance is sought by ensuring tight synchronization of carrier phase between nodes and accurate determination of node positions and beam direction relative to some reference point. The open-loop system has more stringent coordination requirements but can beamform in an arbitrary input direction. Synchronization methods and error analysis for open-loop systems can be found in [1–4, 10, 14–19].

2.3 Node Synchronization

Synchronization requirements for a distributed coherent transmit array of identical nodes with omnidirectional patterns are given in terms of maximum acceptable errors in clock signal, internode ranging, and relative beam angle [1, 20]. Given a distributed transmit array of N homogeneous nodes placed arbitrarily and coordinating to form a beam toward a selected direction, the signal received by a target in the far field of the array nodes is given by

$$s_r(t) = |E| = \sum_{n=1}^{N} h_n A_n(t - \tau_n) e^{j[2\pi(f+\delta f_n)t) + \phi_{s,n} + \delta\phi_{c,n}]}, \tag{1}$$

where

- h_n is the complex channel coefficient for the propagation channel between node n and the far-field receiver,
- A_n is the amplitude of the transmitted signal,
- τ_n is the error in time delay,
- δf_n is the frequency error,
- $\delta\phi_{c,n}$ is the clock error, and
- $\phi_{s,n}$ is the beamsteering phase.

The beamsteering phase is given by

$$\phi_{s,n} = \frac{2\pi}{\lambda}(d_n - \delta d_n)\cos(\theta_n - \delta\theta_n), \tag{2}$$

which contains the node localization/ranging error δd_n and the beam angle/orientation error $\delta\theta_n$.

Many mobile wireless devices are equipped with GPS, which can provide a precise time reference and location in an absolute sense. As to whether or not GPS would be sufficient for synchronization, it does not appear to be a reliable solution for general open-loop coherent beamforming. GPS accuracy depends on many factors, including relative satellite positions, atmospheric conditions, signal blockage, and receiver design. Additionally, there are techniques to improve GPS accuracy, such as the use of additional frequencies. There is a service called TerraStar Correction Services, which uses additional reference satellites to correct GNSS data for higher accuracy. Obtaining position to within 1 m with GPS would be straightforward, but that is not nearly good enough for coherent array operation. The highest accuracy claimed is within 1 cm + 1 ppm using real-time kinematic (RTK), which is a technique that incorporates carrier signal phase for higher accuracy. For GPS time accuracy, GPS.gov states a time transfer accuracy of within 40 ns.

The simulations in [1] show how the gain of an open-loop coherent distributed array is affected by errors in internode distance estimation, node orientation, and carrier phase. Figure 1 shows the curves corresponding to the average received power exceeding X with a probability of 0.9, or $P(G_C \geq X) = 0.9$ for different values of error in internode range σ_d and node orientation σ_θ, where X is some fraction of the maximum theoretical coherent gain that would be possible with perfect synchronization. Figure 2 shows probability of coherent gain exceeding 0.9 versus clock phase error ϕ_ψ for coherent distributed arrays containing different numbers of nodes. As the number of coherent nodes increases, a cutoff threshold for clock phase error emerges. Rather than a gentle degradation of array gain, the array becomes a two-state system that switches based upon error in relative clock phase. The simulations in [1] indicate that when taken individually, the internode ranging needs to be accurate to within $\lambda/15$, the relative angle must be known to within 7°, and the relative errors between any two clocks must be below 18°.

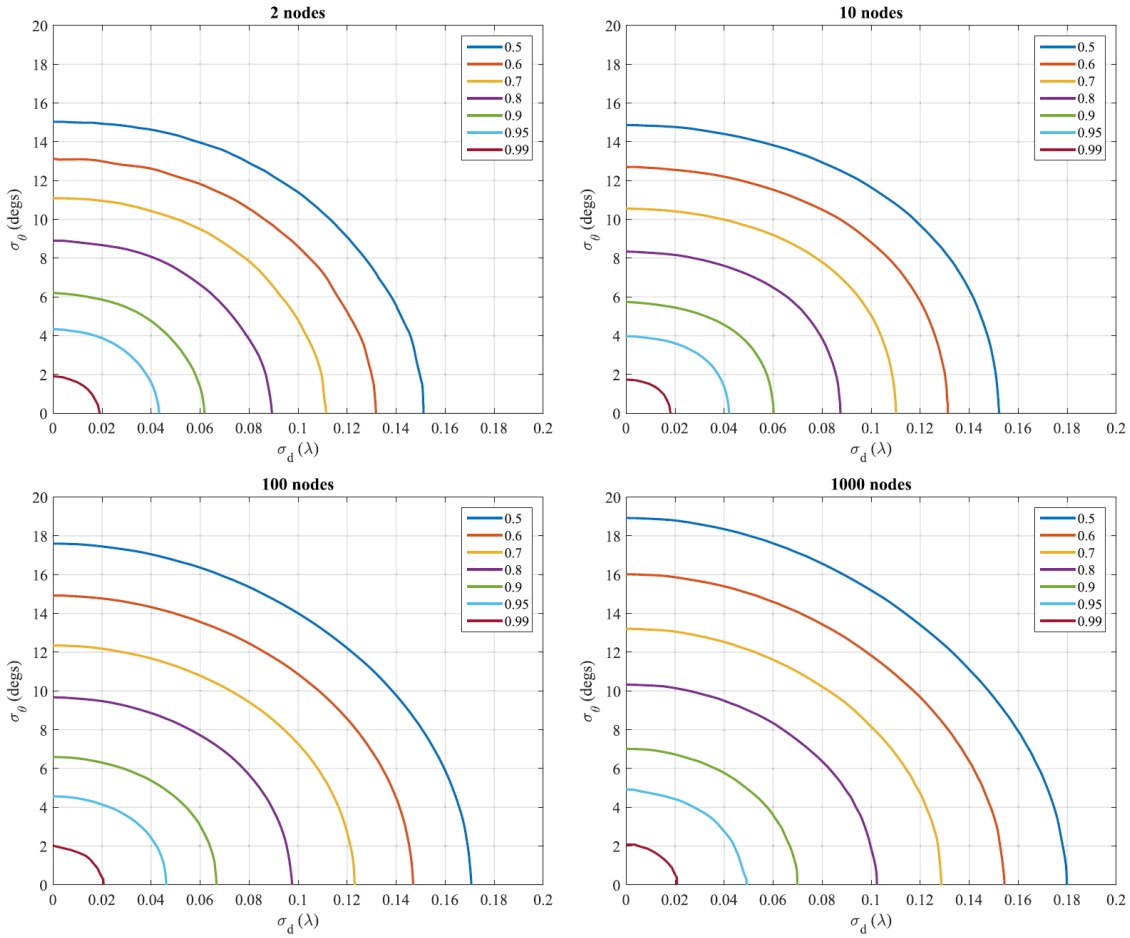

Fig. 1—Curves for $P(G_c \geq X) = 0.9$ for values of internode range and orientation errors, from [1]

So, as a best-case estimate, using correction techniques and accuracy enhancements, high-end GPS receivers claim positioning accuracy to within 1 cm and accuracy of distributed time to within 20 ns. The plots in [1] indicate that internode ranging needs to be accurate to within $\lambda/15$, which, for a 1 GHz carrier, is 2 cm. Theoretically, GPS could be used as the sole technology for internode ranging, but 1 cm accuracy is optimistic

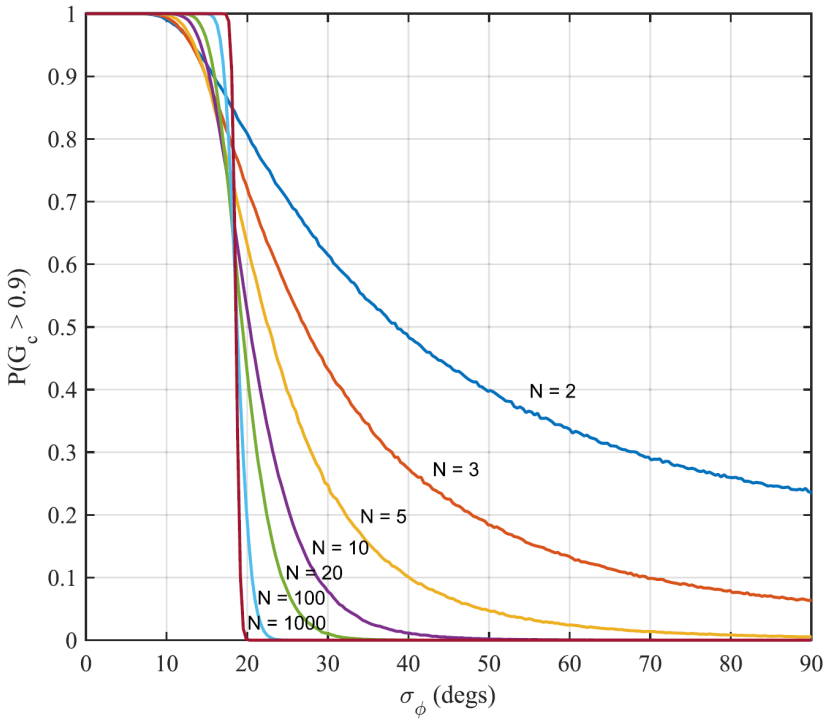

Fig. 2—Probability of coherent gain exceeding 0.9 versus clock phase error, from [1]

and likely depends on ideal operating conditions. GPS may be sufficient for location estimation in certain circumstances but may not be sufficient for synchronizing clock references. Furthermore, a wireless ad hoc network that relies solely upon GPS for localization is less flexible; it would be unable to deploy open-loop coherent distributed array techniques in GPS-denied environments or in situations where the received satellite signals cannot provide localization to the required accuracy.

Another important aspect to consider for open-loop beamforming is frequency synchronization among the collaborating transmitters. From Eq. (1), the frequency error is given by δf_n and represents the offset or difference between the intended frequency and the actual frequency of the local reference. In a distributed array or wireless ad hoc network, each transmitter generates its own carrier frequency from its local oscillator (LO), which is typically derived from an onboard crystal oscillator. The carrier frequency between any two transmitters in the distributed transmit array is not going to be exactly the same. This offset in carrier frequency can be traced to a range of causes, including temperature differences, Doppler shift, and manufacturing defects. When the carrier frequency is in the GHz range, the carrier frequency offsets can be more than several kHz. In order to ensure consistent carrier phase synchronization, the collaborating transmitters need to have their frequency references synchronized.

A promising method for combining internode ranging and frequency synchronization is described in [19]. Absolute localization is not necessary for open-loop coherent distributed array formation, so approaches such as this one that perform relative internode localization can be a more feasible solution. With this technique, a node in the distributed network is elected as the controller for a message. The controller broadcasts a spectrally sparse two-tone continuous wave (TTCW) ranging signal. A 10 MHz frequency reference is modulated onto the lower tone. Other nodes in the network use PLLs to synchronize with the controller's broadcasted

frequency reference. This technique solves the problems of node localization within the network and a distributed one-way frequency reference transfer. Experiments show that the TTCW combined ranging and frequency transfer can achieve high-accuracy ranging to within 3.35 mm and low-error frequency estimation below the desired threshold of 18° [19].

Recently, work was introduced that examines the relationships between beamforming gains and channel coherence time, SNR, preamble lengths, and the number of nodes in the context of a receiver-led coherent distributed beamforming protocol [21]. Another paper builds on this and calculates the maximum distance over which distributed beamforming can be performed reliably on account of beamforming gain degradation due to synchronization and phase estimation errors increasing with distance [22]. In addition to strict open- or closed-loop distributed beamforming, a type of hybrid technique called *guided beamforming* can be used to establish a coherent link to a distant receiver within line of sight (LoS) by assigning one of the Tx nodes to act as the "guide" and having the remaining nodes adjust the phasing of their signals to ensure they combine coherently at the guide [23]. There has also been work done on noncoherent distributed beamforming. Examples include the proposed noncoherent beamformers for downlink in frequency division duplexing (FDD) massive MIMO in [24], which do not rely on feedback from the receive side. Also, in [25], another zero-feedback, noncoherent beamforming technique is proposed that is based on repetition coding and is tested using software-defined radio (SDR).

As this section has shown, a lot of prior work has been done to construct the foundational theory and system models for distributed beamforming and to derive the performance of distributed arrays under nonideal circumstances. According to this prior work, although coherent distributed beamforming requires strict synchronization, particularly for open-loop operation, the requirements do not pose an insurmountable challenge. Thus, it is reasonable to assume reliable open-loop operation under typical conditions (i.e., mild to moderate mobility and interference) for the sake of developing higher-layer methods and algorithms.

3. SYSTEM MODEL

In this chapter, the mathematical model for a combined transmit-and-receive coherent distributed array (CDA) communication system is developed. This model describes a collection of homogeneous wireless nodes distributed throughout a constrained volume that form a coherent distributed transmit array in order to send beamformed messages to a second group of homogeneous wireless nodes. This second group of nodes is spread throughout the confines of a distant volume and acts as a distributed receive array in order to receive the transmit array messages. The transmit array and the receive array together form the distributed array communication system. The model assumes that the nodes are able to localize and synchronize time and phase to high degrees of accuracy and precision, although specific synchronization methods are not explored in depth.

3.1 Volumetric System Model

The fundamental, recurring model for a coherent communication system consists of two distributed arrays; one array is designated as the transmitter array, and the other is designated as the receiver array. All nodes in the communication system exist in a three-dimensional space, and all distance parameters are defined in wavelengths. The transmitter array contains N nodes distributed over a cubic volume with side dimension L_t, and the receiver array contains M nodes distributed over a cubic volume with side dimension L_r. All nodes in the network are considered identical, and each node is equipped with a single omnidirectional antenna for either transmission or reception. The transmitter and receiver arrays are separated by distance D. Let $S_N^t = \{s_1^t, ..., s_N^t\}$ be the set of all N transmitter nodes and $S_M^r = \{s_1^r, ..., s_M^r\}$ be the set of all M receiver nodes. The position of transmitter node s_n^t is described by the coordinates (x_n^t, y_n^t, z_n^t), $1 \le n \le N$; similarly, the position of receiver node s_m^r is described by (x_m^r, y_m^r, z_m^r), $1 \le m \le M$. The distance between transmitter s_n^t and receiver s_n^r is denoted by d_{nm}, $1 \le n \le N$; $1 \le m \le M$.

The transmitters operate as an open-loop coherent distributed array, which can steer beams arbitrarily. The Cartesian coordinate system is centered on the transmitter array volume such that $\{(x_n^t, y_n^t, z_n^t) : x_n^t, y_n^t, z_n^t \in [-\frac{L_t}{2}, \frac{L_t}{2}]\}$. The receiver array volume is centered on the X axis such that $\{(x_m^r, y_m^r, z_m^r) : x_m^r \in [\frac{L_t}{2} + D, \frac{L_t}{2} + D + L_r], y_m^r, z_m^r \in [-\frac{L_r}{2}, \frac{L_r}{2}]\}$. Angle ϕ_m describes the azimuth angle of receiver node s_m^r relative to the transmitter array, and angle θ_m is the elevation angle of s_m^r relative to the transmitter array. For ease of notation, the coordinates of all nodes in the coherent communication system are gathered in $\mathbf{B}$, which is a $3(N + M) \times 1$ matrix defined as

$$\mathbf{B} = \big[x_1^t, ..., x_N^t, y_1^t, ..., y_N^t, z_1^t, ..., z_N^t, \\ x_1^r, ..., x_M^r, y_1^r, ..., y_M^r, z_1^r, ..., z_M^r\big]^{\mathrm{T}}. \tag{3}$$

3.1.1 Array Factor

In an open-loop distributed transmitter array of N nodes, each transmitter sends the same signal with its own time and phase offset calculated such that all the transmitted signals are received simultaneously and in phase at a targeted location. A receiver placed at the targeted location observes constructive interference of the transmit signal and can receive the signal with a power gain of up to N^2. The target of the coherent

transmission is given the coordinates (x_0, y_0, z_0), and the Euclidean distance between node s_n^t and the beam target at (x_0, y_0, z_0^t) is given by $d_{n0}, 1 \le n \le N$. In order to achieve maximum array gain, the transmitted signals need to add up in phase at (x_0, y_0, z_0), so for each node $s_n^t, 1 \le n \le N$, the initial phase is set to

$$\Psi_n = -\frac{2\pi}{\lambda} d_{n0}. \tag{4}$$

Array factor is a function describing the amplitude gain of the transmitter array, relative to a single transmitter, at any location in $\mathbf{R}^3$. The array factor of the transmitter array signal observed at point A located at (x, y, z) is

$$F(x, y, z) = \sum_{n=1}^{N} e^{j\Psi_n} e^{j\frac{2\pi}{\lambda} d_{nA}} = \sum_{n=1}^{N} e^{j\frac{2\pi}{\lambda}(d_{nA} - d_{n0})}, \tag{5}$$

where d_{nA} is the Euclidean distance between node s_n^t and A. Notice that by setting the target of the distributed transmitter array to be observation point A, then $d_{nA} = d_{n0}$, yielding an amplitude gain of N at A and a power gain of N^2.

Multiple forms of the expression for the array factor of the distributed transmitter array can be found in [3, 26], for example. Assume that the target of the transmitter array is in the far field; then, the beam target can be defined using the azimuth angle ϕ_0 and the elevation angle θ_0, and the general expression for the array factor of the transmitter array is given by

$$F(\phi, \theta) = \sum_{n=1}^{N} e^{j(a_0 x_n^t + b_0 y_n^t + c_0 z_n^t)}, \tag{6}$$

where

$$a_0 = 2\pi(\sin\theta\cos\phi - \sin\theta_0\cos\phi_0), \tag{7}$$

$$b_0 = 2\pi(\sin\theta\sin\phi - \sin\theta_0\sin\phi_0), \tag{8}$$

$$c_0 = 2\pi(\cos\theta - \cos\theta_0). \tag{9}$$

3.1.2 Intensity

Radiation intensity is a far-field parameter describing the power radiated from an antenna or an array per unit solid angle [27]. A useful measure for comparing the patterns of distributed array configurations is normalized intensity, which is given by

$$U(\phi, \theta) = \frac{1}{N^2} \left| \sum_{n=1}^{N} e^{j(a_0 x_n^t + b_0 y_n^t + c_0 z_n^t)} \right|^2 . \tag{10}$$

3.1.3 Combined Transmit-and-Receive Arrays

Array factor and intensity describe the amplitude gain of the transmitter array at a single target relative to a single transmitter. However, in the combined transmit-and-receive distributed array system, the beamformed message from the transmitter array is collected and summed by the M receivers of the receive array. A similar coherent communication system is proposed in [17, 28], and the metric called coherent communication gain is introduced. Coherent communication gain (CCG), given by $G(\mathbf{B}, \phi_0, \theta_0)$, is a power gain; because gain is a ratio, a reference, or benchmark, is needed against which the distributed array system can be compared. In the case of the distributed transmitter array, the array factor describes the ratio of the summed amplitude of the signals from all of the transmitters to the amplitude of a single transmitted signal. CCG is a similar concept for combined transmit-and-receive distributed array systems. In the simplest form of the system model, $G(\mathbf{B}, \phi_0, \theta_0)$ can be viewed as the squared array factor summed across all of the receivers.

To derive a true power gain, the total summed received power must be compared against a reference system that does not have array gain. The reference, or benchmark, system is a simple point-to-point communication system. Instead of transmitter-and-receiver arrays, the reference system comprises a single transmitter located at the center of the transmitter array volume at coordinates $(0, 0, 0)$ and a single receiver placed at the center of the receiver array volume at coordinates $(\frac{L_t}{2} + D + \frac{L_r}{2}, 0, 0)$. Let $v(t)$ be the common transmit signal with a duration T and let A_n be the amplitude of the signal transmitted from s_n^t. The complex-valued coefficient describing the channel between s_n^t and s_m^r is given by $h_{nm} = \alpha_{nm} e^{j\beta_{nm}}$, which is dependent on the positions of s_n^t and s_m^r and the specific propagation model. For the analysis and simulations throughout this work, a free-space path loss model is generally assumed.

A second frame of reference is introduced, and it is given by (x', y', z'). This primed coordinate system is relative to the receiver array and is defined as $(x', y', z') = (x - \frac{L_t}{2} - D - \frac{L_r}{2}, y, z)$. The origin of the primed coordinate system resides in the center of the receiver array, and the primed azimuth ϕ' and primed elevation θ' are found relative to the x-axis and the primed origin. The antenna field pattern of s_n^t is given by $G_n^t(\phi, \theta)$, and the antenna field pattern of s_m^r is given by $G_m^r(\phi', \theta')$.

To define CCG, the received power of the reference point-to-point communication system needs to be defined. The total received energy at the benchmark single receiver over the duration of the signal sent from the benchmark single transmitter is given by

$$E_b = \int_{t=0}^{T} \left| G_b^t(\phi_b, \theta_b) G_b^r(\phi_b', \theta_b') \alpha_b A_b v(t) e^{j\beta_b} \right|^2 dt. \tag{11}$$

CCG is the total received power summed across all receivers divided by E_b. Thus, the generalized expression for CCG is

$$G(\mathbf{B}, \phi_0, \theta_0) = \frac{1}{E_b} \sum_{m=1}^{M} \int_{t=0}^{T} \left| \sum_{n=1}^{N} G_n^t(\phi_m, \theta_m) G_m^r(\phi'_n, \theta'_n) \times \alpha_{nm} A_n v(t) e^{j\beta_{nm}} e^{j(a_m x_n^t + b_m y_n^t + c_m z_n^t)} \right|^2 dt, \tag{12}$$

where

$$a_m = 2\pi(\sin\theta_m \cos\phi_m - \sin\theta_0 \cos\phi_0), \tag{13}$$

$$b_m = 2\pi(\sin\theta_m \sin\phi_m - \sin\theta_0 \sin\phi_0), \tag{14}$$

$$c_m = 2\pi(\cos\theta_m - \cos\theta_0). \tag{15}$$

Because in this system model, all nodes in the network are equipped with omnidirectional antennas, let $G_n^t(\phi, \theta) = G_m^r(\phi', \theta') = 1$. The upper bound on CCG for a system where the maximum gain of all nodes is 1 is $G_{UB} = N^2 M$. Additionally, power terms that remain constant in the scope of the problem can be combined into a single term, which is hereby defined as

$$E_G = \frac{1}{E_b} \int_{t=0}^{T} |v(t)|^2 dt. \tag{16}$$

Using these substitutions, Eq. (12) can be simplified to

$$\begin{aligned} G(\mathbf{B}, \phi_0, \theta_0)\Big|_{(G_n^t(\phi,\theta)=1, G_m^r(\phi',\theta')=1)} &= E_G \sum_{m=1}^{M} \left| \sum_{n=1}^{N} h_{nm} A_n \right. \\ &\quad \left. \times e^{j(a_m x_n^t + b_m y_n^t + c_m z_n^t)} \right|^2 \\ &= \tilde{G}(\mathbf{B}, \phi_0, \theta_0). \end{aligned} \tag{17}$$

3.1.4 Benchmark Scenario

The original formulation of CCG in [17, 28] is based on a specific benchmark scenario and a set of assumptions. Here, the original form of G, given by $G_{BM}(\mathbf{B}_p)$, is derived from the generalized expression in Eq. (12). In the benchmark scenario, all nodes have omnidirectional patterns and identical signal amplitudes, so $G_n^t(\phi,\theta) = G_m^r(\phi',\theta') = 1$ and $A_n = A$. The time-domain signal transmitted by the transmitter array nodes is $v(t) = \sin(\frac{2\pi t}{T})$. Using Eq. (11), the received energy over a single period for the benchmark scenario of $N = 1, M = 1$ is

$$\begin{aligned} E_{11} &= \int_{t=0}^{T} \left| h_{11} A \sin\left(\frac{2\pi t}{T}\right)\right|^2 dt \\ &= h_{11}^2 A^2 \int_{t=0}^{T} \left|\sin\left(\frac{2\pi t}{T}\right)\right|^2 dt \\ &= \frac{h_{11}^2 A^2 T}{2}. \end{aligned} \tag{18}$$

Using Eq. (5), Eq. (12) is reformulated as

$$G(\mathbf{B},\phi_0,\theta_0) = \frac{1}{E_b}\sum_{m=1}^{M}\int_{t=0}^{T}\left|\sum_{n=1}^{N} G_n^t(\phi_m,\theta_m) G_m^r(\phi_n',\theta_n') \alpha_{nm} A_n v(t) e^{j\beta_{nm}} e^{j\frac{2\pi}{\lambda}(d_{nA}-d_{n0})}\right|^2 dt, \tag{19}$$

followed by setting $\theta_{nm} = \frac{2\pi}{\lambda} d_{nm}$ and $\theta_n = -\frac{2\pi}{\lambda} d_{n0}$ to yield

$$\begin{aligned} G_{BM}(\mathbf{B},\phi_0,\theta_0) =& \frac{1}{E_{11}}\sum_{m=1}^{M}\left|\sum_{n=1}^{N} h_{nm} A e^{j(\theta_{nm}+\theta_n)}\right|^2 \times \int_{t=0}^{T}\left|\sin\left(\frac{2\pi t}{T}\right)\right|^2 dt \\ =& \frac{1}{h_{11}^2}\sum_{m=1}^{M}\left|\sum_{n=1}^{N} h_{nm} e^{j(\theta_{nm}+\theta_n)}\right|^2. \end{aligned} \tag{20}$$

Euler's formula is applied, which yields

$$G_{BM}(\mathbf{B},\phi_0,\theta_0) = \frac{1}{h_{11}^2}\sum_{m=1}^{M}\left|\sum_{n=1}^{N} h_{nm}\cos(\theta_{nm}+\theta_n) + j\sum_{n=1}^{N} h_{nm}\sin(\theta_{nm}+\theta_n)\right|^2. \tag{21}$$

The following substitutions are made, with accompanying observations:

$$u = \sum_{n=1}^{N} h_{nm} \cos\left(\theta_{nm} + \theta_n\right) \tag{22}$$

$$w = \sum_{n=1}^{N} h_{nm} \sin\left(\theta_{nm} + \theta_n\right) \tag{23}$$

$$r = u + jw \tag{24}$$

$$|r|^2 = rr* = (u + jw)(u - jw) = u^2 + w^2 \tag{25}$$

Substituting back yields the original form of coherent communication gain given in [17, 28], which is

$$G_{BM}(\mathbf{B}, \phi_0, \theta_0) = \frac{1}{h_{11}^2} \sum_{m=1}^{M} \left(\left(\sum_{n=1}^{N} h_{nm} \cos\left(\theta_{nm} + \theta_n\right) \right)^2 + \left(\sum_{n=1}^{N} h_{nm} \sin\left(\theta_{nm} + \theta_n\right) \right)^2 \right). \tag{26}$$

3.2 Planar System Model

In certain scenarios, such as a network of ground nodes deployed over a relatively flat region, the communicating nodes are all roughly coplanar. Using the full volumetric model in these instances may be unnecessary, and, particularly for distributed systems with limited computing resources, using a simplified planar model makes more sense. In this section, the parameters of the coherent distributed array communication system are modified so that the system model can be adapted from a volumetric model to a planar model.

The fundamental architecture for the coherent distributed array communication system is unchanged: a designated transmitter array containing N nodes communicates with a designated receiver array containing M nodes. All nodes in the communication system are coplanar in the XY plane, and all distance parameters are defined in wavelengths. The transmitter array is defined by a square area with side dimension L_t. All N transmitter nodes are distributed within the area defined by the $L_t \times L_t$ boundary. Similarly, the receiver array is defined by an area with dimensions $L_r \times L_r$, and all M receiver nodes are distributed within the square boundary.

The transmitters operate as an open-loop coherent distributed array, which can steer beams arbitrarily. Because all nodes in the coherent communication system are in the XY plane, the transmitter array beam direction can be defined using a single angle. The Cartesian coordinate system is centered on the transmitter array area such that $\{(x_n^t, y_n^t) : x_n^t, y_n^t \in [-\frac{L_t}{2}, \frac{L_t}{2}]\}$. The receiver array area is centered on the X axis such that $\{(x_m^r, y_m^r) : x_m^r \in [\frac{L_t}{2} + D, \frac{L_t}{2} + D + L_r], y_m^r \in [-\frac{L_r}{2}, \frac{L_r}{2}]\}$. Angle ϕ_m describes the azimuth angle of receiver node s_m^r relative to the transmitter array. In a manner similar to that of the volumetric model, the coordinates of all nodes in the coherent communication system are gathered in a single variable $\mathbf{B}_p$, which is a $2(N + M) \times 1$ matrix defined as

$$\mathbf{B}_p = \left[x_1^t, ..., x_N^t, y_1^t, ..., y_N^t, x_1^r, ..., x_M^r, y_1^r, ..., y_M^r\right]^{\mathrm{T}}. \tag{27}$$

3.2.1 Array Factor and Intensity

Because all nodes are in the XY plane, the beam target of the transmitter array can be described solely using the azimuth angle ϕ_0. Assuming that the target of the transmitter array is defined by ϕ_0 and is considered to be in the far field, the general expression for the array factor of the transmitter array along the azimuth direction ϕ is given by

$$F(\phi) = \sum_{n=1}^{N} e^{j2\pi[x_n^t(\cos\phi - \cos\phi_0) + y_n^t(\sin\phi - \sin\phi_0)]}. \tag{28}$$

Normalized intensity for the planar transmitter array along the azimuth direction is given by

$$U(\phi) = \frac{1}{N^2} \sum_{n=1}^{N} \left| e^{j2\pi[x_n^t(\cos\phi - \cos\phi_0) + y_n^t(\sin\phi - \sin\phi_0)]} \right|^2. \tag{29}$$

3.2.2 Coherent Communication Gain

The total SNR gain for the combined transmit-and-receive coherent array system in the planar model is given by coherent communication gain (CCG), which is derived from the CCG equations for the volumetric model. From Eq. (12), the expression for CCG using the planar model is given by

$$\begin{aligned} G(\mathbf{B}_p, \phi_0) = \frac{1}{E_b} \sum_{m=1}^{M} \int_{t=0}^{T} \Bigg| \sum_{n=1}^{N} G_n^t(\phi_m) G_m^r(\phi_n') \alpha_{nm} A_n \\ \times v(t) e^{j\beta_{nm}} e^{j2\pi[x_n^t(\cos\phi_m - \cos\phi_0) + y_n^t(\sin\phi_m - \sin\phi_0)]} \Bigg|^2 dt. \end{aligned} \tag{30}$$

Under the assumption that all nodes are equipped with omnidirectional antennas, yielding $G_n^t(\phi_m) = G_m^r(\phi_n') = 1 \forall \phi_m, \phi_n'$, and using the substitution in Eq. (16), the simplified CCG expression is

$$\begin{aligned} G(\mathbf{B}_p, \phi_0)\Big|_{(G_n^t(\phi)=1, G_m^r(\phi')=1)} &= E_G \sum_{m=1}^{M} \left| \sum_{n=1}^{N} h_{nm} A_n e^{j2\pi[x_n^t(\cos\phi_m - \cos\phi_0) + y_n^t(\sin\phi_m - \sin\phi_0)]} \right|^2 \\ &= \tilde{G}(\mathbf{B}_p, \phi_0). \end{aligned} \tag{31}$$

4. TRANSMIT PATTERN PROPERTIES

In this chapter, the focus is on the distributed transmit array half of the coherent communication system. Characteristics of the far-field array pattern are examined using empirical modeling, the results of which are compared to analytical expressions from literature.

In [3] and [29], equations are given for the average far-field intensity patterns of distributed arrays with nodes randomly distributed within the array area or volume. The model in [3] assumes a planar arrangement of nodes with an array area defined as a circle, whereas [29] discusses a number of different planar and volumetric distributed arrays.

4.1 Planar Transmit Pattern

First, assume the planar system model for a distributed transmit array, where N identical nodes are distributed on the XY plane in a uniform random distribution across a square array area with dimensions $L_t \times L_t$. From [29], the average normalized far-field intensity of the square planar distributed transmit array is found using

$$\alpha(\phi,\theta) = 2\pi L_t(\sin\theta\cos\phi - \sin\theta_0\cos\phi_0) \tag{32}$$

$$\beta(\phi,\theta) = 2\pi L_t(\sin\theta\sin\phi - \sin\theta_0\sin\phi_0) \tag{33}$$

$$U_{av}(\phi,\theta) = \frac{1}{N} + (1-\frac{1}{N})|\mathrm{sinc}(\alpha(\phi,\theta))|^2|\mathrm{sinc}(\beta(\phi,\theta))|^2. \tag{34}$$

Since the beam target is also located on the XY plane, only the azimuth pattern is relevant, so $\theta = \theta_0 = \pi/2$. The average beampattern equations are simplified to

$$\alpha(\alpha) = 2\pi L_t(\cos\phi - \cos\phi_0) \tag{35}$$

$$\beta(\phi) = 2\pi L_t(\sin\phi - \sin\phi_0) \tag{36}$$

$$U_{av}(\phi) = \frac{1}{N} + (1-\frac{1}{N})|\mathrm{sinc}(\alpha(\phi))|^2|\mathrm{sinc}(\beta(\phi))|^2. \tag{37}$$

Consider now a planar distributed transmit array where the array area is instead defined by a circle with radius R. The equations for the average complete far-field beampattern of the circular distributed transmit array are given by

$$\zeta(\phi,\theta) = 2\pi R\sqrt{(\sin\theta\cos\phi - \sin\theta_0\cos\phi_0)^2 + (\sin\theta\sin\phi - \sin\theta_0\sin\phi_0)^2} \tag{38}$$

$$U_{av}(\phi,\theta) = \frac{1}{N} + (1-\frac{1}{N})|2\mathrm{jinc}(\zeta(\phi,\theta))|^2. \tag{39}$$

The average azimuth pattern for the circular distributed array with $\theta_0 = \pi/2$ is given by

$$\zeta(\phi) = 2\pi R\sqrt{(\cos\phi - \cos\phi_0)^2 + (\sin\phi - \sin\phi_0)^2} \tag{40}$$

$$U_{av}(\phi) = \frac{1}{N} + (1 - \frac{1}{N})|2\text{jinc}(\zeta(\phi, \theta))|^2. \tag{41}$$

These equations show that outside of the main beam and the first couple of side beams and nulls, the average normalized intensity stabilizes at $1/N$. So, as the number of participating nodes increases, the average off-beam power level decreases relative to the main beam level. Using MATLAB, simulated square planar random arrays were generated and the transmit patterns were calculated. Figure 3 is a comparison between a portion of the transmit patterns in the azimuth plane for four different distributed arrays with different numbers of nodes. The four arrays contain $N = 16, 64, 256$, and $1,024$ nodes, all distributed over an area of $100\lambda \times 100\lambda$. The beam target is $\phi_0 = 0$.

From Fig. 3, it is apparent that although the sidelobe levels vary with different numbers of collaborating nodes, the width of the main beam is the same across the four different arrays. The beamwidth is determined by the size of the array aperture, which, in this case, is the dimensions of the area over which the nodes are distributed. Since the main beam has the same average width as the sidelobes, the array area or volume size also determines the "feature density" of the transmit pattern, or the number of peaks and nulls per arc length or solid angle.

Expressions for determining the average peak and null positions of the transmit pattern are provided in [29]. For a random planar distributed array with an array area defined by a circle of radius R, the approximate azimuth angles of the peaks and nulls can be found using

$$\phi_{q,p}^{peak} \sim 2\arcsin\left(\frac{q + 3/4}{4R}\right), q = 1, 2, ... \tag{42}$$

$$\phi_{q,p}^{null} \sim 2\arcsin\left(\frac{q + 1/4}{4R}\right), q = 1, 2, ... \tag{43}$$

The average first null beamwidth (FNBW) in the azimuthal plane can be found by finding the difference in ϕ_q^{null} for successive values of q, which yields

$$E\left[\Phi_{FNBW}\right] = 2\arcsin\left(\frac{1}{4R}\right). \tag{44}$$

For a distributed array area or volume defined using a square area or cubic volume with side dimension L_t, then the substitution $L_t \approx 2R$ can be made.

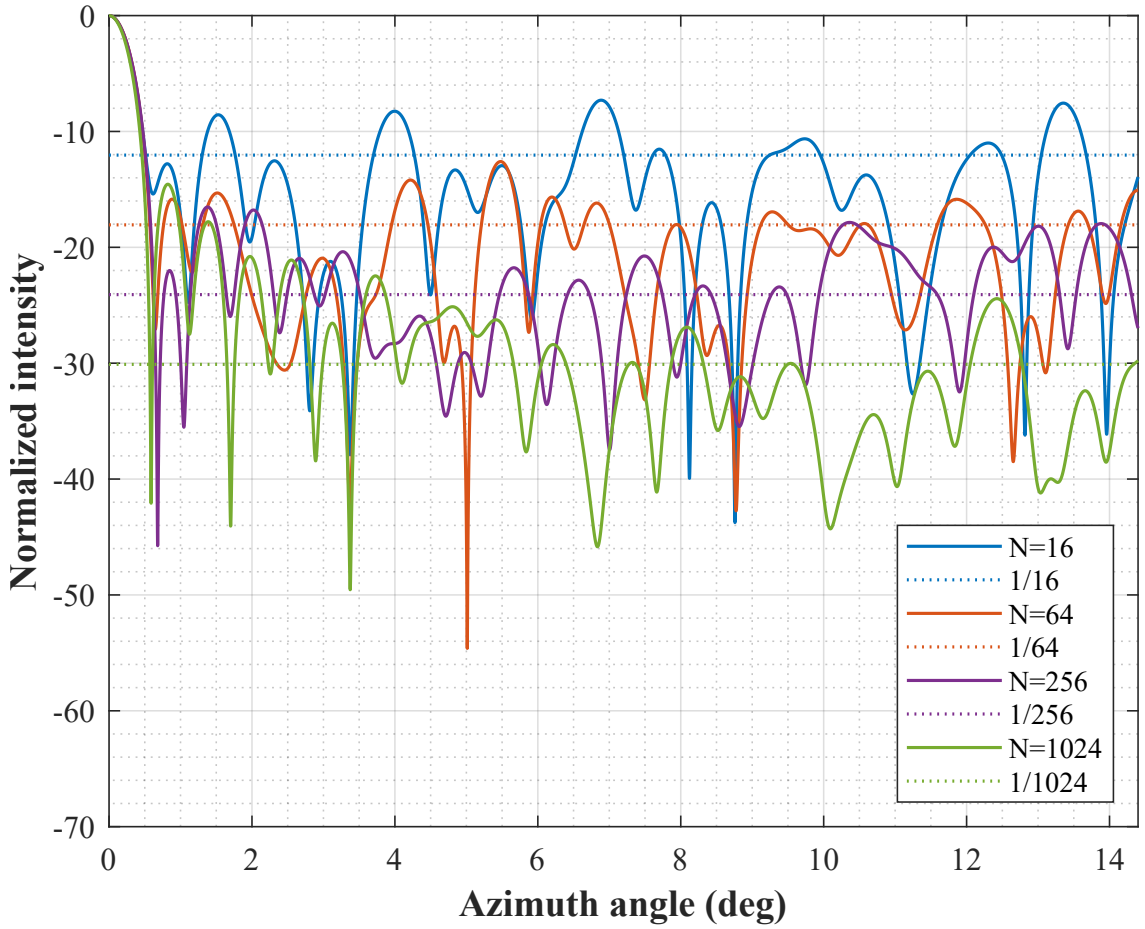

Fig. 3—Portion of normalized far-field power patterns for distributed transmit arrays using different numbers of nodes, along with level lines showing corresponding average power levels; $L_t = 100, \phi_0 = 0$

MATLAB is used to generate random distributed arrays and to calculate their transmit patterns. Once a distributed array is simulated, the peaks and nulls of the transmit pattern can be found. This is done by iterating over the range of azimuth values at a sufficient resolution and comparing successive values of $U(\phi)$ and $\frac{\partial}{\partial\phi}U(\phi)$. Figure 4 shows a portion of the azimuth pattern of a square planar distributed array with $N = 16$ nodes and $L_t = 100$ wavelengths.

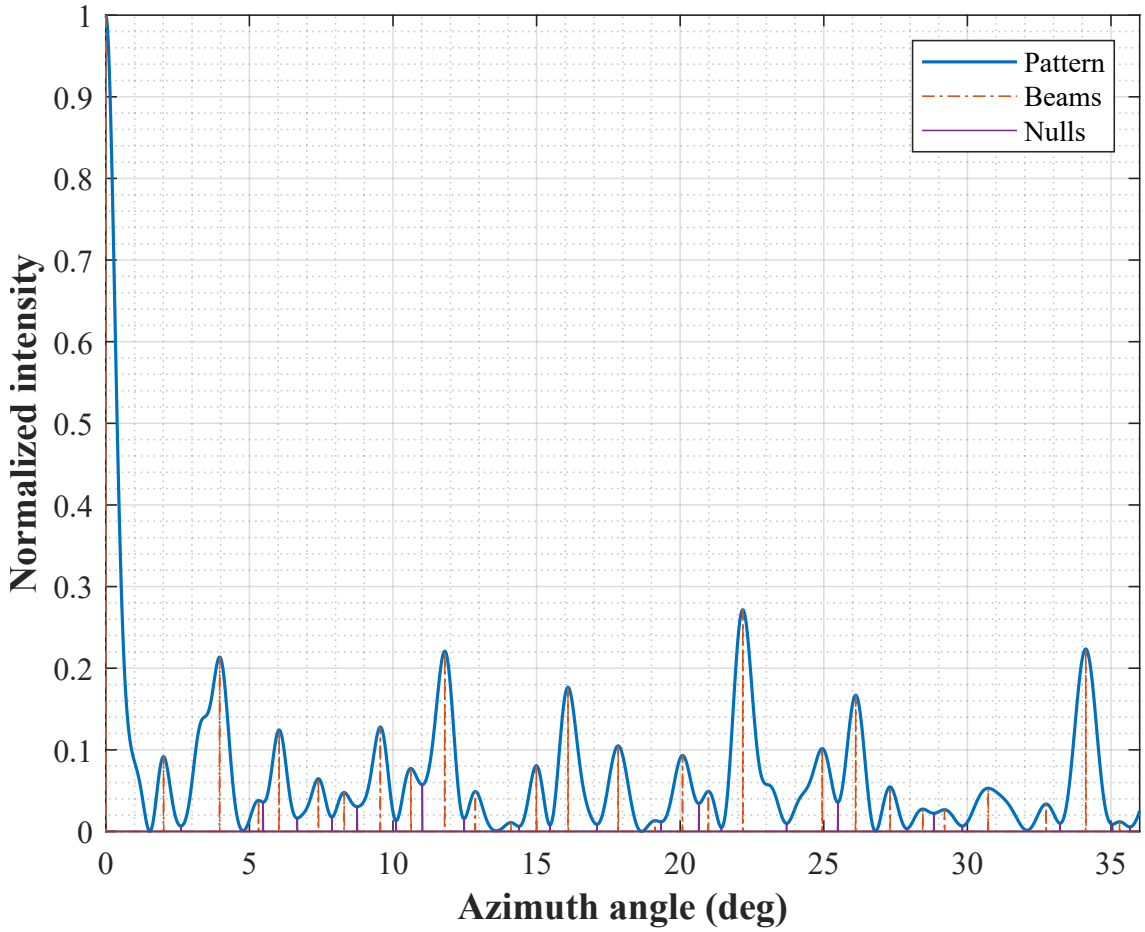

Fig. 4—Portion of normalized far-field power pattern of distributed transmit array; $N = 16$, $L_t = 100$, $\phi_0 = 0$. The peaks and nulls are indicated.

Using the arrays from Fig. 3, the angular widths between each pair of successive nulls is found, and the resulting distributions of FNBW values are shown in Fig. 5. The distribution of FNBW values is consistent across values of N. If R is approximated to be $L_t/2$, then from Eq. (44), the approximate FNBW should be $\sim 0.573°$. The distributions from Fig. 5 show that the median value of FNBW is about 0.8°.

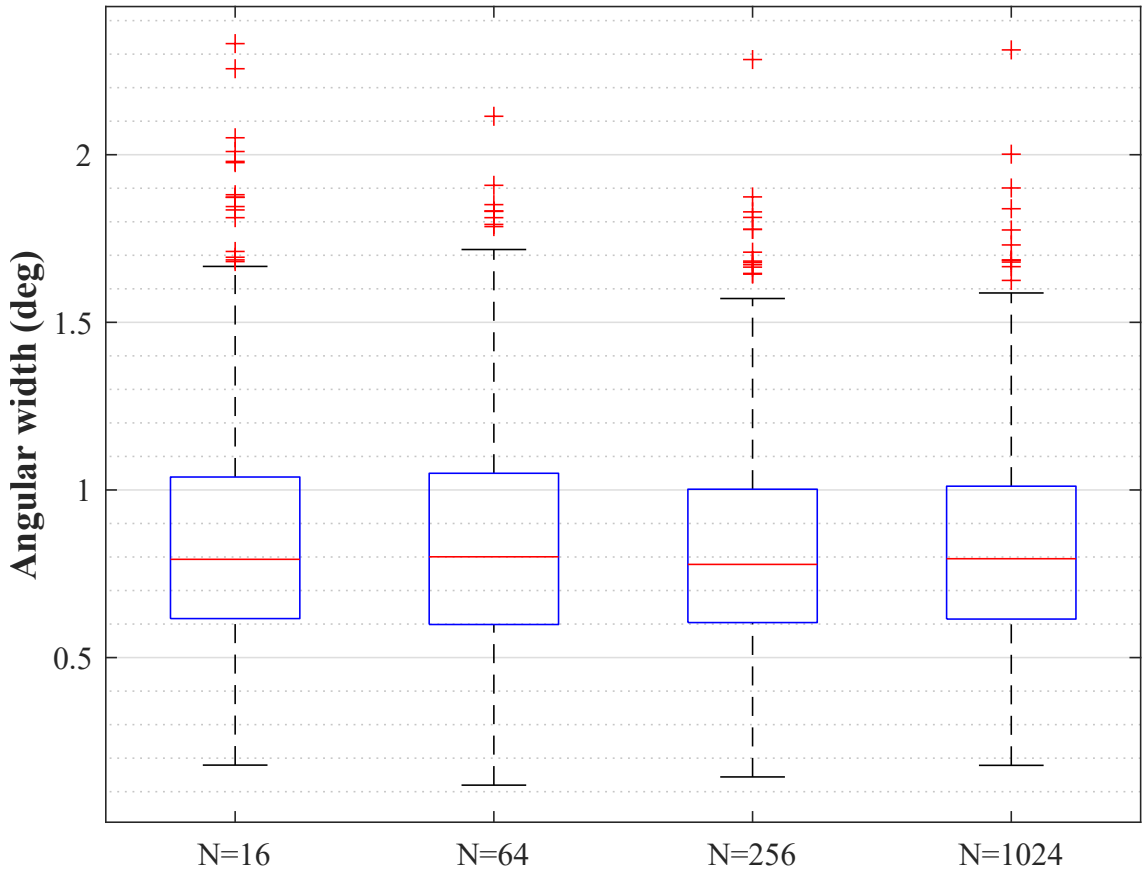

Fig. 5—Distributions of angular width between successive nulls, or FNBW, for planar distributed arrays with different numbers of nodes; $L_t = 100, \phi_0 = 0$

4.2 Volumetric Transmit Pattern

For the cubic volumetric array whereby the array volume is defined by a cube with side length L_t, the average beampattern is given by

$$\xi(\theta) = 2\pi L_t(\cos\theta - \cos\theta_0) \tag{45}$$

$$U_{av}(\phi,\theta) = \frac{1}{N} + (1 - \frac{1}{N})|\text{sinc}(\alpha(\phi,\theta))|^2|\text{sinc}(\beta(\phi,\theta))|^2|\text{sinc}(\xi(\theta))|^2. \tag{46}$$

For a random volumetric distributed array with an array volume defined by a sphere with radius R, the average beampattern is given by

$$\xi(\theta) = 2\pi R(\cos\theta - \cos\theta_0) \tag{47}$$

$$U_{av}(\phi,\theta) = \frac{1}{N} + (1 - \frac{1}{N})|3\text{tinc}(\zeta(\phi,\theta))|^2|2\text{jinc}(\xi(\theta))|^2, \tag{48}$$

where

$$3\frac{j_1(x)}{x} = 3\text{tinc}(x). \tag{49}$$

Continuing with the spherical random array, the peak and null azimuth angles for $\theta = \theta_0 = \pi/2$ can be estimated using

$$\phi_{q,v}^{peak} \sim 2\arcsin\left(\frac{q+1}{4R}\right), q = 1, 2, ... \tag{50}$$

$$\phi_{q,v}^{null} \sim 2\arcsin\left(\frac{q+1/2}{4R}\right), q = 1, 2, ... \tag{51}$$

Upon examination, it is apparent that the difference between successive values of q is the same as with the circular array, which also means that the FNBW in the azimuth pattern of the spherical array is the same as that of the circular array. So, the FNBW estimation given by Eq. (44) also applies to the volumetric arrays. A series of cubic random arrays are generated using MATLAB with $N = 16$ nodes and $L_t = 1k$. The beam is set to $(\phi_0, \theta_0) = (0, \pi/2)$. The transmit pattern is calculated and then is sliced along the azimuth and elevation directions starting from the main beam, which yields a pair of two-dimensional beam patterns. For each pattern, the peaks and nulls are found and the distances between successive nulls are recorded. Figure 6 shows the distributions of the FNBW values along the azimuth and elevation slices. A total of 1,000 values are used in each distribution. The azimuth FNBW distribution has a median value of 0.085° and a mean of 0.090°, and the elevation FNBW distribution has a median value of 0.079° and a mean value of 0.085°.

In order to rectify the discrepancy between the mean of the measured values for FNBW and the average value predicted using Eq. (44), the closed-form expression for calculating average FNBW is adjusted. For the volumetric cubic array with uniform random distribution, the adjusted expression is

$$E\left[\Phi_{FNBW}\right] = 2\arcsin\left(\frac{s_b}{2L_t}\right), \tag{52}$$

where s_b is a constant. The value for s_b is determined empirically. First, many volumetric cubic arrays are simulated and the FNBW of every peak in the pattern of each array is measured. The number of nodes in

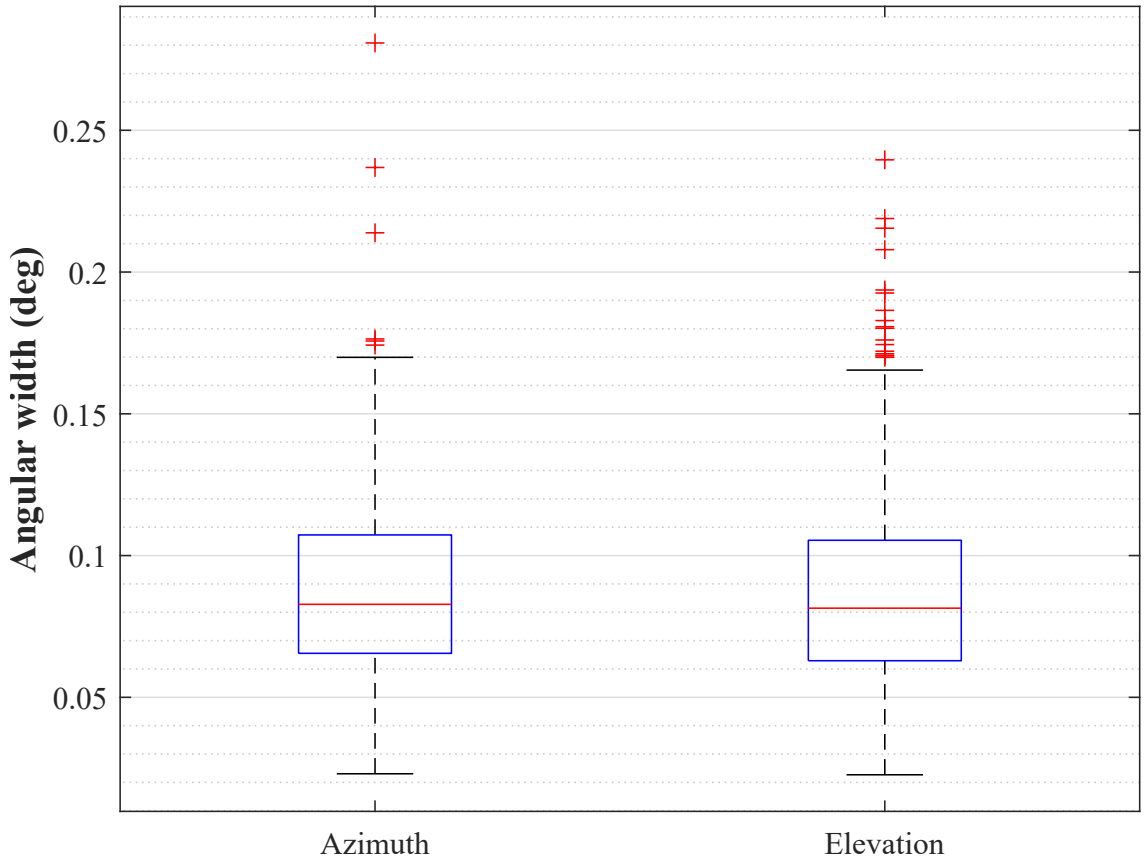

Fig. 6—Distributions of FNBW values for cubic distributed arrays found by slicing transmit pattern along azimuth and elevation directions from the main beam; $N = 16$, $L_t = 1k$, $\phi_0 = 0, \theta_0 = \pi/2$

each array is varied from $N = 16$ all the way up to $N = 1,024$ while the array volume is varied using values of $L_t = 10$ up to $L_t = 10k$. Then, the value of s_b is estimated from the measured FNBW values using

$$s_b \approx \mathrm{E}\left[2L_t \sin\left(\frac{\Phi_{FNBW}}{2}\right)\right]. \tag{53}$$

Figure 7 is a box plot showing the distributions of s_b values calculated from the FNBW values of simulated arrays with $N = 16$. The s_b distributions are separated by value of L_t, showing that it is independent of array size. All of the calculated values for s_b across all of the values of N and L_t are concatenated and binned. A histogram of s_b values with 60 bins is shown in Fig. 8 with a gamma distribution that has been fit to the data. The fitted gamma distribution has a shape parameter of $\alpha = 7.03$ and a rate parameter of $\beta = 4.60$. The 95% confidence intervals for α and β are $[6.89, 7.18]$ and $[4.50, 4.70]$, respectively. The mean value of the fitted gamma distribution is $E[s_b] = \alpha/\beta = 1.53$. This value can be plugged into Eq. (52) to get an accurate estimate of the FNBW of a square planar array or a volumetric cubic array with side length L_t and uniformly distributed nodes.

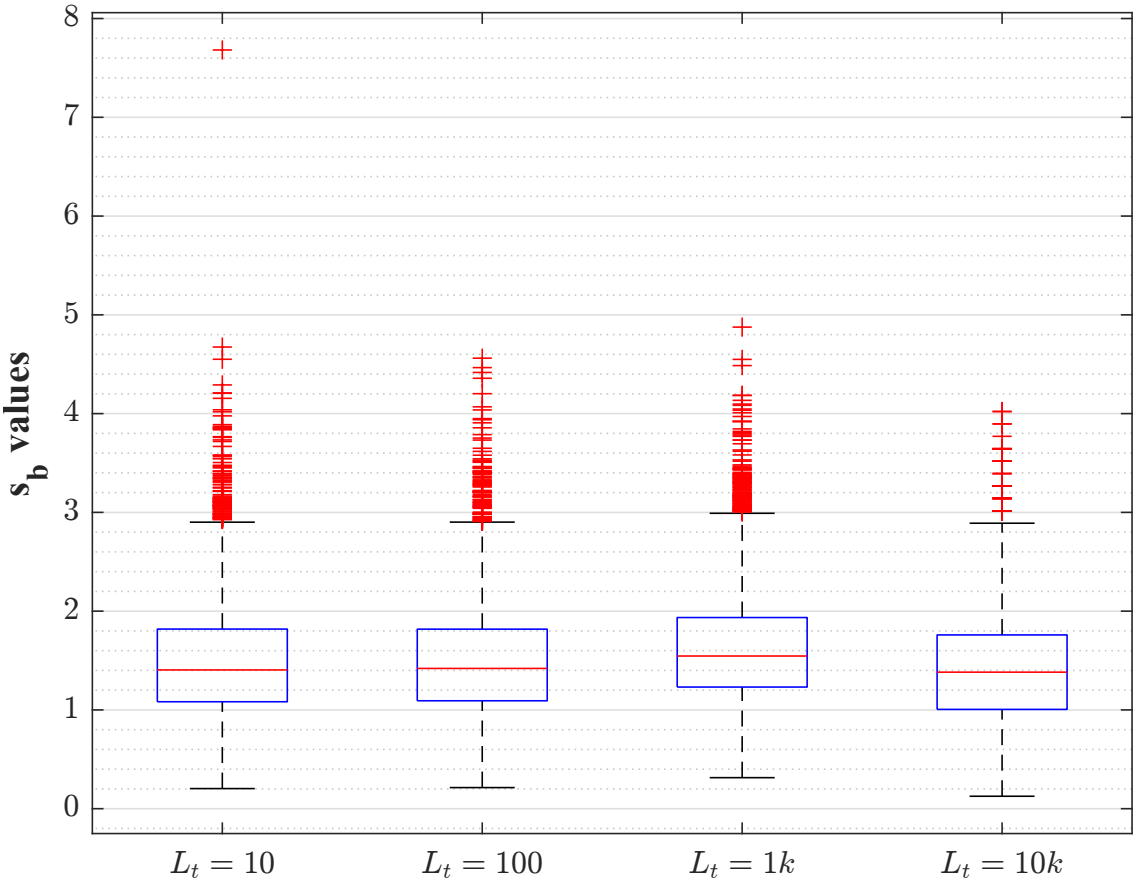

Fig. 7—Distributions of s_b values for different array sizes, calculated using FNBW values generated from cubic distributed arrays; $N = 16$

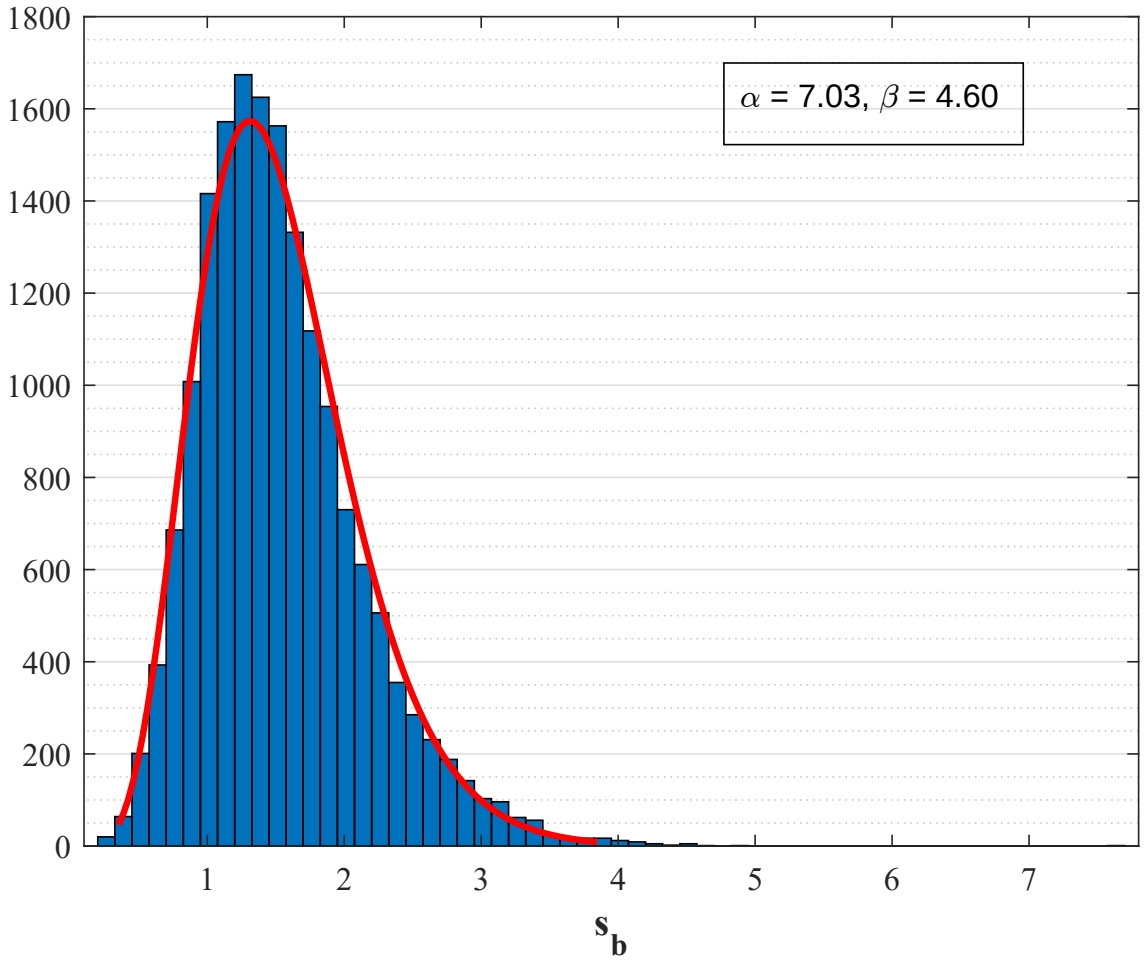

Fig. 8—Histogram of s_b values with a fitted gamma distribution

4.3 Periodic Arrays

In this section, periodic volumetric arrays are simulated and their transmit patterns are calculated. The cubic array volume with side length L_t is used in order to provide a close comparison with the aperiodic cubic arrays.

First, a periodic cubic array is generated with $N = 27$ nodes arranged in an evenly spaced cubic formation of $3 \times 3 \times 3$ nodes. The array volume side length is $L_t = 10$, and the nodes are arranged so that the array fills the entire volume. The beam target for the periodic array is set to $(\phi_0, \theta_0) = (0, \pi/2)$ to yield a main beam in the azimuth plane. Figure 9 shows the node configuration in three dimensions. Figure 10 shows the azimuth cut of the transmit pattern of the periodic array. This plot highlights the differences between the patterns of an aperiodic distributed array versus a periodic distributed array. For a periodic array, because the internode distances are greater than one-half wavelength, the pattern will have grating lobes. Therefore, not only is the pattern periodic, but it will have many strong sidelobes that are at are near maximum intensity. Figure 11 shows the full three-dimensional transmit pattern, which features periodicity in azimuth and elevation.

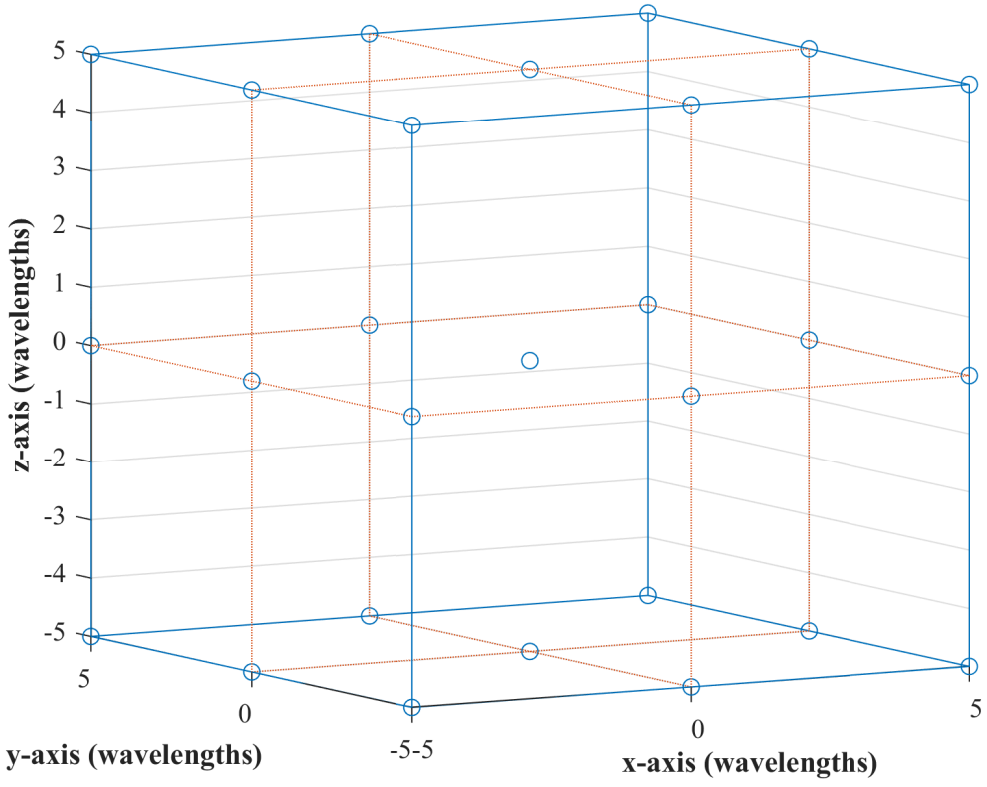

Fig. 9—Node arrangement for a periodic volumetric distributed array with $N = 27$ nodes arranged in a cubic grid with dimension $L_t = 10$

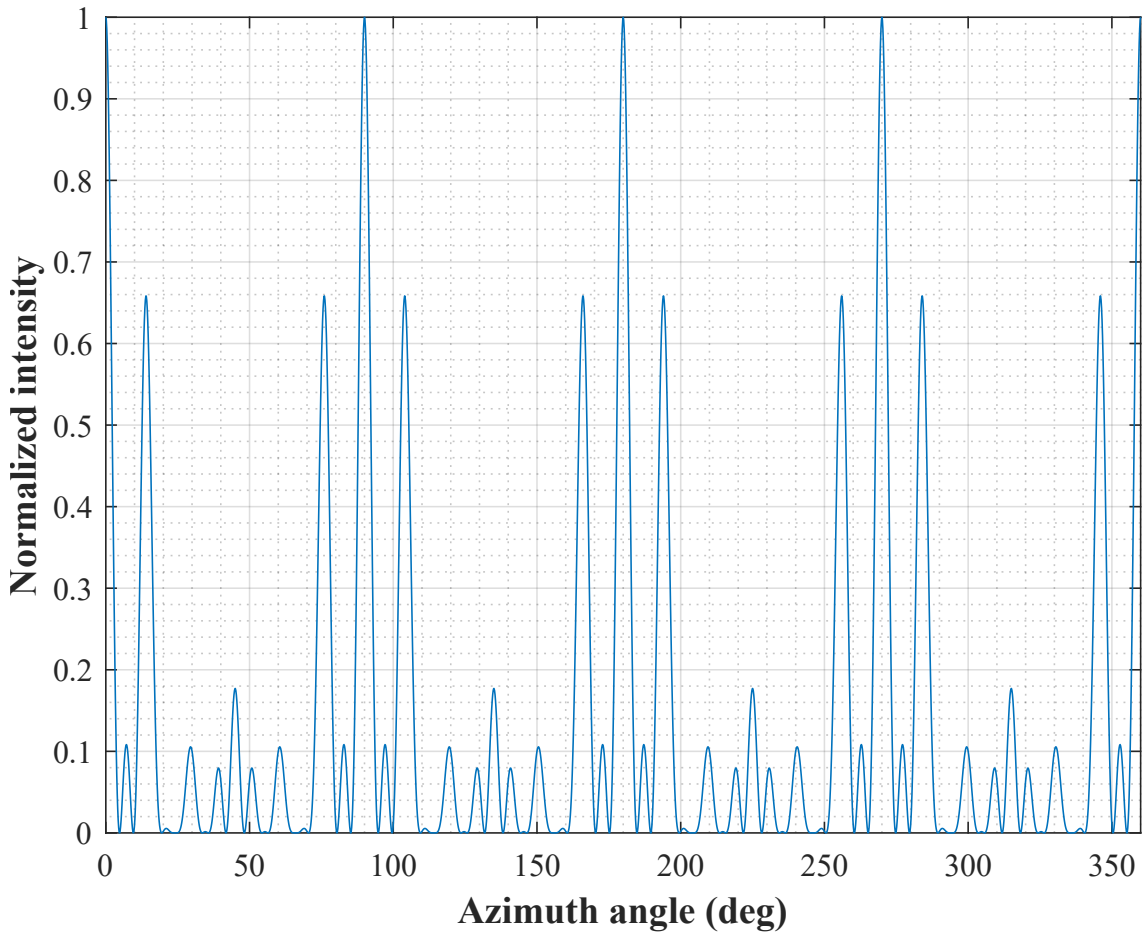

Fig. 10—Azimuth cut of the transmit pattern of periodic volumetric distributed array with $N = 27$ nodes arranged in a cubic grid with dimension $L_t = 10$ and main beam at $(\phi_0, \theta_0) = (0, \pi/2)$

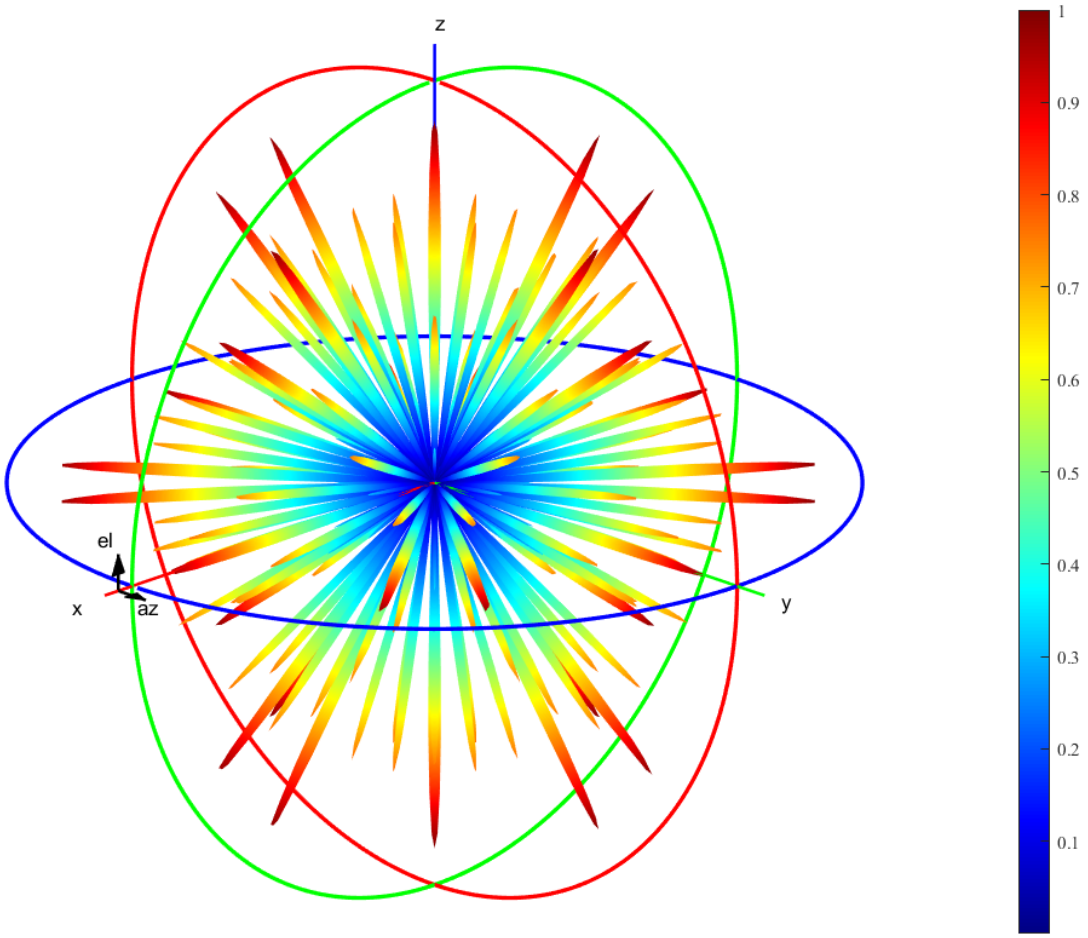

Fig. 11—Three-dimensional transmit pattern of periodic volumetric distributed array with $N = 27$ nodes arranged in a cubic grid with dimension $L_t = 10$ and main beam at $(\phi_0, \theta_0) = (0, \pi/2)$.

Next, simulations were run to ascertain distributions for peak width of the periodic volumetric array transmit patterns. Maximum array volume was increased to $L_t = 100$. Also, additional periodic arrays were generated by adjusting the internode spacing so that the array volume dimension was shrunk to $L_t = 80$ and $L_t = 67$. Then, similarly to what was done in the previous section, for each pattern, the azimuth cut pattern is calculated and the null-to-null width is calculated for each peak in the pattern cut. The values are binned and displayed in Fig. 12, which shows the relative incidence of different peak widths for $3 \times 3 \times 3$ periodic cubic arrays with different internode spacing, along with relative peak width incidence for a sample aperiodic distributed array with $L_t \approx 100$. The approximate is used with the aperiodic because the cube with side length $L_t = 100$ is used to bound the volume within which a node can be located. The peak widths of the aperiodic arrays tend to have less variance than the periodic arrays, and it is apparent that the peak width is increasing for the periodic arrays as the internode spacing decreases.

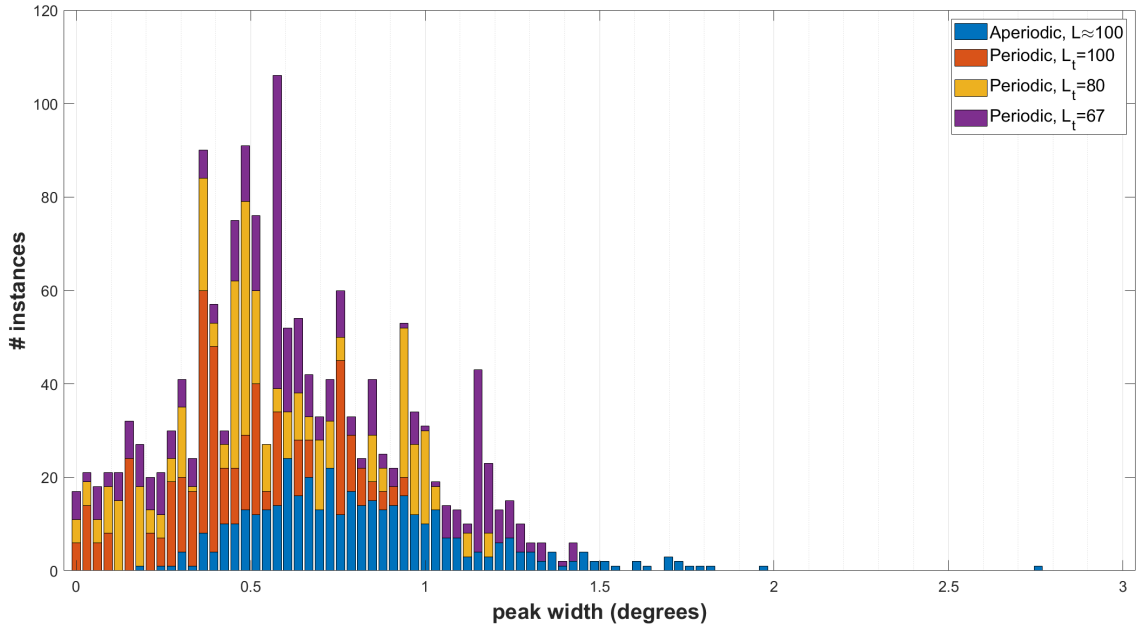

Fig. 12—Histogram showing relative incidence of peak widths in degrees for periodic vs. aperiodic distributed arrays with $N = 27$ nodes and L_t around 100

Then, for $N = 27$ periodic volumetric arrays with cube side lengths of $L_t \in \{100, 80, 75, 67\}$, the normalized intensity values of the azimuth pattern are sampled at a high angular resolution and are placed into one of 20 bins between 0 and 1. This procedure is performed for a representative aperiodic cubic array with $L_t \approx 100$. Figure 13 is the resulting histogram showing relative incidence of the different ranges of normalized intensity for the four different sizes of periodic cubic array compared against the aperiodic cubic array. The intensity values for the aperiodic array azimuth pattern samples drop quickly, and there are no values after the first five bins, whereas the periodic cubic arrays have intensity values that span all bins.

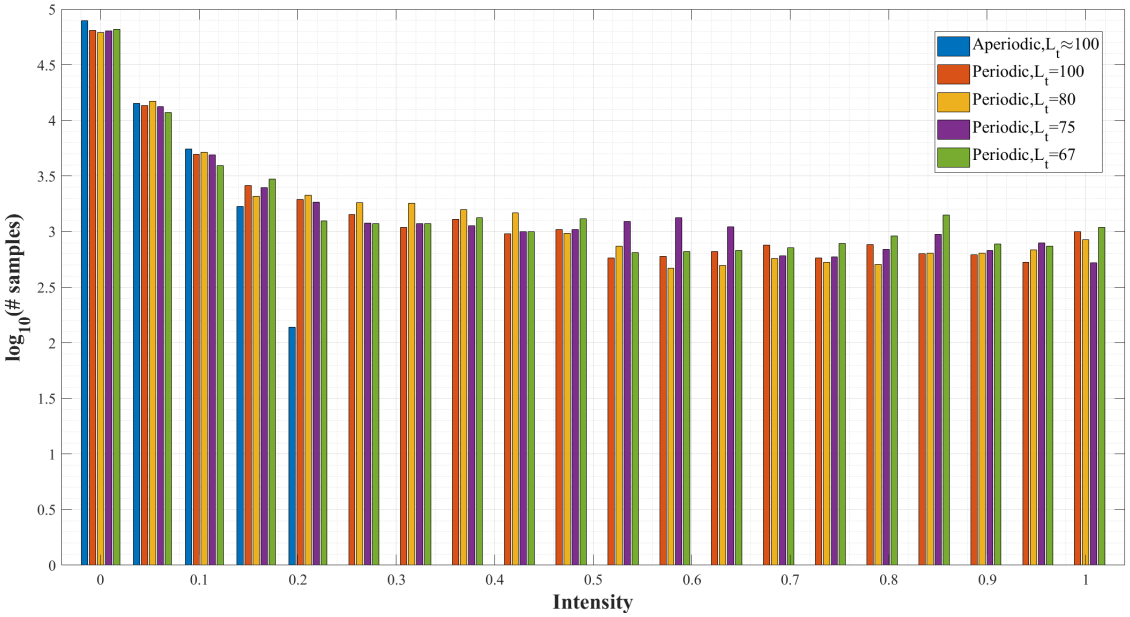

Fig. 13—Histogram showing relative incidence of normalized intensity values for periodic vs. aperiodic distributed arrays with $N = 27$ nodes and L_t around 100.

This procedure is repeated for periodic cubic arrays of $N = 64$ nodes arranged in a $4 \times 4 \times 4$ grid structure with even spacing between nodes. The cube dimension of the periodic array was set to $L_t \in \{75, 80, 100\}$. Figure 14 is the histogram showing number of occurrences of different peak widths for the periodic cubic arrays of different volumes compared with an aperiodic array of similar volume. Azimuth patterns are calculated for $N = 64$, $L_t \in \{67, 75, 80, 100\}$ and normalized intensity values are placed into 20 bins between 0 and 1. Figure 15 is the histogram showing relative incidence of normalized intensity values in the azimuth pattern for the four different sized $N = 64$ periodic cubic arrays, compared against an aperiodic cubic array with $N = 64$ nodes and $L_t \approx 100$.

The regular node spacing of the periodic cubic distributed arrays leads to strong constructive interference in multiple unintended directions. Due to the irregular spacing of the aperiodic distributed arrays, the likelihood of strong constructive interference in an off-beam direction is low, so the average off-beam intensity is much lower. However, the patterns of aperiodic distributed arrays can still feature strong side beams.

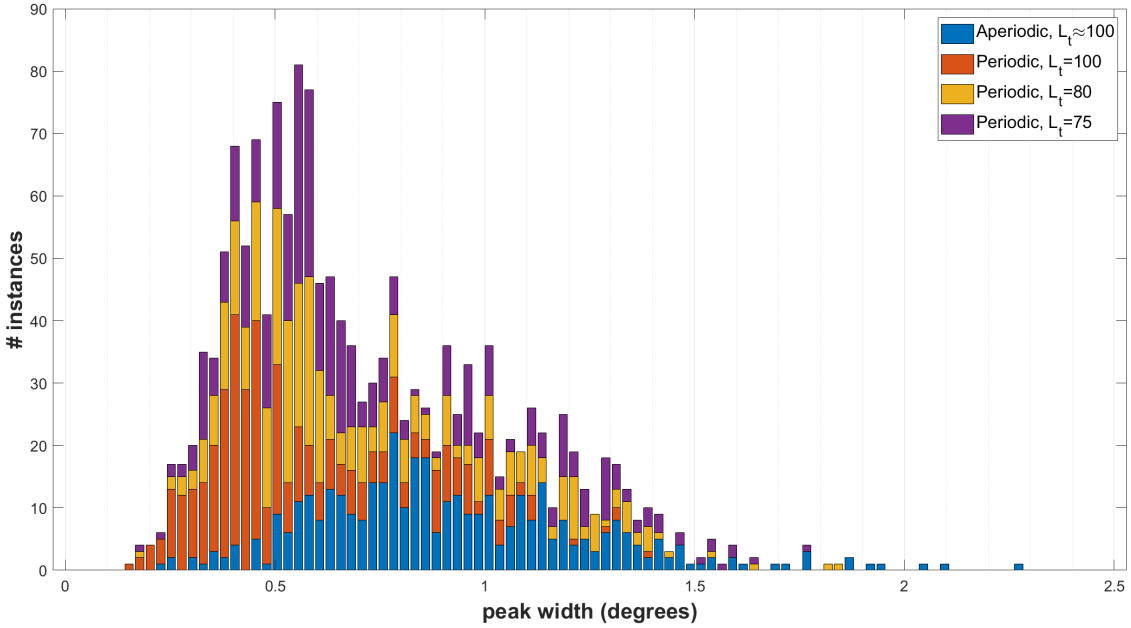

Fig. 14—Histogram showing relative incidence of peak widths in degrees for periodic vs. aperiodic distributed arrays with $N = 64$ nodes and L_t around 100

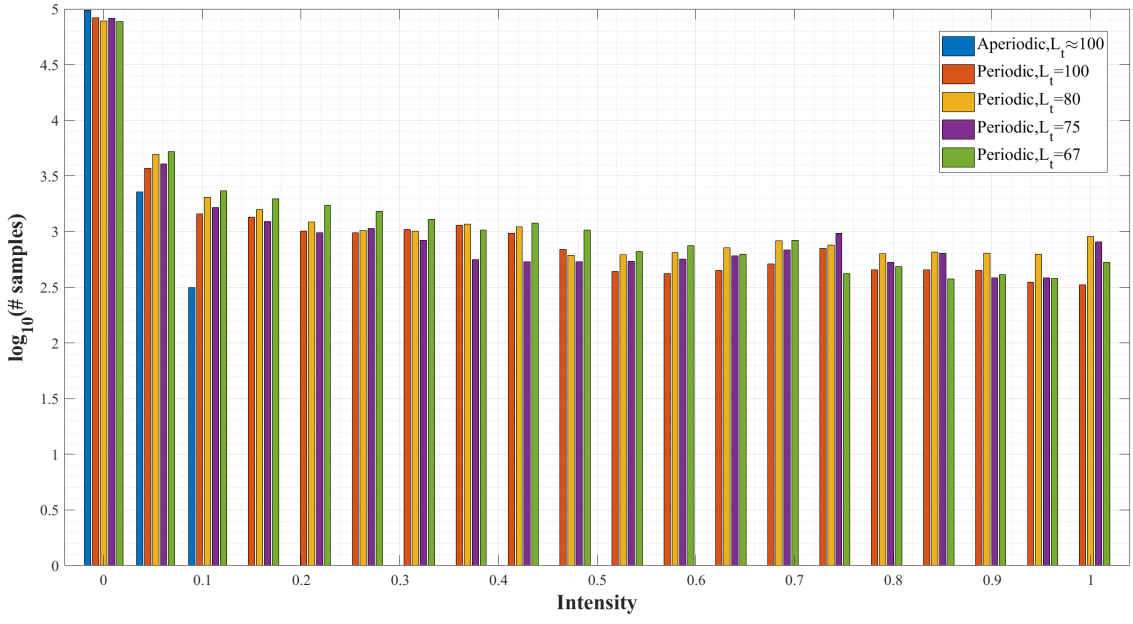

Fig. 15—Histogram showing relative incidence of normalized intensity values for periodic vs. aperiodic distributed arrays with $N = 64$ nodes and L_t around 100

5. OPTIMIZATION PROBLEM

Consider a coherent communication system defined as per the system model, consisting of a distributed transmitter array of N homogeneous nodes spread over a square area of dimensions $L_t \times L_t$ and a receiver array of M homogeneous nodes spread over a square area of dimensions $L_r \times L_r$, with a distance of D between the two arrays. The theoretical upper bound for the coherent communication gain of such a system is $G_{UB} = N^2M$, assuming omnidirectional antennas on all network nodes with individual gains capped at 1. However, a practical system will have a gain much lower than G_{UB}. In a realistic deployment scenario, there will likely be constraints on the positioning of network nodes. Geographic considerations may restrict the movement options for some or all nodes, and programmatic or hardware specifics will put limitations on the spacing between nodes; after all, the forming of ad hoc coherent distributed arrays is an ancillary function that is meant to enhance and expand network functionality.

Now, consider the problem of maximizing the coherent communication gain of a system. Without directional antennas, the main variables under consideration are the positions of all of the transmitter and receiver nodes and the beam angle of the transmitter array. This global optimization problem can be stated as

$$\begin{aligned} \max \quad & \tilde{G}(\mathbf{B}, \phi_0, \theta_0) \\ \text{s.t.} \quad & |x_n^t|, |y_n^t|, |z_n^t| \leq 0.5L_t, \\ & 0.5L_t + D \leq x_m^r \leq 0.5L_t + D + L_r, \\ & |y_m^r|, |z_m^r| \leq 0.5L_r, \\ & n \in \{1, \ldots, N\}; m \in \{1, \ldots, M\}, \\ & \phi_{m,min} \leq \phi_0 \leq \phi_{m,max} \\ & \theta_{m,min} \leq \theta_0 \leq \theta_{m,max}. \end{aligned} \tag{54}$$

5.1 Problem Analysis

For any nontrivial system, $G(\mathbf{B}, \phi_0, \theta_0)$ will have many local extrema over the constrained search space for $\mathbf{B}$, ϕ_0, and θ_0. To demonstrate, an example coherent communication system is simulated with $N = 16$, $M = 16$ nodes that are randomly placed over their respective array volumes, with dimensions given by $L_t = L_r = 100$ wavelengths, using a uniform random distribution. Interarray distance D is 1,000 wavelengths, the Tx array beam angle ϕ_0 is fixed at 0, and the elevation angle θ_0 is fixed at $\pi/2$. Tx node s_1^t is allowed to move while all other nodes remain stationary. Figure 16 is a plot of G versus (y_1^t, z_1^t) as s_1^t moves over a plane within the Tx array volume where x_i^t is fixed. Multiple extrema are visible in one small region of the objective function, even when it is reduced to only two variables.

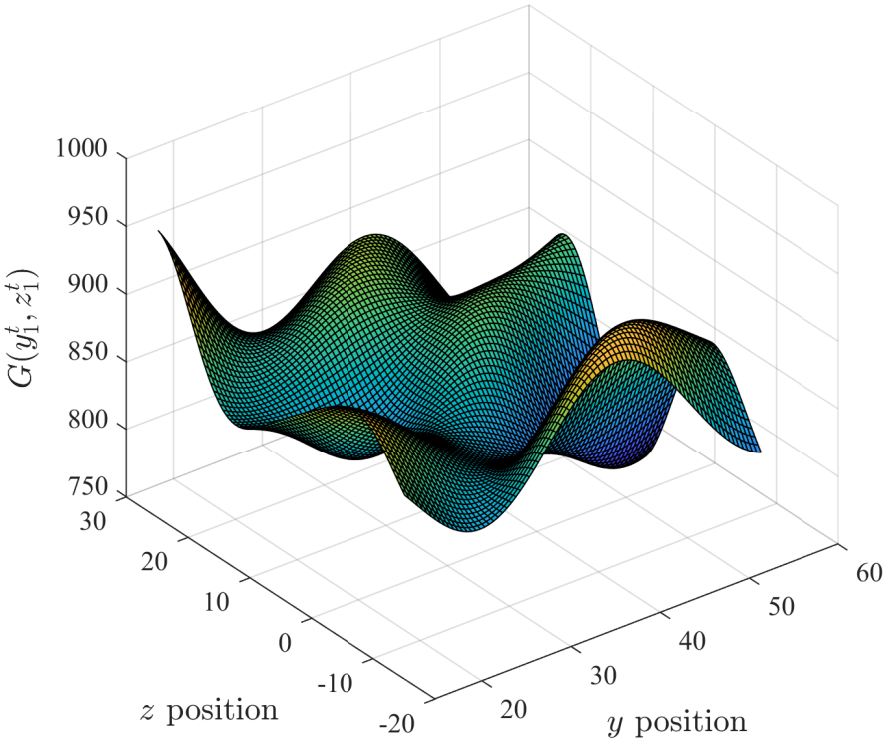

Fig. 16—Plot of coherent communication gain as a function of the y and z positions of a single Tx node, with all other system nodes' positions fixed. For example system, $N = 16, M = 16, L_t = L_r = 100, D = 1,000$.

The global optimization problem given by Eq. (54) consists of a nonlinear, nonconvex objective function with convex constraints. This problem is shown to be NP-hard via a straightforward reduction of the SUBSET-SUM problem. Take $\mathbf{a} = \{a_1, \ldots, a_n\}$ to be the set of input integers, and let $\mathbf{x} = \{x_1, \ldots, x_n\}$ be the variables encoding the subset. The nonconvex program is formulated as

$$\begin{aligned} \min \quad & (\mathbf{a} \cdot \mathbf{x}) + \sum_{i=1}^{n} x_i^2 (1 - x_i)^2 \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i \geq 1. \end{aligned} \tag{55}$$

The global optimization problem is known to be NP-hard, but what sort of growth rate would be expected when attempting to find a solution via exhaustive search? For simplicity, the fundamental operation is defined as the evaluation of $G(\mathbf{B}, \phi_0, \theta_0)$ for a certain configuration given by $\mathbf{B}, \phi_0, \theta_0$. The node position variables contained in $\mathbf{B}$ and the Tx array beam angles ϕ_0, θ_0 are then converted into discrete variables. Define t_p as the

linear spatial resolution for node placement, or the interval between consecutive values in x_n^t, y_n^t, z_n^t, x_m^r, y_m^r, and z_m^r. Then, the number of possible positions for a transmitter node is given by $K_t = (\frac{L_t}{t_p})^3$, and the number of possible positions for a receiver node is given by $K_r = (\frac{L_r}{t_p})^3$. In the general case, each node in the system is assumed to be unique, so the number of possible configurations of N transmitter nodes is calculated as the number of permutations without replacement, or $K_{perm}^t = \frac{K_t!}{(K_t-N)!}$. Similarly, the number of possible configurations of M receiver nodes is calculated as $K_{perm}^r = \frac{K_r!}{(K_r-M)!}$.

For each permutation of node positions, the range of Tx beam angles needs to be tested. Define $\delta\phi_M$ as the azimuth angular span to be searched with azimuth angular resolution of t_{az}, and define $\delta\theta_M$ as the span of viable elevation angles with elevation angular resolution t_{el}. Thus, $K_{az} = \frac{\delta\phi_M}{t_{az}}$ is the total number of possible azimuth angles and $K_{el} = \frac{\delta\theta_M}{t_{el}}$. Thus, the total number of $G(\mathbf{B}, \phi_0, \theta_0)$ evaluations required for an exhaustive search, in the general case, is $K_{az} \cdot K_{el} \cdot \frac{K_t!}{(K_t-N)!} \cdot \frac{K_r!}{(K_r-M)!}$. Certain modifiers can reduce this number somewhat, like assuming that the network nodes are all identical. A best-case lower-bound on the complexity is $O\left((\frac{K_t}{N})^N(\frac{K_r}{M})^M\right)$. It is important to consider, however, that in almost all practical scenarios, K_t and K_r are many orders of magnitude greater than N and M. For instance, suppose the example used in the beginning of this section had a spatial sampling of 0.1 wavelengths. Then, $K_t = K_r = (1,000/0.1)^3 = 10^{12}$, compared to $N = M = 16$. This example would require $> 2.94 \times 10^{345}$ evaluations of $G(\mathbf{B}, \phi_0, \theta_0)$ for an exhaustive search of all node configurations of node positions.

5.2 Motivating Gradient Descent

In Section 5.1, it was shown that even a relatively sparse discretization of a nontrivial problem of the form described by Eq. (54) results in an extraordinary number of potential solutions. Keeping the variables continuous opens up additional tractable solution approaches, but the examination in Section 5.1 illuminates an inescapable reality of this problem: there is no way to guarantee the optimality of a solution. Of course, if it were possible to discretize the search space to a sufficient density and to evaluate the objective function for all points, then the global maximum or a point very near the global maximum could be found. Obviously, an exhaustive search of all possible combinations of values for discrete input variables is not possible, which is why many heuristic algorithms exist for finding approximate global solutions. However, reliably finding a close approximation to the global maximum for the problem presented here, which is highly nonconvex with many dimensions, can still be rather costly and time consuming even for a heuristic algorithm.

Consider also that a wireless network may have nodes that are mobile, or nodes that are disadvantaged in terms of computing capability. In such circumstances, the optimization algorithm would need to be reloaded and rerun frequently, and the computation cost of a heuristic that is looking for a close global approximation would be prohibitive. Accordingly, rather than search for a globally optimal configuration for the network nodes, the more tractable problem of local optimization is examined. This is motivated in part by a recognition that the distributed arrays are formed from existing networks. In general, network node positioning is deliberate and sometimes necessary to the proper operation or functionality of the network. As a result, the nodes are not necessarily able to be moved to any arbitrary locations, but their positions may be adjusted slightly in order to gain a greater array advantage.

5.2.1 Defining Gradient

Examination of the general CCG expression given in Eq. (12) shows that $G(\mathbf{B}, \phi_0, \theta_0)$ comprises a sum of products of sine and cosine functions that are Lipschitz continuous functions. The sum of two Lipschitz

functions is a Lipschitz function, as is the product of two bounded Lipschitz functions. Therefore, $G(\mathbf{B}, \phi_0, \theta_0)$ is Lipschitz. Additionally, when the isotropic radiator assumption is applied to the system model, represented in Eq. (17), then G is bounded such that $0 \leq G \leq N^2 M$. Because $G(\mathbf{B}, \phi_0, \theta_0)$ is Lipschitz, then it is also differentiable everywhere, so a gradient can be introduced. Having access to the gradient opens up classes of useful optimization algorithms; in this case, a straightforward first-order gradient descent method is of interest.

In order to use gradient descent, the gradient of $G(\mathbf{B}, \phi_0, \theta_0)$ must first be defined. The gradient represents the rate of change of G with respect to the input variables. To construct a useful definition, first the optimization variables need to be defined. The node positions are not entirely fixed, so they may be used as optimization variables. Additionally, the beam of the transmit array, defined by (ϕ_0, θ_0), can be a variable as well. An open-loop coherent distributed transmit array can direct its beam in an arbitrary direction. Most prior work on coherent distributed arrays focuses on transmit beamforming to a single target. An $N \times 1$ coherent system has a straightforward beamforming objective: target the receiver. In an $N \times M$ coherent system, the beam target is not always obvious. When the distance between transmitter and receiver arrays is relatively large, $D \gg L_t$ and $D \gg L_r$, the receiver array appears as a point target to the transmitter array. So, the transmitter array can target the receiver array by targeting any one of the receiver nodes and all of the receiver nodes end up on or near the beam apex, or $(\phi_m, \theta_m) \approx (\phi_0, \theta_0)$, $1 \leq m \leq M$.

Accounting for all of the input variables yields (x_n^t, y_n^t, z_n^t) values for each transmitter node, (x_m^r, y_m^r, z_m^r) values for each receiver node, and (ϕ_0, θ_0) values defining the transmitter array beam angle. Thus, there are $3(N + M) + 2$ total variables, and the resulting gradient expression is

$$\begin{aligned}\nabla G =& \hat{x}_1^t \frac{\partial}{\partial x_1^t} G + \hat{y}_1^t \frac{\partial}{\partial y_1^t} G + \hat{z}_1^t \frac{\partial}{\partial z_1^t} G + \ldots + \hat{x}_N^t \frac{\partial}{\partial x_N^t} G + \hat{y}_N^t \frac{\partial}{\partial y_N^t} G + \hat{z}_N^t \frac{\partial}{\partial z_N^t} G \\ &+ \hat{x}_1^r \frac{\partial}{\partial x_1^r} G + \hat{y}_1^r \frac{\partial}{\partial y_1^r} G + \hat{z}_1^r \frac{\partial}{\partial z_1^r} G + \ldots + \hat{x}_M^r \frac{\partial}{\partial x_M^r} G + \hat{y}_M^r \frac{\partial}{\partial y_M^r} G + \hat{z}_M^r \frac{\partial}{\partial z_M^r} G \\ &+ \hat{\phi}_0 \frac{\partial}{\partial \phi_0} G + \hat{\theta}_0 \frac{\partial}{\partial \theta_0} G. \end{aligned} \tag{56}$$

Closed-form expressions for each term in Eq. (56) are derived in Chapter 6.

5.3 Steepest Descent Approach

Using the closed-form expressions derived in Chapter 6, first-order gradient methods can be employed to find locally optimal network configurations that deliver an increase in CCG over the initial configuration. In this section, the main optimization approach is introduced. The main approach comprises two stages. First, the initial transmitter array beam angles (ϕ_0, θ_0) are calculated by solving a simple two-dimensional optimization problem. Next, a gradient descent algorithm is employed to find the locally optimal positioning of the distributed array nodes and transmitter array beam angle. The optimization approach assumes an initial positioning of nodes is given.

5.4 Finding Initial Beam Angle

In general, the transmitter array is open loop, which means that it can direct the main beam of its array pattern toward an arbitrary direction defined by the combination of azimuth angle $\phi_0 \in [0, 2\pi]$ and elevation angle $\theta_0 \in [0, \pi]$ by adjusting the phase offsets of the individual nodes. Each node in the receiver array can be described by its azimuth and elevation angles relative to the transmitter array, and these angles are given by the vectors $\Phi_M = \{\phi_m | m = 1, ..., M\}$ and $\Theta_M = \{\theta_m | m = 1, ..., M\}$. The azimuth width of the receiver array relative to the transmitter array is defined as $\delta\phi_M = \phi_{m,max} - \phi_{m,min}$, where $\phi_{m,min} = \min\{\Phi_M\}$ and $\phi_{m,max} = \max\{\Phi_M\}$, and the elevation width as $\delta\theta_M = \theta_{m,max} - \theta_{m,min}$, where $\theta_{m,min} = \min\{\Theta_M\}$ and $\theta_{m,max} = \max\{\Theta_M\}$.

Combining transmit and receive distributed arrays into a coherent communication system is a concept that has not been well explored. Typically, the focus is on transmit-side distributed array beamforming, so in literature, there is not much in the way of discussion about transmitter array beam angle selection. This optimization problem can be stated as

$$\begin{aligned} \max \quad & G(\phi_0, \theta_0) \\ \text{s.t.} \quad & \phi_{m,min} \leq \phi_0 \leq \phi_{m,max}, \\ & \theta_{m,min} \leq \theta_0 \leq \theta_{m,max}, \end{aligned} \tag{57}$$

where the node positions given by $\mathbf{B}$ are fixed as constants. When there is only a single receiver, there is only one viable value for azimuth and elevation angles for the transmitter array beam in order to maximize gain. Similarly, when the individual receiver is replaced with an array of multiple receivers, there may be scenarios in which the interarray distance is at least several orders of magnitude greater than the the array dimensions ($D \gg L_t, L_r$). In such scenarios, $\delta\phi_m, \delta\theta_m$ approach 0, making the entire receiver array appear as a point target to the transmitter array. As the ratio between D and L_t, L_r begins to shrink, the receiver array ceases to appear as a point target and begins to occupy greater and greater angular width from the perspective of the transmitter array. Under certain configurations of array parameters, it is possible for some receivers to be fully illuminated by the main beam while others are only partially illuminated by the main beam or fall outside of the main beam entirely. Those receivers that fall outside of the main beam can potentially be illuminated by sidelobes.

When D and L_t are small relative to L_r, the received power can vary considerably between individual receivers in the array. Also, it is important to remember that the distributed coherent array is subject to the thinned array curse: specifically, the typical spacing between elements is relatively large, at least tens or hundreds of wavelengths, which means the transmitter array's first null beamwidth (FNBW) in both azimuth and elevation planes, given by Φ_{FNBW} and Θ_{FNBW}, respectively, will be quite small. As the ratios $\delta\phi_m : \Phi_{FNBW}$ and $\delta\theta_m : \Theta_{FNBW}$ increase, the function $G(\mathbf{B}, \phi_0, \theta_0)$ becomes more "feature dense" in the sense that it acquires more local extrema; also, the range of gain values tends to increase.

The problem stated in Eq. (57) is a simple constrained bivariate optimization problem featuring an objective function that is generally nonconvex. Any number of bivariate optimization methods can be used to solve for the optimal ϕ_0, θ_0 values, which serve as initial conditions for the main optimization algorithm detailed in the next subsection.

5.5 Distributed Array SQSD Algorithm

The main optimization approach is a gradient descent algorithm that improves by finding locally optimal values of $\mathbf{B}$, ϕ_0, and θ_0. Again, given the NP-hardness of the problem stated in Eq. (54), attempting to find the provably globally optimal solution would be infeasible. Additionally, some observations and assumptions are made about the system and the deployment scenario that further influence the algorithmic choices:

1. The initial or starting configuration of node placement, although arbitrary for the sake of the system model and simulations, is significant and intentional in the context of the deployment scenario. In other words, node positioning is important to the mission of the network, but there is some leeway in terms of exact placement. Furthermore, there is a cost associated with moving a node to a new position, in terms of both time and energy. Therefore, the repositioning of nodes should not be arbitrary, and the increase in system gain should be weighed against total node displacement.
2. There is no a priori knowledge that $G(\mathbf{B}, \phi_0, \theta_0)$ is unimodal over some interval x_i, y_i, or $z_i \in [a, b]$, so a simple line search descent is not suitable. Instead, an open-ended, unconstrained gradient descent is desired, most preferably with a step size that is adjusted automatically.
3. The gradient contours for a given node configuration are likely to be poorly scaled, which will result in slow convergence for a standard steepest descent algorithm.
4. The distributed wireless network is resource constrained, so a second-order gradient descent algorithm requiring repeated calculations of the Hessian would be too computationally costly.

Knowing that the objective function is Lipschitz continuous but nonconvex, a gradient descent optimization approach is chosen. Local optimization using gradient descent yields improvements to CCG while being more likely to preserve the general layout of the network.

A simple, gradient-only method that has been shown to be stable and reliable is the spherical quadratic steepest descent (SQSD) method [30, 31]. In SQSD, the steepest descent method is applied to successive spherical quadratic approximations of the objective function. The fundamental assumption behind the method is that many continuously differentiable functions appear quadratic in the close vicinity of local minima. In accordance with the stated observations and assumptions, SQSD does not require computation of the Hessian of the objective function, nor does it include any explicit line searches. Additionally, SQSD features an adjustable step size that is calculated using an estimate of the curvature of the objective function at the evaluation point.

In order to simulate improving system gain using the SQSD method on node positions and beam angle, several heuristic adjustments were made to the original algorithm. For the implementation of SQSD to find the local maximum as a function of beam angles and node positions, it was necessary to handle the gradient components corresponding to beam angles separately from the node position gradient components. To calculate step sizes and curvature estimates, the length of the displacement vector and the gradient magnitude are used. However, there is typically a difference of many orders of magnitude between the scale of azimuth or elevation displacement, in radians, and the node displacement, in wavelengths. This is also true of the rate of change of G with respect to $\mathbf{B}$ versus the rate of change of G with respect to ϕ_0 or θ_0. Because the magnitudes associated with beam angle are generally so much larger, when the beam angle and node

positions are combined into a single vector, the beam angle components end up dominating the overall vector magnitudes for both displacement and gradient. Therefore, the input variables are split into vectors, $\mathbf{B}$ and $\Omega_0 = [\phi_0, \theta_0]$. Keeping $\mathbf{B}$ and ϕ_0, θ_0 components in separate vectors ensures smooth gradient descent across all dimensions without requiring significant alterations to the basic algorithm.

Because SQSD finds the local minimum and the goal is to find the local maximum of $G(\mathbf{B}, \phi_0, \theta_0)$, the obvious solution is to minimize the inverse of the objective function:

$$f(\mathbf{B}, \phi_0, \theta_0) = G_{UB} - G(\mathbf{B}, \phi_0, \theta_0); \tag{58}$$

$$\nabla f(\mathbf{B}, \phi_0, \theta_0) = -\nabla G(\mathbf{B}, \phi_0, \theta_0). \tag{59}$$

The modified SQSD algorithm is labeled *distributed array square quadratic steepest descent* (DA-SQSD) and is used to minimize $f(\mathbf{B}, \phi_0, \theta_0)$, which is equivalent to maximizing $G(\mathbf{B}, \phi_0, \theta_0)$. The steps of DA-SQSD are given in Algorithm 1. The gradient components corresponding to node positions are stored in $\nabla f(\mathbf{B})$ and the gradient component corresponding to beam angle is stored in $\nabla f(\Omega)$. Also, the subscript 0 is omitted from ϕ_0 in the pseudocode to avoid confusion with the iteration number k. It should be noted that the original SQSD algorithm uses threshold tolerances of $||\delta x||$ (per-iteration change in independent variable magnitude) or $||\nabla f(x)||$ (gradient magnitude) as dual termination criteria, whereas a hard cap on the number of iterations is featured in DA-SQSD to reflect the conditions used during simulation. Because the objective is overall improvement in system gain rather than a globally optimal configuration, a hard iteration limit is more sensible in the context of a resource-constrained wireless network. Other algorithm variants could include total improvement in system gain or total node displacement in their terminating conditions.

The system model assumes an initial network state, which is the arbitrary or naïve placement of all of the network nodes within their respective array volumes and the optimal transmitter array beam angles for that configuration. SQSD takes the initial state as input and assigns it to the starting values of $\mathbf{B}$, ϕ_0, and θ_0. Also given as starting values are the iteration limit and the step-size limit $\rho_{\mathbf{B}}, \rho_{\Omega}$, which define the maximum allowable values for the magnitude of the node displacement vector and the magnitude of the transmitter array beam angle displacement vector, respectively.

Algorithm 1 Distributed Array Square Quadratic Steepest Descent (DA-SQSD)

Input: $\mathbf{B}, \phi_0, N_{iter}, \rho_{\mathbf{B}}, \rho_{\Omega}$

set $\mathbf{B}^0 = \mathbf{B}, \Omega^0 = [\phi_0, \theta_0]$

set $c_{\mathbf{B}}^0 = ||\nabla f(\mathbf{B}^0)||/\rho_{\mathbf{B}}$, $c_{\Omega}^0 = ||\nabla f(\Omega^0)||/\rho_{\Omega}$

set $k = 1$

while $k < N_{iter}$ **do**

set $\mathbf{B}^k = \mathbf{B}^{k-1} - \frac{\nabla f(\mathbf{B}^{k-1})}{c_{\mathbf{B}}^{k-1}}$

if $||\mathbf{B}^k - \mathbf{B}^{k-1}|| > \rho_{\mathbf{B}}$ **then**

set $\mathbf{B}^k = \mathbf{B}^{k-1} - \rho_{\mathbf{B}} \frac{\nabla f(\mathbf{B}^{k-1})}{||\nabla f(\mathbf{B}^{k-1})||}$

end if

set $\Omega^k = \Omega^{k-1} - \frac{\nabla f(\Omega^{k-1})}{c_{\Omega^{k-1}}}$

if $||\Omega^k - \Omega^{k-1}|| > \rho_{\Omega}$ **then**

set $\Omega^k = \Omega^{k-1} - \rho_{\Omega} \frac{\nabla f(\Omega^{k-1})}{||\nabla f(\Omega^{k-1}||}$

end if

set $c_{\mathbf{B}}^k = \frac{2(f(\mathbf{B}^{k-1},\Omega^{k-1}) - f(\mathbf{B}^k,\Omega^k) - \nabla^{\mathrm{T}} f(\mathbf{B}^k)(\mathbf{B}^{k-1} - \mathbf{B}^k)}{||\mathbf{B}^{k-1} - \mathbf{B}^k)||^2}$

set $c_{\Omega}^k = \frac{2(f(\mathbf{B}^{k-1},\Omega^{k-1}) - f(\mathbf{B}^k,\Omega^k) - \nabla^{\mathrm{T}} f(\Omega^k)(\Omega^{k-1} - \Omega^k)}{||\Omega^{k-1} - \Omega^k)||^2}$

if $c_{\mathbf{B}} < 0$ **then**

set $c_{\mathbf{B}} = 10^{-60}$

end if

if $c_{\Omega} < 0$ **then**

set $c_{\Omega} = 10^{-60}$

end if

$k = k + 1$

end while

5.6 Alternative Optimization Algorithms

In addition to gradient descent via the modified SQSD algorithm, global heuristic algorithms are employed in order to improve the CCG of a combined transmit-and-receive coherent distributed array system. It is important to consider that simply turning the algorithm loose with no constraints has the potential to yield configurations that are technically optimal but practically nonsensical, namely, placing some or all of the nodes in the exact same location. At the very least, a minimum distance between nodes needs to be enforced, both before and after optimization. The optimization problem of maximizing CCG within a fixed number of iterations using a global optimization algorithm is given by

$$\begin{aligned} \max \quad & \tilde{G}(\mathbf{B}, \phi_0, \theta_0) \\ \text{s.t.} \quad & \mathbf{B}^0 - \delta s \leq \mathbf{B}^k \leq \mathbf{B}^0 + \delta s, \\ & \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \geq d_{min}, i \neq j, \\ & \phi_{m,min} \leq \phi_0^k \leq \phi_{m,max}, \\ & \theta_{m,min} \leq \theta_0^k \leq \theta_{m,max}, \\ & k \leq N_{iter}. \end{aligned} \tag{60}$$

The specific algorithms tested are pattern search (PS) and particle swarm optimization (PSO).

5.6.1 Pattern Search

Pattern search (PS), also called direct search, is a class of numerical optimization algorithms that can be used on a broad variety of functions regardless of continuity or differentiability. The name pattern search is first used in [32], which provides a formal definition for direct search and a flow chart for a pattern search routine. Direct search methods were introduced as a means to take on numerical problems that had evaded solution via classical techniques. They employ simple search strategies relative to classical techniques and have proven reliable in providing solutions to many different problems that are typically difficult to solve using other approaches.

The pattern search algorithm used starts with an initial value, or starting point, which, in this case, is the starting locations of all nodes in the network; these values are used as the anchor point. The algorithm calculates the value of the objective function at a series of mesh points that are equidistant from the starting location in all dimensions. If one of the mesh points is an improvement, then it becomes the new anchor point and the size of the mesh is expanded; if no improvement is found among the mesh points, then the anchor point remains the same and the mesh is contracted.

The PS algorithm was run using the $patternsearch()$ function from the Global Optimization toolbox. For pattern search (PS), the generalized pattern search (GPS) algorithm is employed using the positive basis $2N$ polling method. The initial mesh size is set to 10, the expansion factor is set to 2, and the contraction factor is set to 0.75. Complete polling was used over "short-circuit" polling, which generally resulted in better results at the expense of slightly longer algorithm running times.

For its terminating condition, the PS algorithm has an iteration cap of $N_{iter} = 100$ iterations. It was found that increasing N_{iter} beyond 100 resulted in greatly diminished returns (increased running time without increased CCG gain).

5.6.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a generalized metaheuristic that does not rely on the use of a gradient and can be applied to a broad variety of problem classes [33, 34]. The algorithm works by first selecting a population of candidate solutions, or "particles;" in this context, a particle is a specific value of $(\mathbf{B}, \phi_0, \theta_0)$. Each particle is assigned an initial velocity from a range of possible values. The particles are moved about the search space and the objective function is evaluated at the particle locations. Particle velocity v is updated using the following expression:

$$v = Wv + y_1 u_1 (p - x) + y_2 u_2 (g - x), \tag{61}$$

where W is inertia, x is current position of the particle, p is the best position seen by the particle, g is the best position seen by the current neighborhood, u_1, u_2 are randomly chosen from $\tilde{U}(0, 1)$, and y_1, y_2 are user-controlled weighting parameters. The particles are moved in an iterative process toward better locations until the stopping criteria are met.

As with the other optimization methods used, PSO is stopped after a fixed number of iterations. For the simulations on CCG improvement, N_{iter} is set to 500 iterations. PSO initializes with 100 particles and particle velocity is randomly selected from $[-r, r]$, where $r = \delta s$. Neighborhood size is set to 25, and the self and social adjustment weights y_1 and y_2 are kept at the default of 1.49. In addition, a hybrid function is employed whereby the optimization is continued after PSO ends; in this case, $fmincon$ is used, which finds the minimum of a constrained nonlinear multivariable function using an interior point method.

6. CLOSED-FORM GRADIENT EXPRESSIONS

In this chapter, closed-form expressions for the gradient terms of $G(\mathbf{B}, \phi_0, \theta_0)$ are derived for free-space propagation.

6.1 Gradient Expressions for Planar Model

First, the planar system model is considered. Using an expression for CCG, the system gain can be calculated for a particular configuration of nodes and transmitter phase offsets that are determined by the values of $\mathbf{B}$ and ϕ_0. In order to explore how G changes with small changes in node positions and transmitter array beam angle, a gradient concept for CCG must be introduced. If the variables in Eq. (31) are considered, the expression has $2(N+M)+1$ degrees of freedom: an X value and a Y value for each node, and the azimuth angle of the transmitter array. Therefore, the gradient of G should represent the rate of change of G along each of the $2(N+M)+1$ dimensions, which is written as

$$\nabla G = \hat{x}_1^t \frac{\partial}{\partial x_1^t} G + \hat{y}_1^t \frac{\partial}{\partial y_1^t} G + \ldots + \hat{x}_N^t \frac{\partial}{\partial x_N^t} G + \hat{y}_N^t \frac{\partial}{\partial y_N^t} G + \hat{x}_1^r \frac{\partial}{\partial x_1^r} G + \hat{y}_1^r \frac{\partial}{\partial y_1^r} G + \ldots + \hat{x}_M^r \frac{\partial}{\partial x_M^r} G + \hat{y}_M^r \frac{\partial}{\partial y_M^r} G + \hat{\phi}_0 \frac{\partial}{\partial \phi_0} G. \tag{62}$$

A free-space propagation model is used in deriving the gradient terms, so $h_{nm} = (4\pi \mathrm{d_{nm}})^{-2}$, where $\mathrm{d_{nm}} = \sqrt{(x_n^t - x_m^r)^2 + (y_n^t - y_m^r)^2}$.

6.1.1 Transmitter Node Position

Because $\tilde{G}(\mathbf{B}, \phi_0)$ is calculated as the magnitude of the sum of complex terms, the coherent communication gain expression of Eq. (31) is manipulated in order to make it more amenable to having its partial derivative taken. The gradient components associated with the transmitter nodes' X and Y values are found.

First, the complex exponentials in Eq. (31) are replaced using Euler's identity and the expression is simplified:

$$\begin{aligned}
\tilde{G}(\mathbf{B}, \phi_0) &= E_G \sum_{m=1}^{M} \left| \sum_{n=1}^{N} h_{nm} A_n \; e^{j(a_m x_n^t + b_m y_n^t)} \right|^2 \\
&= E_G \sum_{m=1}^{M} \left| \sum_{n=1}^{N} h_{nm} A_n \cos(a_m x_n^t + b_m y_n^t) + j \sum_{n=1}^{N} h_{nm} A_n \sin(a_m x_n^t + b_m y_n^t) \right|^2 \\
&= E_G \sum_{m=1}^{M} \left(\left[\sum_{n=1}^{N} h_{nm} A_n \cos(a_m x_n^t + b_m y_n^t) \right]^2 + \left[\sum_{n=1}^{N} h_{nm} A_n \sin(a_m x_n^t + b_m y_n^t) \right]^2 \right).
\end{aligned} \tag{63}$$

Another set of substitution variables is defined:

$$\mathcal{U}_{nm} = \sum_{n=1}^{N} h_{nm} A_n \cos\left(a_m x_n^t + b_m y_n^t\right), \tag{64}$$

$$\mathcal{V}_{nm} = \sum_{n=1}^{N} h_{nm} A_n \sin\left(a_m x_n^t + b_m y_n^t\right). \tag{65}$$

Each transmitter node contributes two components to the gradient expression: an X component and a Y component. Let (x_i^t, y_i^t) be the current position of the transmitter node of interest, labeled s_i^t. The gradient components associated with s_i^t indicate how $\tilde{G}(\mathbf{B}, \phi_0)$ changes according to changes in x_i^t and y_i^t, assuming (x_n^t, y_n^t), (x_m^r, y_m^r), $1 \le n \le N; 1 \le m \le M; n \ne i$, and ϕ_0 are held constant. Therefore, the partial derivatives of $\tilde{G}$ with respect to x_i^t and y_i^t need to be found. Using the chain rule, the expression for the partial derivative of $\mathcal{U}_{nm}^2$ with respect to x_i^t is given by

$$\frac{\partial}{\partial x_i^t}\mathcal{U}_{nm}^2 = -2\mathcal{U}_{nm} A_i h_{im} \left[a_m \sin\left(a_m x_i^t + b_m y_i^t\right) + \frac{2(x_i^t - x_m^r)\cos\left(a_m x_i^t + b_m y_i^t\right)}{\mathrm{d_{im}}^2}\right]. \tag{66}$$

Similarly, the partial derivative of $\mathcal{V}_{nm}^2$ with respect to x_i^t is given by

$$\frac{\partial}{\partial x_i^t}\mathcal{V}_{nm}^2 = 2\mathcal{V}_{nm} A_i h_{im} \left[a_m \cos\left(a_m x_i^t + b_m y_i^t\right) - \frac{2(x_i^t - x_m^r)\sin\left(a_m x_i^t + b_m y_i^t\right)}{\mathrm{d_{im}}^2}\right]. \tag{67}$$

Next, the process is repeated to find the gradient components for the Y value of the transmitter node position. The partial derivative of $\mathcal{U}_{nm}^2$ with respect to y_i^t is given by

$$\frac{\partial}{\partial y_i^t}\mathcal{U}_{nm}^2 = -2\mathcal{U}_{nm} A_i h_{im} \left[b_m \sin\left(a_m x_i^t + b_m y_i^t\right) + \frac{2(y_i^t - y_m^r)\cos\left(a_m x_i^t + b_m y_i^t\right)}{\mathrm{d_{im}}^2}\right], \tag{68}$$

and the partial derivative of $\mathcal{V}_{nm}^2$ with respect to y_i^t is given by

$$\frac{\partial}{\partial y_i^t}\mathcal{V}_{nm}^2 = 2\mathcal{V}_{nm} A_i h_{im} \left[b_m \cos\left(a_m x_i^t + b_m y_i^t\right) - \frac{2(y_i^t - y_m^r)\sin\left(a_m x_i^t + b_m y_i^t\right)}{\mathrm{d_{im}}^2}\right]. \tag{69}$$

Since $G(\mathbf{B}, \phi_0)$ is simply $g(m)$ summed across all $m, 1 \leq m \leq M$, and scaled by E_G, the gradient terms for s_i^t are given by

$$\begin{aligned}\frac{\partial}{\partial x_i^t}\tilde{G}(\mathbf{B}, \phi_0) &= E_G \sum_{m=1}^{M}\left(\frac{\partial}{\partial x_i^t}\mathcal{U}_{nm}^2 + \frac{\partial}{\partial x_i^t}\mathcal{V}_{nm}^2\right)\\ \frac{\partial}{\partial y_i^t}\tilde{G}(\mathbf{B}, \phi_0) &= E_G \sum_{m=1}^{M}\left(\frac{\partial}{\partial y_i^t}\mathcal{U}_{nm}^2 + \frac{\partial}{\partial y_i^t}\mathcal{V}_{nm}^2\right).\end{aligned} \tag{70}$$

6.1.2 Receiver Node Position

Next, the gradient expressions that describe the rate of change of $\tilde{G}(\mathbf{B}, \phi_0)$ with respect to the positions of the receiver nodes are determined. From Eq. (62), there are an X gradient component and a Y gradient component for each receiver node, similar to transmitter nodes. Let s_i^r be the receiver node of interest with current position (x_i^r, y_i^r). Note that the variables x_i^r and y_i^r do not explicitly appear in Eq. (31), so they must be substituted in using

$$\phi_i = \arctan\left(\frac{y_i^r}{x_i^r}\right). \tag{71}$$

The substitution variables Eqs. (13) – (14) and Eqs. (64) – (65) are retained, except upon evaluation, the m subscript in the terms is replaced with i to show dependence on the position of the specific receiver node. Starting with x_i^r, the partial derivative of $\tilde{G}(\mathbf{B}, \phi_0)$ is accomplished by first finding the partial derivatives of $\mathcal{U}_{nm}^2$ and $\mathcal{V}_{nm}^2$ separately and summing the expressions. The first $\hat{x}_i^r$ expression is

$$\begin{aligned}\frac{\partial}{\partial x_i^r}\mathcal{U}_{nm}^2 = 4\mathcal{U}_{ni}\sum_{n=1}^{N} h_{ni}A_n\Bigg[&\frac{\cos\left(a_i x_n^t + b_i y_n^t\right)}{\mathrm{d}_{\mathrm{ni}}^2}(x_n^t - x_i^r)\\ &+ (y_n^t\cos\phi_i - x_n^t\sin\phi_i)\sin\left(a_i x_n^t + b_i y_n^t\right)\frac{\pi y_i^r}{(x_i^r)^2 + (y_i^r)^2}\Bigg].\end{aligned} \tag{72}$$

The second partial derivative is given by

$$\begin{aligned}\frac{\partial}{\partial x_i^r}\mathcal{V}_{nm}^2 = 4\mathcal{V}_{ni}\sum_{n=1}^{N} h_{ni}A_n\Bigg[&\frac{\sin\left(a_i x_n^t + b_i y_n^t\right)}{\mathrm{d}_{\mathrm{ni}}^2}(x_n^t - x_i^r)\\ &+ (x_n^t\sin\phi_i - y_n^t\cos\phi_i)\cos\left(a_i x_n^t + b_i y_n^t\right)\frac{\pi y_i^r}{(x_i^r)^2 + (y_i^r)^2}\Bigg].\end{aligned} \tag{73}$$

The same two-part process is used to find the partial derivative of $\tilde{G}(\mathbf{B}, \phi_0)$ with respect to y_i^r. The first $\hat{y}_i^r$ expression is

$$\frac{\partial}{\partial y_i^r}\mathcal{U}_{nm}^2 = 4\mathcal{U}_{ni}\sum_{n=1}^{N} h_{ni}A_n\left[\frac{\cos\left(a_i x_n^t + b_i y_n^t\right)}{\mathrm{d}_{\mathrm{ni}}^2}(y_n^t - y_i^r)\right.$$
$$\left. + \left(x_n^t \sin\phi_i - y_n^t\cos\phi_i\right)\sin\left(a_i x_n^t + b_i y_n^t\right)\frac{\pi x_i^r}{(x_i^r)^2 + (y_i^r)^2}\right]. \tag{74}$$

The second half of the y_i^r partial derivative is given by

$$\frac{\partial}{\partial y_i^r}\mathcal{V}_{nm}^2 = 4\mathcal{V}_{ni}\sum_{n=1}^{N} h_{ni}A_n\left[\frac{\sin\left(a_i x_n^t + b_i y_n^t\right)}{\mathrm{d}_{\mathrm{ni}}^2}(y_n^t - y_i^r)\right.$$
$$\left. + \left(y_n^t \cos\phi_i - x_n^t\sin\phi_i\right)\cos\left(a_i x_n^t + b_i y_n^t\right)\frac{\pi x_i^r}{(x_i^r)^2 + (y_i^r)^2}\right]. \tag{75}$$

Upon taking the partial derivative of $\tilde{G}(\mathbf{B}, \phi_0)$ with respect to either x_i^r or y_i^r, only the $m = i$ term remains in the outer summation, so the expressions of Eqs. (72) – (75) are simply summed and scaled by E_G to yield

$$\frac{\partial}{\partial x_i^r}\tilde{G}(\mathbf{B}, \phi_0) = E_G\left(\frac{\partial}{\partial x_i^r}\mathcal{U}_{nm}^2 + \frac{\partial}{\partial x_i^r}\mathcal{V}_{nm}^2\right) \tag{76}$$

$$\frac{\partial}{\partial y_i^r}\tilde{G}(\mathbf{B}, \phi_0) = E_G\left(\frac{\partial}{\partial y_i^r}\mathcal{U}_{nm}^2 + \frac{\partial}{\partial y_i^r}\mathcal{V}_{nm}^2\right). \tag{77}$$

6.1.3 Transmitter Array Azimuth Angle

The last expression required in the planar model is for the gradient component describing the rate of change of $\tilde{G}(\mathbf{B}, \phi_0)$ with respect to ϕ_0, the beam angle of the transmitter array. For this derivation, the substitution variables Eqs. (64) – (65) are used, along with additional substitution variables defined as

$$p_{nm} = x_n\cos\phi_m + y_n\sin\phi_m, \tag{78}$$

$$q_{nm} = x_n\cos\phi_0 + y_n\sin\phi_0. \tag{79}$$

After applying the substitutions, $\tilde{G}(\mathbf{B}, \phi_0)$ can be expressed as

$$\tilde{G}(\mathbf{B},\phi_0) = E_G \sum_{m=1}^{M}\left(\left[\sum_{n=1}^{N} h_{nm}A_n \cos\left(2\pi(p_{nm}-q_{nm})\right)\right]^2 + \left[\sum_{n=1}^{N} h_{nm}A_n \sin\left(2\pi(p_{nm}-q_{nm})\right)\right]^2\right). \quad (80)$$

The partial derivative of $\mathcal{U}_{nm}^2$ with respect to ϕ_0 is found using the chain rule after expanding v_{nm}:

$$\frac{\partial}{\partial\phi_0}\mathcal{U}_{nm}^2 = 4\pi\mathcal{U}_{nm}\sum_{n=1}^{N} A_n h_{nm}\cos\left(2\pi(p_{nm}-x_n\cos\phi_0-y_n\sin\phi_0)\right)(y_n\cos\phi_0-x_n\sin\phi_0). \quad (81)$$

Similarly, the partial derivative of $\mathcal{V}_{nm}^2$ with respect to ϕ_0 is given by

$$\frac{\partial}{\partial\phi_0}\mathcal{V}_{nm}^2 = 4\pi\mathcal{V}_{nm}\sum_{n=1}^{N} A_n h_{nm}\cos\left(2\pi(p_{nm}-x_n\cos\phi_0-y_n\sin\phi_0)\right)(x_n\sin\phi_0-y_n\cos\phi_0). \quad (82)$$

Finally, the closed-form expression for the $\hat{\phi}_0$ gradient component of $\tilde{G}(\mathbf{B},\phi_0)$ is

$$\frac{\partial}{\partial\phi_0}\tilde{G}(\mathbf{B},\phi_0) = E_G\sum_{m=1}^{M}\left(\frac{\partial}{\partial\phi_0}\mathcal{U}_{nm}^2 + \frac{\partial}{\partial\phi_0}\mathcal{V}_{nm}^2\right). \quad (83)$$

6.2 Gradient Expressions for Volumetric Model

In this section, the gradient expressions are derived for the volumetric model of coherent distributed arrays. Under consideration are $3(N+M)+2$ degrees of freedom: an X value, a Y value, and a Z value for each node, and the azimuth and elevation angles of the transmitter array. Therefore, the gradient of G should represent the rate of change of G along each of the $3(N+M)+2$ dimensions, which is written as

$$\begin{aligned}\nabla G = {} & \hat{x}_1^t\frac{\partial}{\partial x_1^t}G + \hat{y}_1^t\frac{\partial}{\partial y_1^t}G + \hat{z}_1^t\frac{\partial}{\partial z_1^t}G + \ldots + \hat{x}_N^t\frac{\partial}{\partial x_N^t}G + \hat{y}_N^t\frac{\partial}{\partial y_N^t}G + \hat{z}_N^t\frac{\partial}{\partial z_N^t}G + \hat{x}_1^r\frac{\partial}{\partial x_1^r}G \\ & + \hat{y}_1^r\frac{\partial}{\partial y_1^r}G + \hat{z}_1^r\frac{\partial}{\partial z_1^r}G + \ldots + \hat{x}_M^r\frac{\partial}{\partial x_M^r}G + \hat{y}_M^r\frac{\partial}{\partial y_M^r}G + \hat{z}_M^r\frac{\partial}{\partial z_M^r}G + \hat{\phi}_0\frac{\partial}{\partial\phi_0}G + \hat{\theta}_0\frac{\partial}{\partial\theta_0}G. \end{aligned} \quad (84)$$

Closed-form expressions for finding the gradient terms that make up Eq. (84) are derived in this section. A free-space propagation model is used in deriving the gradient terms, so $h_{nm} = (4\pi \mathrm{d}_{\mathrm{nm}})^{-2}$, where $\mathrm{d}_{\mathrm{nm}} = \sqrt{(x_n^t - x_m^r)^2 + (y_n^t - y_m^r)^2 + (z_n^t - z_m^r)^2}$.

6.2.1 Transmitter Node Position

For ease of tracking derivations of the gradient expressions relevant to the volumetric model, a set of substitution variables is introduced:

$$a_m = 2\pi(\sin\theta_m \cos\phi_m - \sin\theta_0 \cos\phi_0), \tag{85}$$

$$b_m = 2\pi(\sin\theta_m \sin\phi_m - \sin\theta_0 \sin\phi_0), \tag{86}$$

$$c_m = 2\pi(\cos\theta_m - \cos\theta_0), \tag{87}$$

$$u_{nm} = \cos\left(a_m x_n^t + b_m y_n^t + c_m z_n^t\right), \tag{88}$$

$$v_{nm} = \sin\left(a_m x_n^t + b_m y_n^t + c_m z_n^t\right), \tag{89}$$

$$\mathcal{U}_{nm} = \sum_{n=1}^{N} h_{nm} A_n u_{nm}, \tag{90}$$

$$\mathcal{V}_{nm} = \sum_{n=1}^{N} h_{nm} A_n v_{nm}. \tag{91}$$

The volumetric CCG expression Eq. (17) is modified via the Euler's identity conversion used in the planar case, shown in Eq. (63). With the new substitution variables, the CCG of the distributed array system is given by

$$\tilde{G}(\mathbf{B}, \phi_0, \theta_0) = E_g \sum_{m=1}^{M} \left(\mathcal{U}_{nm}^2 + \mathcal{V}_{nm}^2\right). \tag{92}$$

Let s_i^t be the transmitter node of interest with current location given by (x_i^t, y_i^t, z_i^t). The partial derivative of Eq. (92) is taken with respect to x_i^t, y_i^t, and z_i^t. Beginning with the $\hat{x}_i^t$ component, the partial derivatives of the two summands of Eq. (92) are taken. The partial derivative of $\mathcal{U}_{nm}^2$ with respect to x_i^t is given by

$$\begin{aligned}\frac{\partial}{\partial x_i^t}\mathcal{U}_{nm}^2 = 2A_i\mathcal{U}_{nm}\Bigg[&\frac{-2(x_i^t - x_m^r)\cos\left(a_m x_i^t + b_m y_i^t + c_m z_i^t\right)}{16\pi^2\left[(x_i^t - x_m^r)^2 + (y_i^t - y_m^r)^2 + (z_i^t - z_m^r)^2\right]^2} \\ &- \frac{a_m \sin\left(a_m x_i^t + b_m y_i^t + c_m z_i^t\right)}{16\pi^2\left[(x_i^t - x_m^r)^2 + (y_i^t - y_m^r)^2 + (z_i^t - z_m^r)^2\right]}\Bigg] \\ = \frac{2A_i\mathcal{U}_{nm}}{16\pi^2}&\left[\frac{-2(x_i^t - x_m^r)u_{im}}{d_{im}^4} - \frac{a_m v_{im}}{d_{im}^2}\right].\end{aligned} \tag{93}$$

The partial derivative of $\mathcal{V}_{nm}^2$ with respect to x_i^t is given by

$$\frac{\partial}{\partial x_i^t}\mathcal{V}_{nm}^2 = \frac{2A_i\mathcal{V}_{nm}}{16\pi^2}\left[\frac{-2(x_i^t - x_m^r)v_{im}}{d_{im}^4} + \frac{a_m u_{im}}{d_{im}^2}\right]. \tag{94}$$

Next, the process is repeated to find the $\hat{y}_i^t$ gradient terms. The partial derivative of $\mathcal{U}_{nm}^2$ with respect to y_i^t is given by

$$\frac{\partial}{\partial y_i^t}\mathcal{U}_{nm}^2 = \frac{2A_i\mathcal{U}_{nm}}{16\pi^2}\left[\frac{-2(y_i^t - y_m^r)u_{im}}{d_{im}^4} - \frac{b_m v_{im}}{d_{im}^2}\right]. \tag{95}$$

The partial derivative of $\mathcal{V}_{nm}^2$ with respect to y_i^t is given by

$$\frac{\partial}{\partial y_i^t}\mathcal{V}_{nm}^2 = \frac{2A_i\mathcal{V}_{nm}}{16\pi^2}\left[\frac{-2(y_i^t - y_m^r)v_{im}}{d_{im}^4} + \frac{b_m u_{im}}{d_{im}^2}\right]. \tag{96}$$

Lastly, the partial derivatives of the summand terms are with respect to z_i^t are given by

$$\frac{\partial}{\partial z_i^t}\mathcal{U}_{nm}^2 = \frac{2A_i\mathcal{U}_{nm}}{16\pi^2}\left[\frac{-2(z_i^t - z_m^r)u_{im}}{d_{im}^4} - \frac{c_m v_{im}}{d_{im}^2}\right], \tag{97}$$

$$\frac{\partial}{\partial z_i^t}\mathcal{V}_{nm}^2 = \frac{2A_i\mathcal{V}_{nm}}{16\pi^2}\left[\frac{-2(z_i^t - z_m^r)v_{im}}{d_{im}^4} + \frac{c_m u_{im}}{d_{im}^2}\right]. \tag{98}$$

The gradient terms for the transmitter node location components are stated together as

$$\frac{\partial}{\partial x_i^t}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 2E_G A_i \sum_{m=1}^{M} h_{im}\left[a_m(\mathcal{V}_{im}u_{im} - \mathcal{U}_{im}v_{im}) - 2\frac{x_i^t - x_m^r}{d_{im}^2}(\mathcal{U}_{im}u_{im} + \mathcal{V}_{im}v_{im})\right], \tag{99}$$

$$\frac{\partial}{\partial y_i^t}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 2E_G A_i \sum_{m=1}^{M} h_{im}\left[b_m(\mathcal{V}_{im}u_{im} - \mathcal{U}_{im}v_{im}) - 2\frac{y_i^t - y_m^r}{d_{im}^2}(\mathcal{U}_{im}u_{im} + \mathcal{V}_{im}v_{im})\right], \tag{100}$$

$$\frac{\partial}{\partial z_i^t}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 2E_G A_i \sum_{m=1}^{M} h_{im}\left[c_m(\mathcal{V}_{im}u_{im} - \mathcal{U}_{im}v_{im}) - 2\frac{z_i^t - z_m^r}{d_{im}^2}(\mathcal{U}_{im}u_{im} + \mathcal{V}_{im}v_{im})\right]. \tag{101}$$

6.2.2 Receiver Node Position

In this subsection, the gradient expressions describing rate of change of $\tilde{G}(\mathbf{B}, \phi_0, \theta_0)$ with respect to a receiver node are derived. Let s_i^r be the receiver node of interest with current position given by (x_i^r, y_i^r, z_i^r). Note that in Eq. (17), these coordinates for s_i^r are not explicitly included. Therefore, they must be substituted into the expression in place of the transmitter array beam angles using the identity of ϕ_i given by Eq. (71) and the identity of θ_i given by

$$\theta_i = \arctan\left(\frac{\sqrt{(x_i^r)^2 + (y_i^r)^2}}{z_i^r}\right). \tag{102}$$

In addition to the substitution variables Eqs. (85) – (91), the following variables are introduced:

$$\eta_{nm} = 2\pi x_n^t \sin\theta_m \cos\phi_m = 2\pi x_n^t \sin\left(\arctan\left(\frac{\sqrt{(x_m^r)^2 + (y_m^r)^2}}{z_m^r}\right)\right)\cos\left(\arctan\left(\frac{y_m^r}{x_m^r}\right)\right), \tag{103}$$

$$\rho_{nm} = 2\pi y_n^t \sin\theta_m \cos\phi_m = 2\pi y_n^t \sin\left(\arctan\left(\frac{\sqrt{(x_m^r)^2 + (y_m^r)^2}}{z_m^r}\right)\right)\sin\left(\arctan\left(\frac{y_m^r}{x_m^r}\right)\right), \tag{104}$$

$$\sigma_{nm} = 2\pi z_n^t \sin\theta_m \cos\phi_m = 2\pi z_n^t \sin\left(\arctan\left(\frac{\sqrt{(x_m^r)^2 + (y_m^r)^2}}{z_m^r}\right)\right). \tag{105}$$

These substitution variables can be used to rewrite Eqs. (88) – (89) as

$$u_{nm} = \cos\left(\eta_{nm} + \rho_{nm} + \sigma_{nm} - 2\pi(x_n^t \sin\theta_0 \cos\phi_0 + y_n^t \sin\theta_0 \sin\phi_0 + z_n^t \cos\theta_0)\right), \tag{106}$$

$$v_{nm} = \sin\left(\eta_{nm} + \rho_{nm} + \sigma_{nm} - 2\pi(x_n^t \sin\theta_0 \cos\phi_0 + y_n^t \sin\theta_0 \sin\phi_0 + z_n^t \cos\theta_0)\right). \tag{107}$$

The partial derivatives of the new substitution variables with respect to x_i^r, y_i^r, and z_i^r are given by

$$\frac{\partial}{\partial x_i^r}\eta_{nm} = 2\pi x_n^t \left[\frac{\cos\phi_i \cos\theta_i x_i^r z_i^r}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)\sqrt{(x_i^r)^2 + (y_i^r)^2}} + \frac{y_i^r \sin\theta_i \sin\phi_i}{(x_i^r)^2 + (y_i^r)^2} \right], \tag{108}$$

$$\frac{\partial}{\partial x_i^r}\rho_{nm} = 2\pi y_n^t \left[\frac{\sin\phi_i \cos\theta_i x_i^r z_i^r}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)\sqrt{(x_i^r)^2 + (y_i^r)^2}} - \frac{y_i^r \sin\theta_i \cos\phi_i}{(x_i^r)^2 + (y_i^r)^2} \right], \tag{109}$$

$$\frac{\partial}{\partial x_i^r}\sigma_{nm} = -2\pi z_n^t \left[\frac{\sin\theta_i x_i^r z_i^r}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)\sqrt{(x_i^r)^2 + (y_i^r)^2}} \right], \tag{110}$$

$$\frac{\partial}{\partial y_i^r}\eta_{nm} = 2\pi x_n^t \left[\frac{\cos\phi_i \cos\theta_i y_i^r z_i^r}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)\sqrt{(x_i^r)^2 + (y_i^r)^2}} - \frac{x_i^r \sin\theta_i \sin\phi_i}{(x_i^r)^2 + (y_i^r)^2} \right], \tag{111}$$

$$\frac{\partial}{\partial y_i^r}\rho_{nm} = 2\pi y_n^t \left[\frac{\sin\phi_i \cos\theta_i y_i^r z_i^r}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)\sqrt{(x_i^r)^2 + (y_i^r)^2}} + \frac{x_i^r \sin\theta_i \cos\phi_i}{(x_i^r)^2 + (y_i^r)^2} \right], \tag{112}$$

$$\frac{\partial}{\partial y_i^r}\sigma_{nm} = -2\pi z_n^t \left[\frac{\sin\theta_i y_i^r z_i^r}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)\sqrt{(x_i^r)^2 + (y_i^r)^2}} \right], \tag{113}$$

$$\frac{\partial}{\partial z_i^r}\eta_{nm} = -2\pi x_n^t \left[\frac{\cos\phi_i \cos\theta_i \sqrt{(x_i^r)^2 + (y_i^r)^2}}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)} \right], \tag{114}$$

$$\frac{\partial}{\partial z_i^r}\rho_{nm} = -2\pi y_n^t \left[\frac{\sin\phi_i \cos\theta_i \sqrt{(x_i^r)^2 + (y_i^r)^2}}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)} \right], \tag{115}$$

$$\frac{\partial}{\partial z_i^r}\sigma_{nm} = 2\pi z_n^t \left[\frac{\sin\theta_i \sqrt{(x_i^r)^2 + (y_i^r)^2}}{((x_i^r)^2 + (y_i^r)^2 + (z_i^r)^2)} \right], \tag{116}$$

$$\tag{117}$$

Then, the gradient terms for the receiver node position can be stated as

$$\frac{\partial}{\partial x_i^r}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 2E_G\sum_{n=1}^{N} A_n h_{ni}\left[2\frac{(x_n^t - x_i^r)}{d_{ni}^2}(\mathcal{U}_{ni}u_{ni} + \mathcal{V}_{ni}v_{ni}) + (\mathcal{V}_{ni}u_{ni} - \mathcal{U}_{ni}v_{ni})\left(\frac{\partial}{\partial x_i^r}\{\eta_{nm} + \rho_{nm} + \sigma_{nm}\}\right)\right], \tag{118}$$

$$\frac{\partial}{\partial y_i^r}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 2E_G\sum_{n=1}^{N} A_n h_{ni}\left[2\frac{(y_n^t - y_i^r)}{d_{ni}^2}(\mathcal{U}_{ni}u_{ni} + \mathcal{V}_{ni}v_{ni}) + (\mathcal{V}_{ni}u_{ni} - \mathcal{U}_{ni}v_{ni})\left(\frac{\partial}{\partial y_i^r}\{\eta_{nm} + \rho_{nm} + \sigma_{nm}\}\right)\right], \tag{119}$$

$$\frac{\partial}{\partial z_i^r}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 2E_G\sum_{n=1}^{N} A_n h_{ni}\left[2\frac{(z_n^t - z_i^r)}{d_{ni}^2}(\mathcal{U}_{ni}u_{ni} + \mathcal{V}_{ni}v_{ni}) + (\mathcal{V}_{ni}u_{ni} - \mathcal{U}_{ni}v_{ni})\left(\frac{\partial}{\partial z_i^r}\{\eta_{nm} + \rho_{nm} + \sigma_{nm}\}\right)\right]. \tag{120}$$

6.2.3 Transmitter Array Azimuth and Elevation Angles

The final gradient expressions for the volumetric model describe the rate of change of the CCG with respect to the azimuth and elevation angles of the transmitter array beam, given by ϕ_0 and θ_0, respectively. Once again, the substitution variables Eqs. (85) – (91) are used.

Beginning with the azimuth angle ϕ_0, partial derivatives are taken for the two summand terms of Eq. (92), which are

$$\frac{\partial}{\partial \phi_0}\mathcal{U}_{nm}^2 = 4\pi\sin\theta_0\mathcal{U}_{nm}\sum_{n=1}^{N} A_n h_{nm}v_{nm}\left[y_n^t\cos\phi_0 - x_n^t\sin\phi_0\right], \tag{121}$$

$$\frac{\partial}{\partial \phi_0}\mathcal{V}_{nm}^2 = 4\pi\sin\theta_0\mathcal{V}_{nm}\sum_{n=1}^{N} A_n h_{nm}u_{nm}\left[x_n^t\sin\phi_0 - y_n^t\cos\phi_0\right]. \tag{122}$$

Next, the partial derivatives are taken with respect to the elevation angle θ_0, yielding

$$\frac{\partial}{\partial \theta_0}\mathcal{U}_{nm}^2 = 4\pi\mathcal{U}_{nm}\sum_{n=1}^{N} A_n h_{nm}v_{nm}\left[x_n^t\cos\theta_0\cos\phi_0 + y_n^t\cos\theta_0\sin\phi_0 - z_n^t\sin\theta_0\right], \tag{123}$$

$$\frac{\partial}{\partial \theta_0}\mathcal{V}_{nm}^2 = -4\pi\mathcal{V}_{nm}\sum_{n=1}^{N} A_n h_{nm}u_{nm}\left[x_n^t\cos\theta_0\cos\phi_0 + y_n^t\cos\theta_0\sin\phi_0 - z_n^t\sin\theta_0\right]. \tag{124}$$

Lastly, the gradient components for transmitter array beam angle are

$$\frac{\partial}{\partial\theta_0}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 4\pi E_G \sin\theta_0 \sum_{m=1}^{M}\sum_{n=1}^{N} A_n h_{nm}(\mathcal{U}_{nm}v_{nm} - \mathcal{V}_{nm}u_{nm})\left[y_n^t\cos\phi_0 - x_n^t\sin\phi_0\right], \tag{125}$$

$$\begin{aligned}\frac{\partial}{\partial\theta_0}\tilde{G}(\mathbf{B},\phi_0,\theta_0) = 4\pi E_G \sum_{m=1}^{M}\sum_{n=1}^{N} A_n h_{nm}(&\mathcal{U}_{nm}v_{nm} \\ &- \mathcal{V}_{nm}u_{nm})\left[x_n^t\cos\theta_0\cos\phi_0 + y_n^t\cos\theta_0\sin\phi_0 - z_n^t\sin\theta_0\right].\end{aligned} \tag{126}$$

7. DISTRIBUTED ARRAY SIMULATION

A series of combined transmit-and-receive distributed array systems was simulated using MATLAB 2021. The basic distributed array generation procedure is borrowed from the code used in the transmit array simulation from Chapter 4 and is modified and expanded to also create a receive array. For each simulation run, two volumetric distributed arrays are generated using the usual method. A transmit array is constructed by first selecting array volume dimension L_t and using that value to define the bounds of a cubic volume centered on the origin of the Cartesian coordinate system. Within the cubic array volume, (x_n^t, y_n^t, z_n^t) values are generated for each of the N transmitter nodes using a uniform random distribution. For the receive array, the same process is used. The receive array size dimension L_r is selected and is then used to bound the cube centered on $(\frac{L_t}{2} + D + \frac{L_r}{2}, 0, 0)$ that defines the receive array volume. The M nodes are placed in the receive array volume using a uniform random distribution to generate the values of (x_m^r, y_m^r, z_m^r).

For each combination of parameters and optimization algorithm, a total of 500 networks are randomly generated and optimized. Free space propagation is used to calculate h_{nm}, so there are no additional phase shifts associated with complex propagation effects. For SQSD, the iterations are capped at $N_{iter} = 500$, maximum step size for node movement is set to $\rho_{\mathbf{B}} = 1$ wavelength per iteration, and the maximum step size for Tx array beam angles is typically between $10^{-7} \leq \rho_\Omega \leq 10^{-6}$, although this is highly dependent on $\delta\phi_m$ and $\delta\theta_m$. The initial Tx array beam angle (ϕ_0, θ_0) is found by calculating $G(\mathbf{B}, \phi_0, \theta_0)$ over the range of angles $\phi_{m,min} \leq \phi_0 \leq \phi_{m,max}, \theta_{m,min} \leq \theta_0 \leq \theta_{m,max}$ and finding the global maximum. A resolution of 250 points for $\delta\phi_m$ and $\delta\theta_m$ was found to yield a good balance between resolution and computing time on a reasonably powerful desktop computer.

The baseline configuration for the distributed array system is $N = 16$, $M = 16$, $L_t = 1k$, $L_r = 1k$. Several values are used for interarray distance, so $D \in \{10k, 20k, 30k, 40k, 50k\}$. For PS and PSO, maximum node displacement per dimension δs varies along with D, and the corresponding values are $\delta s \in \{10, 14, 18, 22, 26\}$. What this means is the global optimization algorithms search for the optimal location of each node within a constrained volume; to find the optimal location for transmit node s_n^t, the algorithm searches within the volume defined by the cube with edge length $2\delta s$ centered on (x_n^t, y_n^t, z_n^t). The reason the node displacement constraint was set up to be a hypercube rather than a hypersphere is because it is much easier to compute displacement per dimension rather than across all dimensions. This is an important consideration for distributed networks that must balance limited battery and compute resources.

7.1 Metrics

For each simulated network that forms a combined transmit-and-receive distributed array system, initial CCG is found by optimizing $G(\phi_0, \theta_0)$. The initial node positions given by $\mathbf{B}$ along with the optimal value of Ω_0 comprise the initial conditions of either SQSD, PS, or PSO, which is run for a fixed number of iterations N_{iter}. After the optimization algorithm is run, several metrics are recorded:

- CCG of the distributed array system in its initial configuration, given by $G(\mathbf{B}^0, \phi_0^0, \theta_0^0)$,
- CCG of the distributed array system in its final configuration, given by $G(\mathbf{B}^{N_{iter}}, \phi_0^{N_{iter}}, \theta_0^{N_{iter}})$,
- average Tx node displacement from initial position, given by $\frac{\|p^t\|_1}{N}$,
- average Rx node displacement from initial position, given by $\frac{\|p^r\|_1}{M}$, and
- algorithm running time, given by t_{opt}.

The displacement of Tx node s_n^t is the distance between the initial position and the final position after optimization and is given by

$$p_n^t = \sqrt{(x_n^t - x_n^{t(N_{iter})})^2 + (y_n^t - y_n^{t(N_{iter})})^2 + (z_n^t - z_n^{t(N_{iter})})^2}. \tag{127}$$

Similarly, the displacement of Rx node s_m^r is given by

$$p_m^r = \sqrt{(x_m^r - x_m^{r(N_{iter})})^2 + (y_m^r - y_m^{r(N_{iter})})^2 + (z_m^r - z_m^{r(N_{iter})})^2}. \tag{128}$$

7.2 Results

Results from the simulations of combined Tx and Rx distributed arrays are presented and discussed in this section. First, distributed array systems are simulated and the CCG for the starting configuration of node positions is calculated. Starting gain is found by finding the optimal Tx array beam angles (ϕ_0, θ_0) that maximize $G(\mathbf{B}, \phi_0, \theta_0)$ for the starting node locations given by $\mathbf{B}^0$. Next, one of the optimization algorithms is run and the CCG for the optimized system is found. Distributed array systems with different array volumes or network sizes are simulated.

7.2.1 Starting Gain

The first three network configurations feature $N = 16$ Tx nodes and $M = 16$ Tx nodes. Both Tx and Rx array volumes are identical and are one of three volumes determined by $L_t = L_r = L \in 100, 1k, 10k$. Interarray distances scale with array volumes such that D ranges from 10 to 50 times L_t, L_r; this way, the relative positioning of the network nodes is proportional. Figure 17 shows distributions of initial CCG versus interarray distance for $L = 100$. Figure 18 displays the initial CCG values versus interarray distance for $L = 1k$. Figure 19 shows initial CCG for the distributed array systems with the largest array volumes of $L = 10k$. As a general trend, for aperiodic distributed arrays with uniformly distributed nodes, as interarray distance D increases, mean CCG tends to increase. This is expected behavior because as D increases, the angular width of the Rx array, given by $\delta\phi_m, \delta\theta_m$, decreases relative to the Tx array. As a result, more Rx nodes are likely to be illuminated by the main beam of the Tx array, resulting in higher CCG. Likewise, L_t, L_r are scaled up and down while keeping D proportional such that the Rx array angular width is constant. From Chapter 4, it was shown that Tx array beam width is inversely proportional to array volume; thus, smaller array volumes result in greater chances that Rx nodes will be illuminated by the main beam. This is supported by Figs. 17–19, which demonstrate that CCG is inversely proportional to Tx and Rx array volumes.

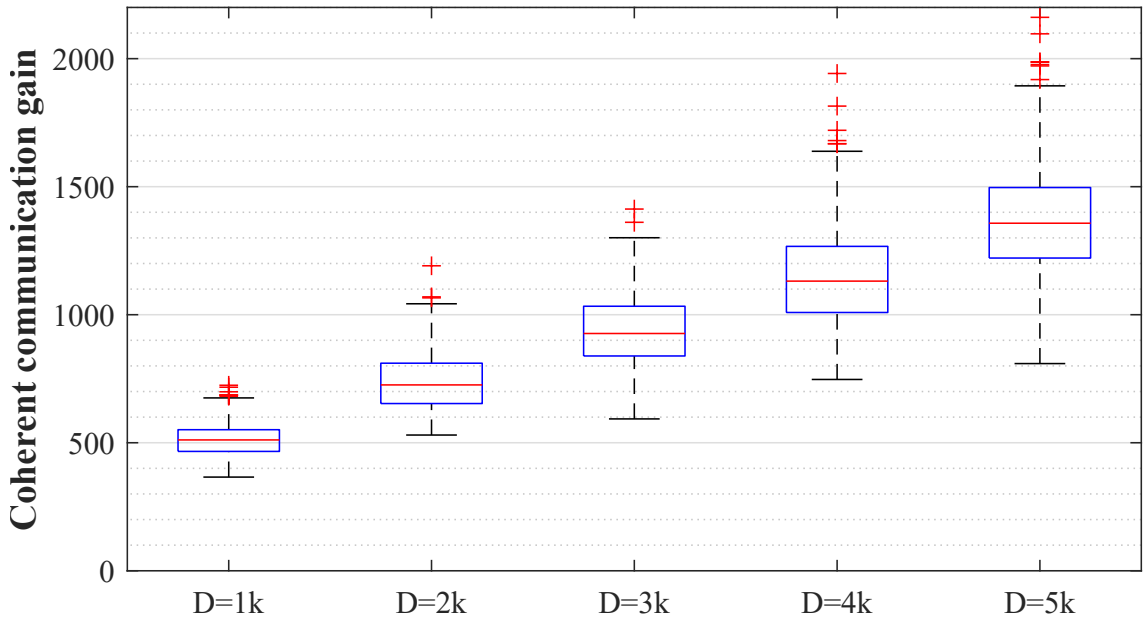

Fig. 17—Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 16, M = 16, L_t = L_r = 100, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

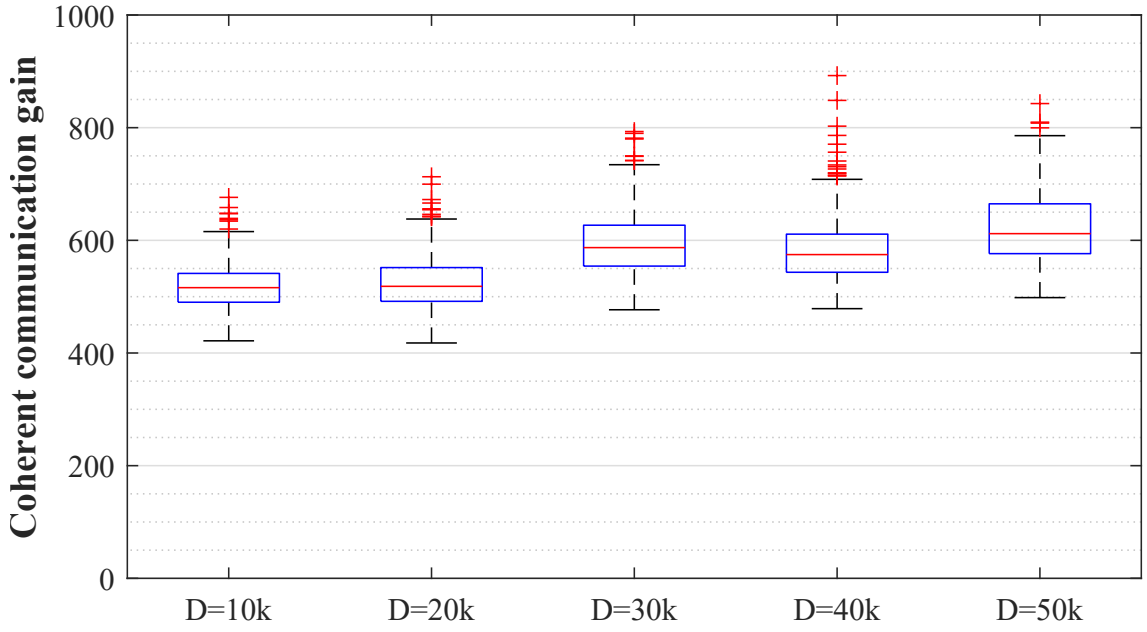

Fig. 18—Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 16, M = 16, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

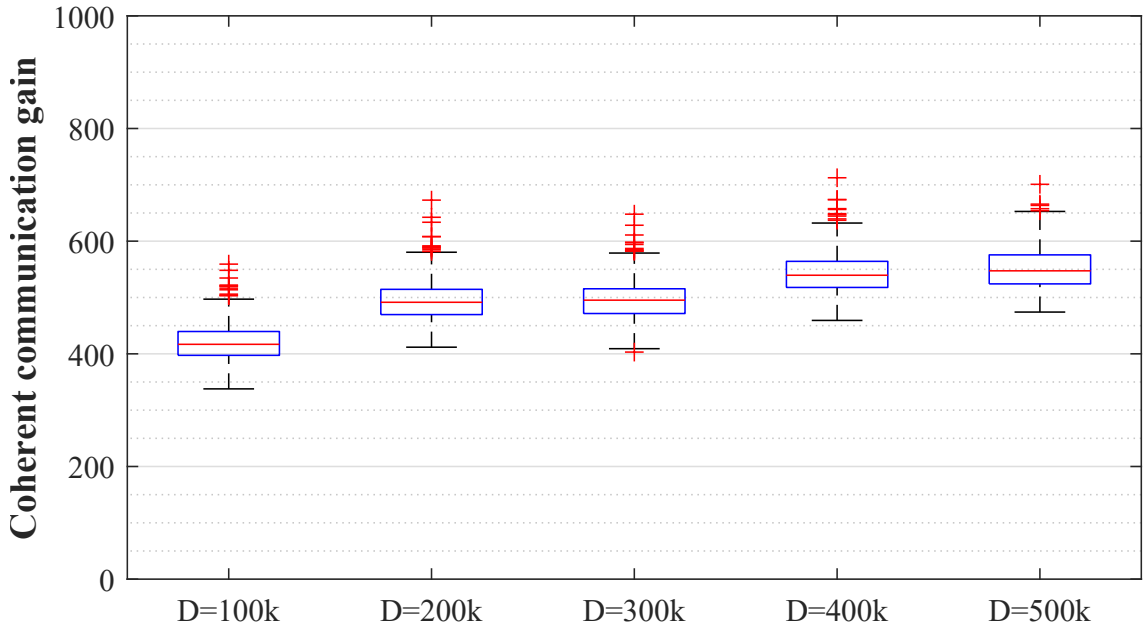

Fig. 19—Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 16, M = 16, L_t = L_r = 10k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

Next, L_t, L_r are fixed at $1k$ and the number of Tx and Rx nodes is modified. Figure 20 shows initial CCG distributions versus D for $N = 8$ Tx nodes and $M = 8$ Rx nodes. As expected, initial CCG before optimization is lower than it is for networks with the same array volumes but twice as many nodes, as shown in Fig. 18. However, the mean CCG values in Fig. 20 are a higher percentage of the upper bound than the CCG values in Fig. 18, about 25–30% compared to about 12–15%. Figure 21 shows initial CCG values versus D for networks with $N = 4$ Tx nodes and $M = 32$ Rx nodes. Initial system gain of these networks is very similar to the initial system gain of the previous configuration, shown in Fig. 20, and both configurations have the same upper bound for G. Figure 22 shows initial CCG values versus D for networks with $N = 32$ Tx nodes and $M = 4$ Rx nodes. These Tx-weighted networks have higher initial gain than the balanced $N = 16, M = 16$ networks, which makes sense, given the quadratic growth of G with number of Tx nodes. Both the balanced array systems and the Tx-weighted array systems have the same theoretical upper bound for G. The initial CCG of a simulated distributed array system rarely measured above $0.3 \cdot G_{UB}$, which suggests that there is considerable room for improvement.

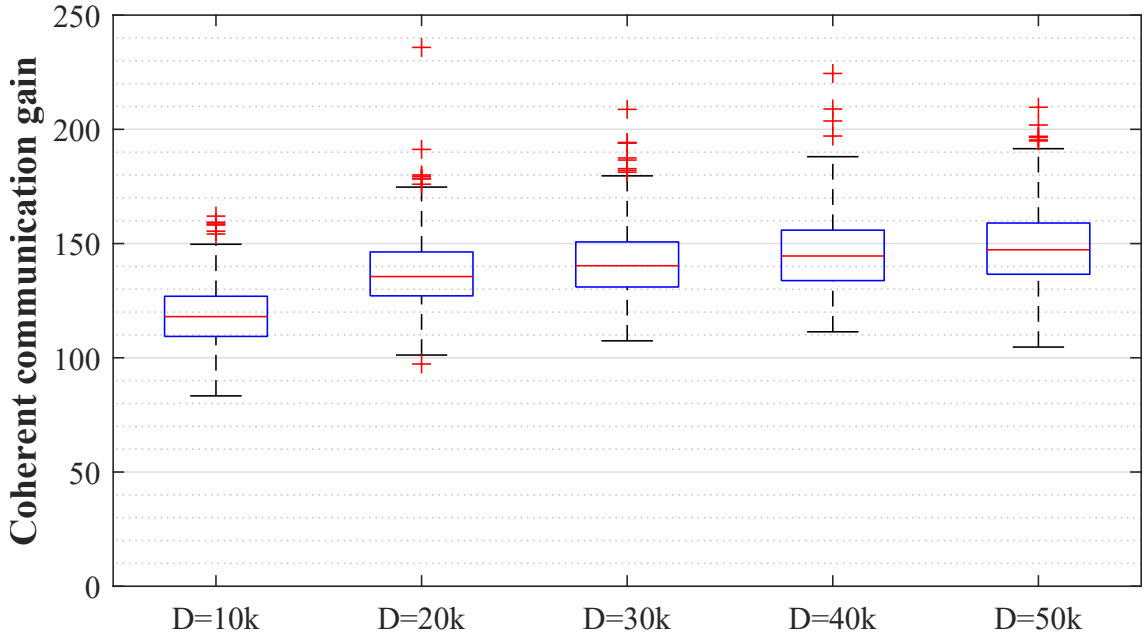

Fig. 20—Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 8, M = 8, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

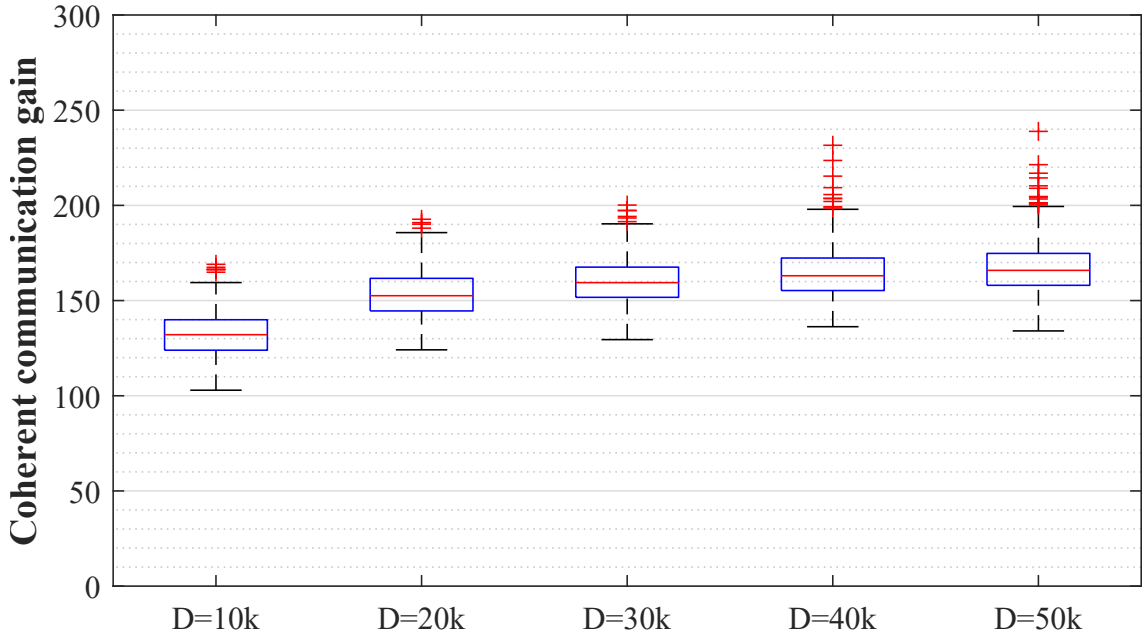

Fig. 21—Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 4, M = 32, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

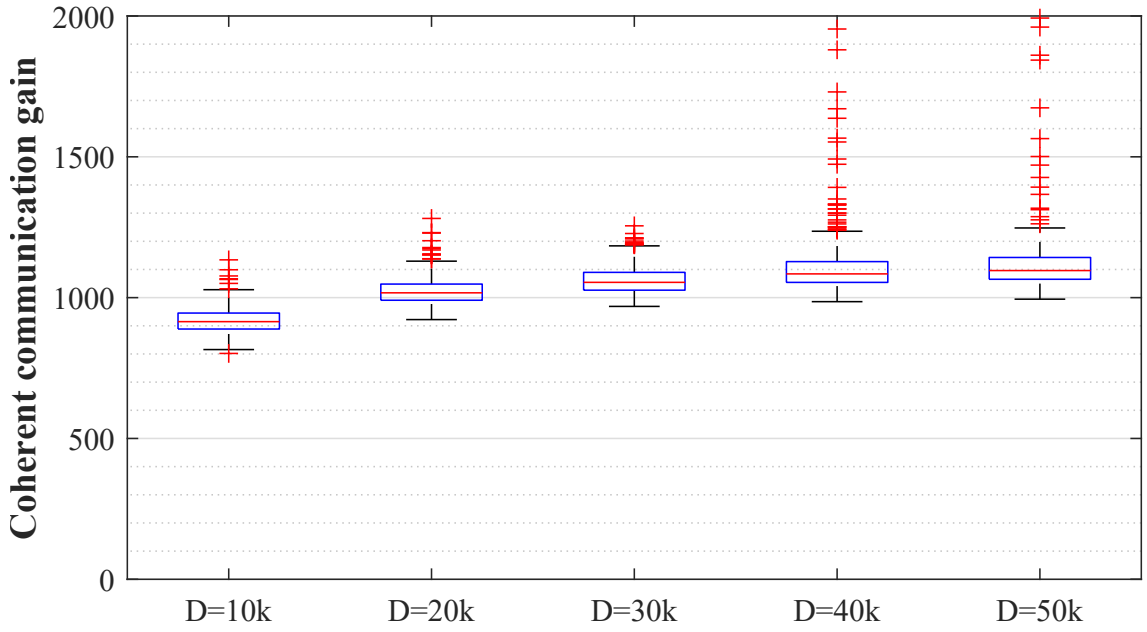

Fig. 22—Distributions of the coherent communication gain of distributed array systems prior to optimizing node locations, shown for a range of interarray distances; $N = 32, M = 4, L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

7.2.2 Baseline Distributions

Once an optimization algorithm has run a fixed number of iterations, the CCG is calculated for the final node positions. The difference between the initial CCG value and the final CCG value is represented as an improvement in dB. Distributions for the performance metrics are shown for the baseline network configuration of $N = 16, M = 16, L_t = L_r = 1k$. Figure 23 displays the distributions of improvement in CCG after using SQSD to find the final node positions, shown for the different internode distances D. CCG improvement is similar across inter-array distances and averages about a 4.5 dB improvement. Figure 24 shows the improvement in CCG after using PS versus interarray distance, and Fig. 25 shows improvement in CCG after using PSO versus interarray distance. For the three optimization algorithms used, improvement in CCG is not greatly influenced by interarray distance. PS yielded the lowest improvement in gain, averaging around 3 to 3.5 dB of increase in CCG. PSO was right in the middle of effectiveness, delivering an average improvement of about 4 dB.

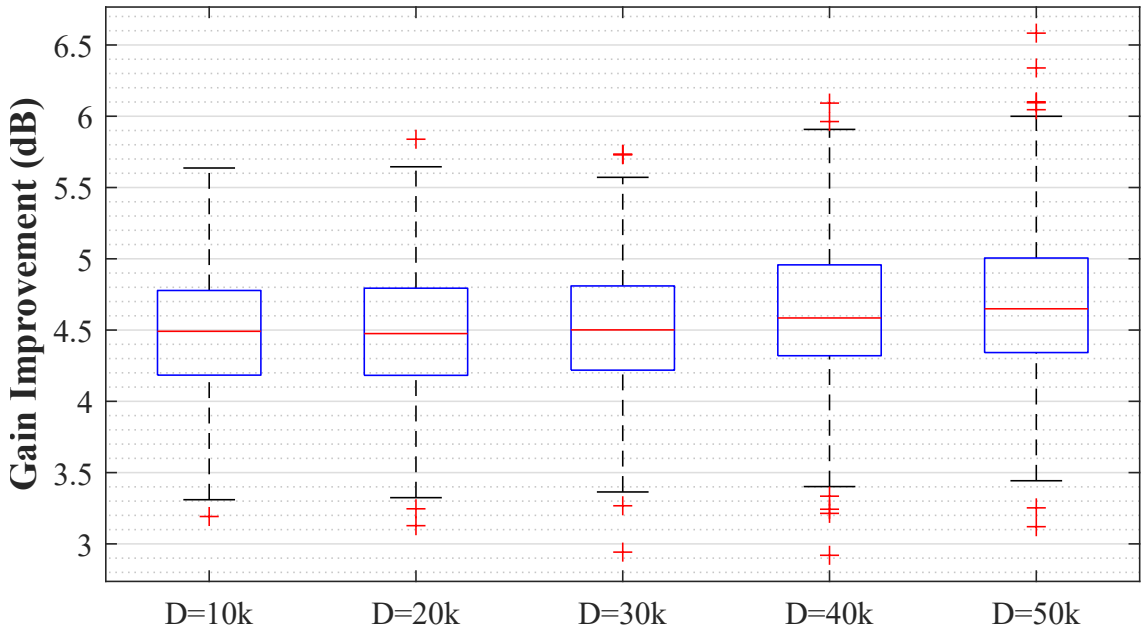

Fig. 23—Distributions of the improvement in coherent communication gain after optimizing node locations using SQSD versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$

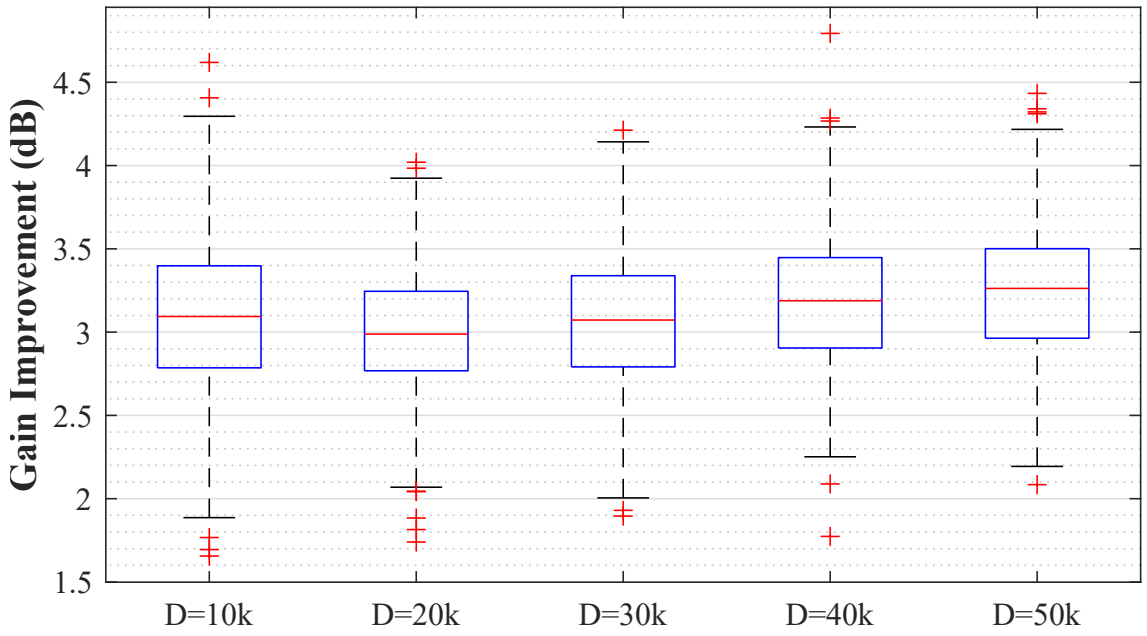

Fig. 24—Distributions of the improvement in coherent communication gain after optimizing node locations using PS versus interarray distance; $N = 16$, $M = 16$, $L_t = L_r = 1k$

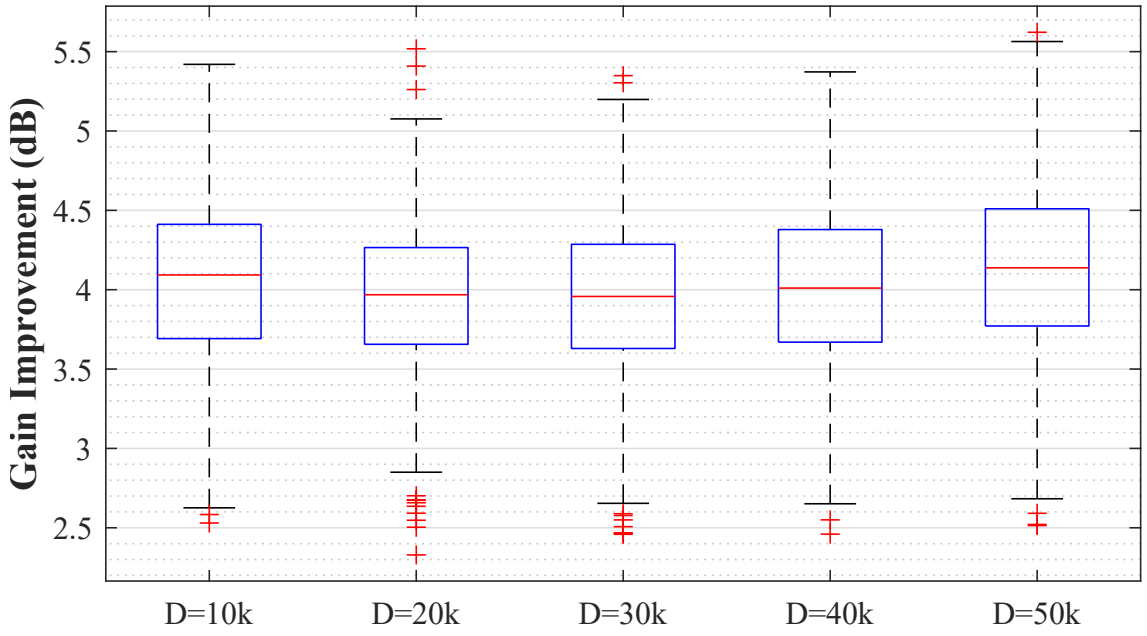

Fig. 25—Distributions of the improvement in coherent communication gain after optimizing node locations using PSO versus interarray distance; $N = 16$, $M = 16$, $L_t = L_r = 1k$

The next distribution plots illustrate the average displacement from initial position experienced by a network node as a result of a CCG optimization algorithm. Figure 26 plots the distributions of average node displacement due to SQSD versus interarray distance, with separate color-coded distributions for each value of D distinguishing between Tx and Rx nodes. Average node displacement generally increases with increasing internode distance. It was mentioned earlier that as D decreases, the objective function becomes more feature-dense, which means a shorter distance between local maxima. Figure 27 shows the distributions of average Tx and Rx node displacement as a result of PS for each value of D, and Fig. 28 shows distributions of average node displacement due to PSO for the different values of D. PS resulted in the least amount of node displacement, and PSO incurred about twice as much displacement as PS.

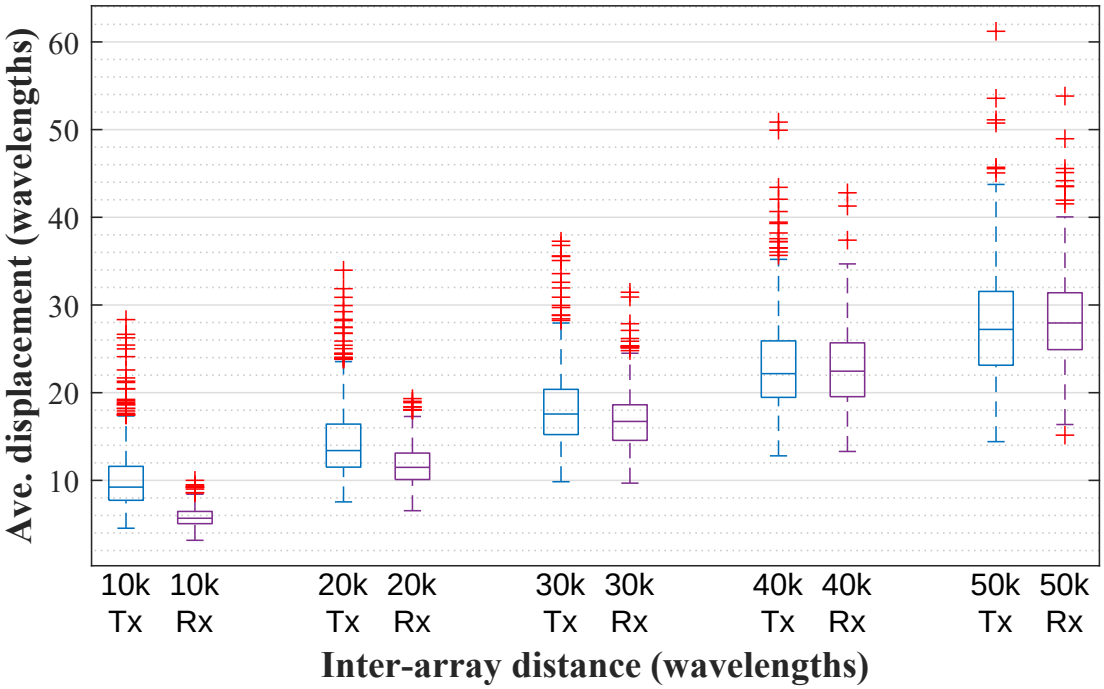

Fig. 26—Distributions of the average node displacement after optimizing node locations using SQSD versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$

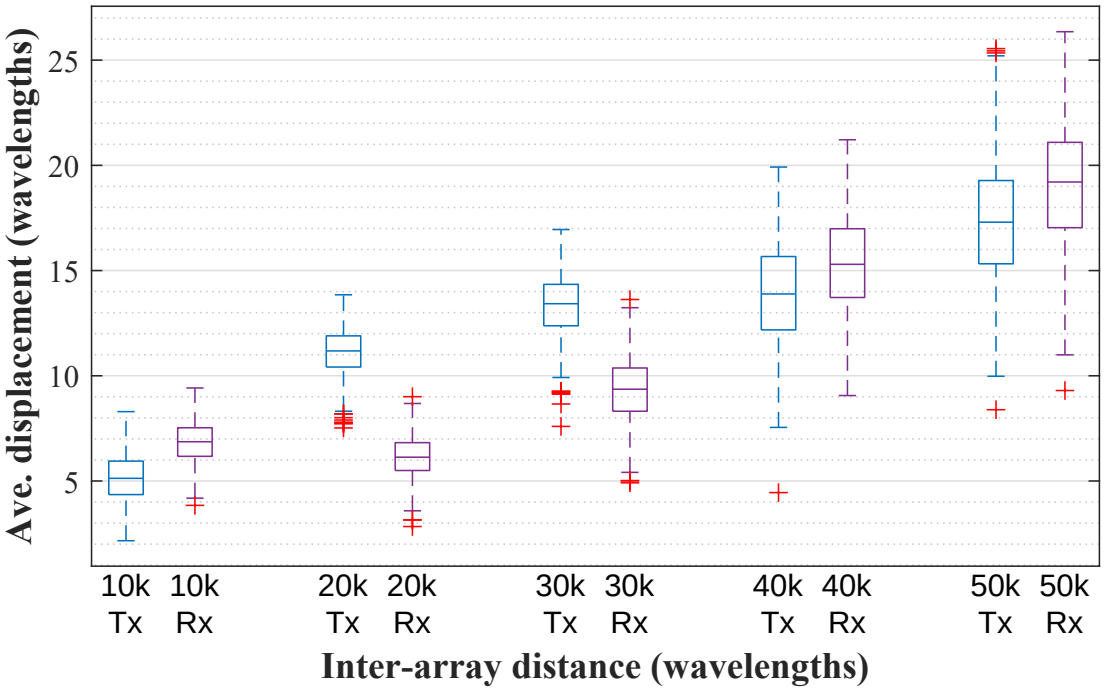

Fig. 27—Distributions of the average node displacement after optimizing node locations using PS versus interarray distance; $N = 16$, $M = 16$, $L_t = L_r = 1k$

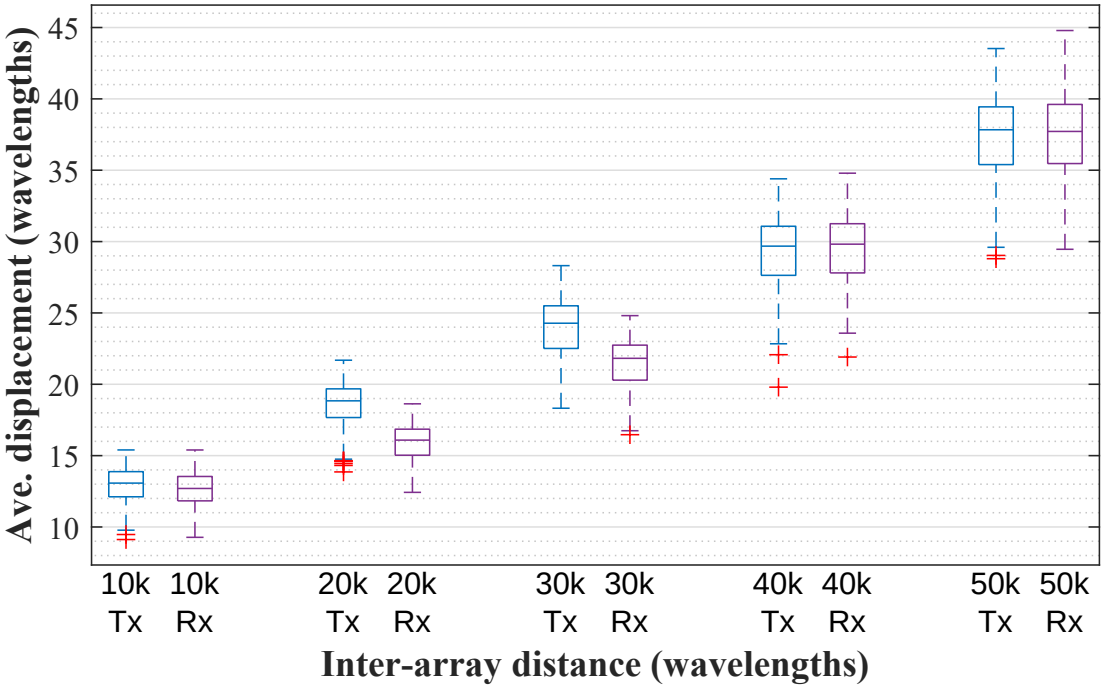

Fig. 28—Distributions of the average node displacement after optimizing node locations using PSO versus interarray distance; $N = 16$, $M = 16$, $L_t = L_r = 1k$

The following box plots display the distributions of algorithm running time for each of the three optimization algorithms. Figure 29 displays running time for SQSD versus interarray distance D. Figure 30 shows algorithm running time versus D for PS, and Fig. 31 shows running time distributions for PSO algorithm. Algorithm running time is not correlated with interarray distance in any meaningful way, which is expected. These plots highlight perhaps the greatest advantage of using a gradient descent method, which is the computational savings over heuristic algorithms like PS and PSO. SQSD averaged about 0.2 s running time, PS averaged between about 0.75 to 1 s running time, and PSO averaged about 7.5 to 8 s running time.

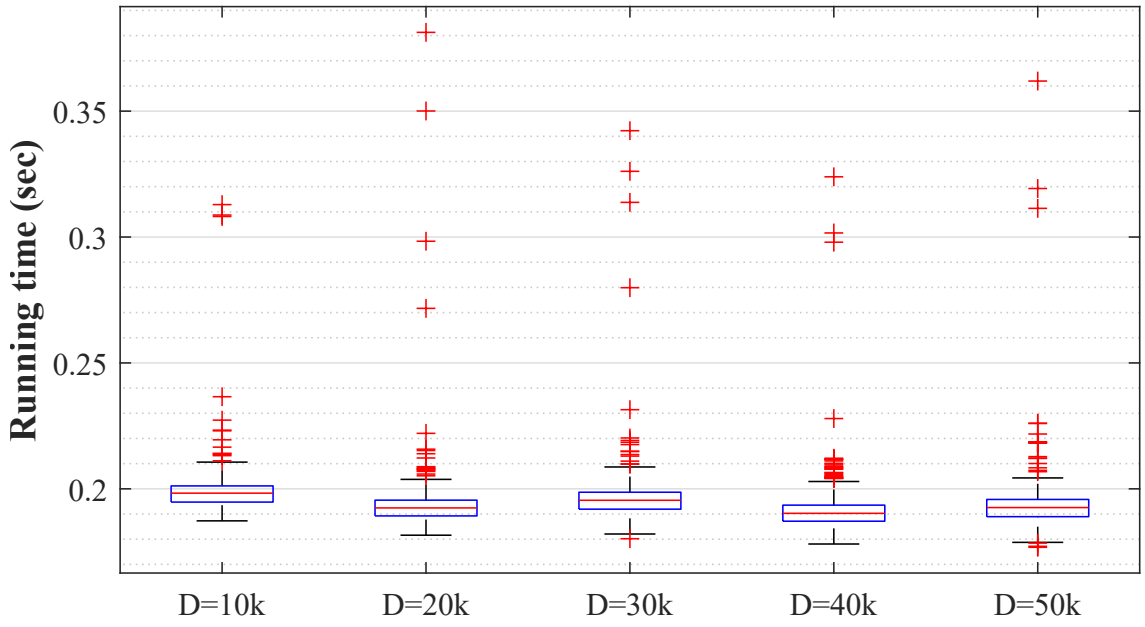

Fig. 29—Distributions of algorithm running time of SQSD used to optimize system gain versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$

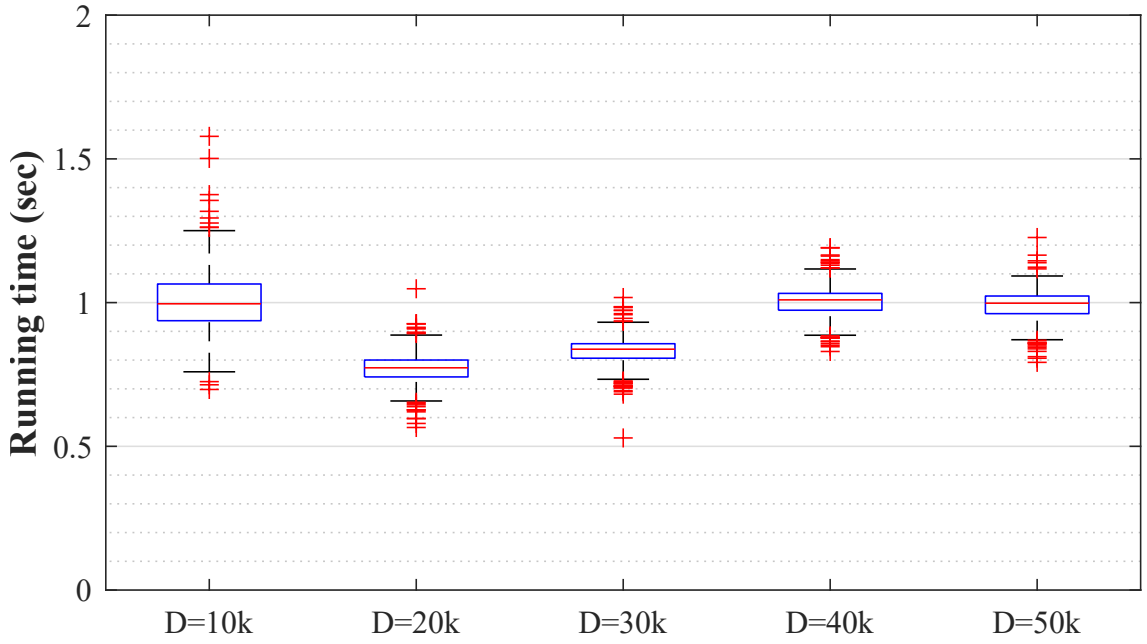

Fig. 30—Distributions of algorithm running time of PS used to optimize system gain versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$

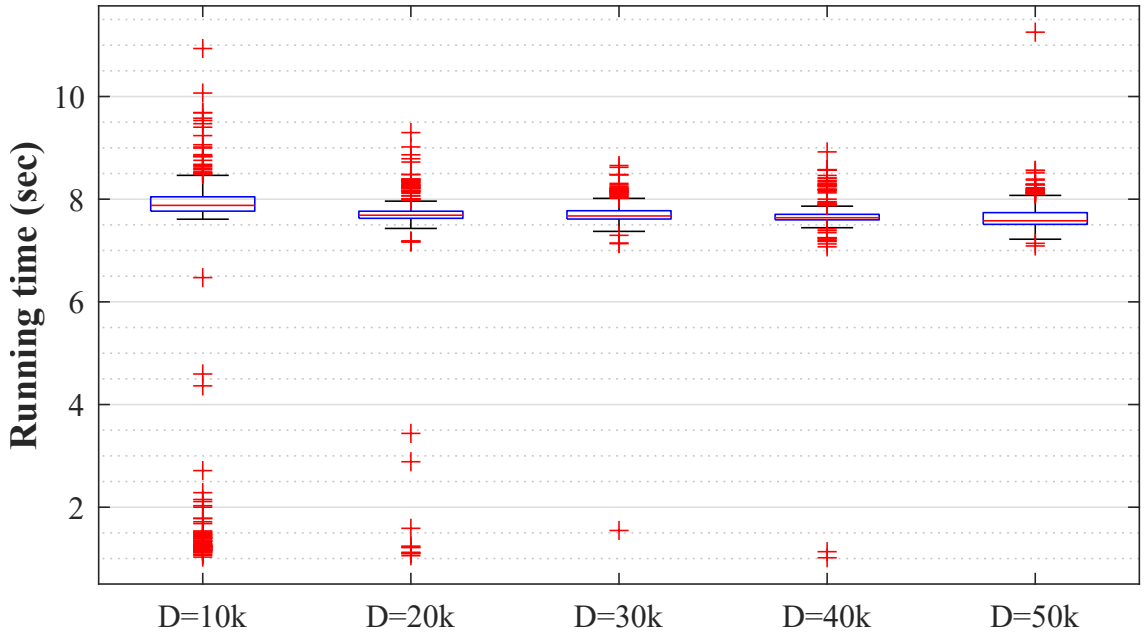

Fig. 31—Distributions of algorithm running time of PSO used to optimize system gain versus interarray distance; $N = 16, M = 16, L_t = L_r = 1k$

7.2.3 Varying Array Volume and Number of Nodes

After the baseline configuration, networks were simulated with larger and smaller array volumes and with different numbers of both Tx and Rx nodes. The distributions of initial CCG value prior to optimization for the various system configurations were shown in Section 7.2.1. The shapes of the results distributions do not change significantly from the baseline configuration, so rather than repeat the corresponding plots from the previous subsection, consolidated plots with quartile values are shown. For CCG improvement in dB, the lower quartile value for a configuration is shown. For node displacement and algorithm running time, the upper quartile value is shown.

First, the array volumes are varied. The network size is kept at $N = 16$ Tx nodes and $M = 16$ Rx nodes, while $L_t = L_r = L$ is changed to 100 and $10k$. The interarray distance changes with L_t/L_r in order to keep the networks spatially proportional, so the x-axis markers used are $c \cdot D_0$, where $c \in 1, 2, 3, 4, 5$ and D_0 is set to $10 \cdot L$. Figure 32 shows the lower quartile values of improvement in CCG versus interarray distance for all three algorithms and all three array volumes. The $L = 100$ configurations see the largest improvement in CCG by a large margin, and they experience the largest variation in improvement versus interarray distance. The $L = 1k$ and $L = 10k$ configurations experience similar levels of improvement, with the $L = 1k$ networks seeing slightly greater improvement over the $L = 10k$ networks.

Figure 33 shows the upper quartile for average Tx node displacement versus interarray distance, and Fig. 34 shows the upper quartile for average Rx node displacement. Average displacement is similar for both Tx and Rx nodes and does not vary considerably between orders of magnitude in the side dimension of the array cubic volume. As with the baseline configuration, PSO consistently displaced nodes further from their starting positions, whereas PS displaced nodes the least. Node displacement due to SQSD is right in the middle. Figure 35 shows the upper quartile for algorithm running time across the different array volumes and algorithms. Results for each algorithm are fairly constant across array volumes and interarray distance.

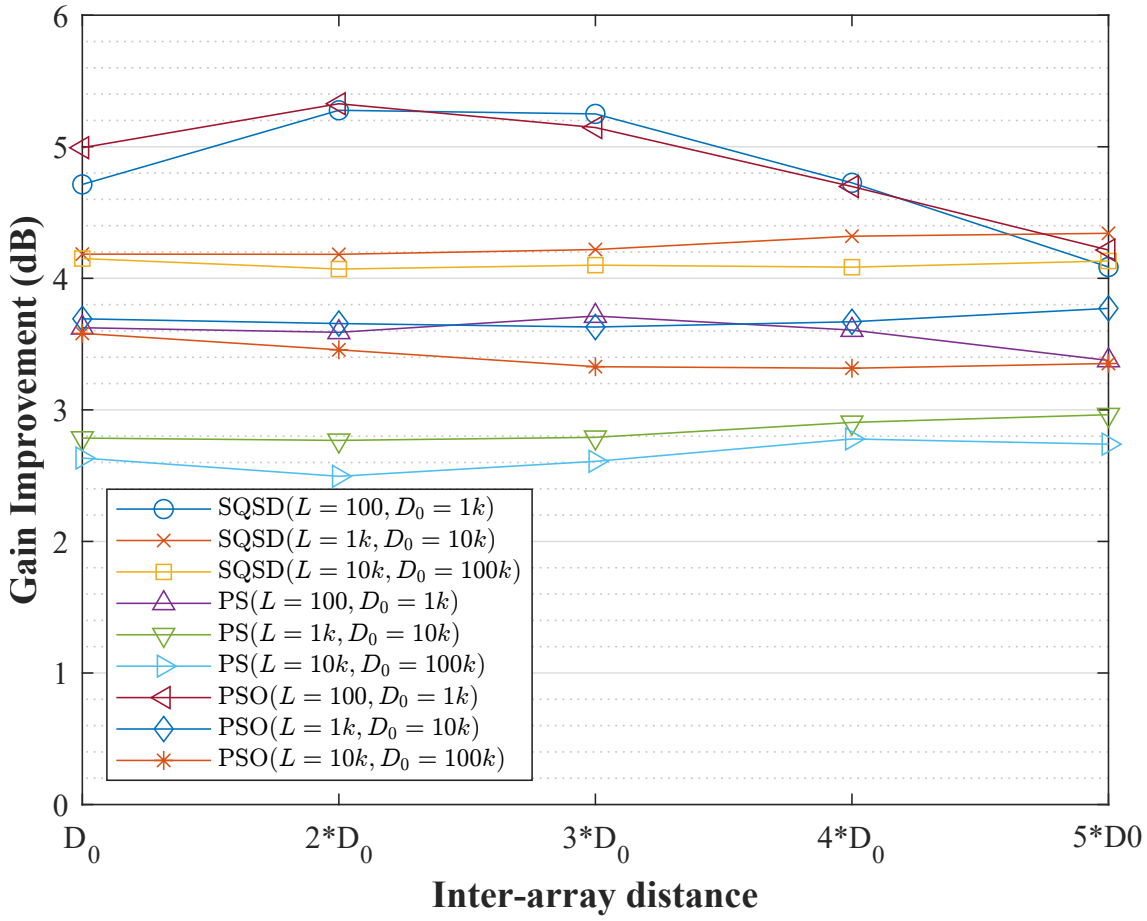

Fig. 32—Lower quartile values of improvement in coherent communication gain versus interarray distance compared across different optimization algorithms and array volumes; $N = 16, M = 16, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

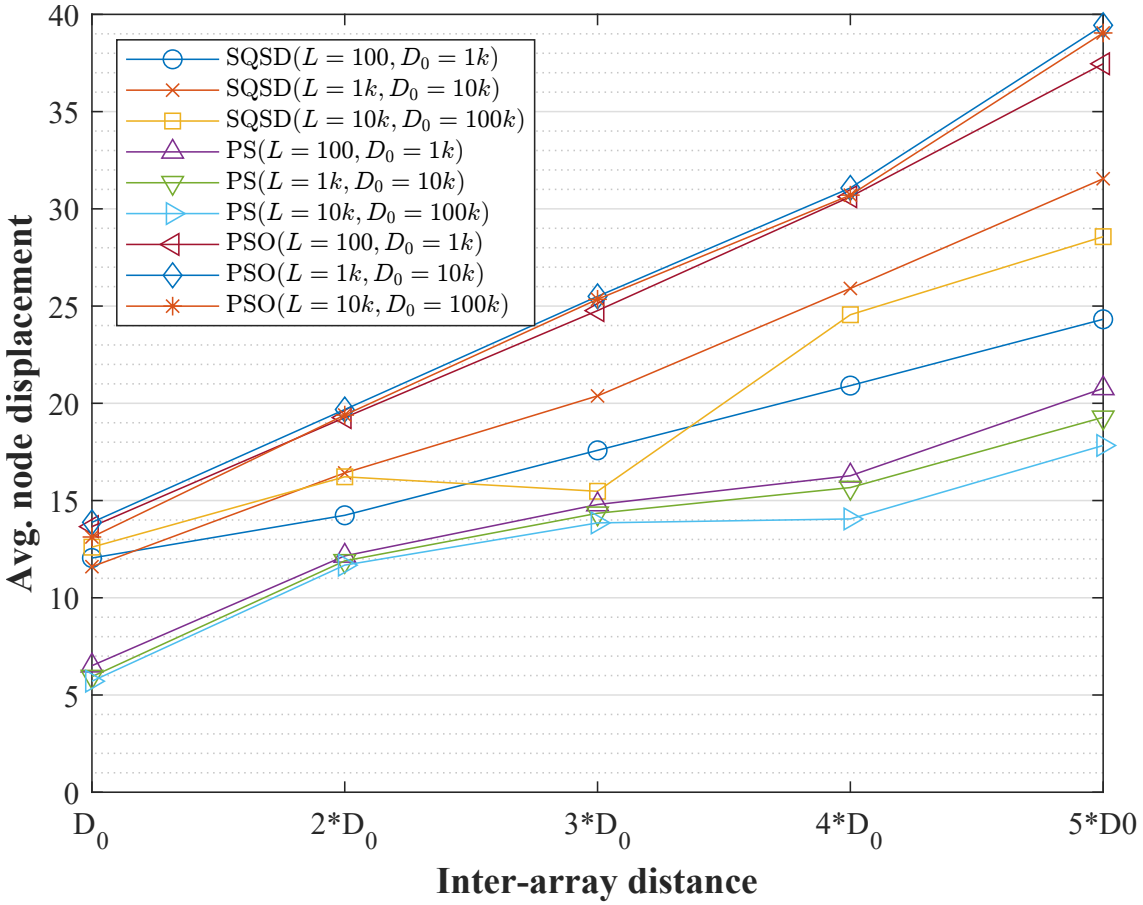

Fig. 33—Upper quartile values of average transmitter node displacement versus interarray distance compared across different optimization algorithms and array volumes; $N = 16$, $M = 16$, $\rho_{\mathbf{B}} = 1$, $\rho_{\phi} \approx 10^{-6}$

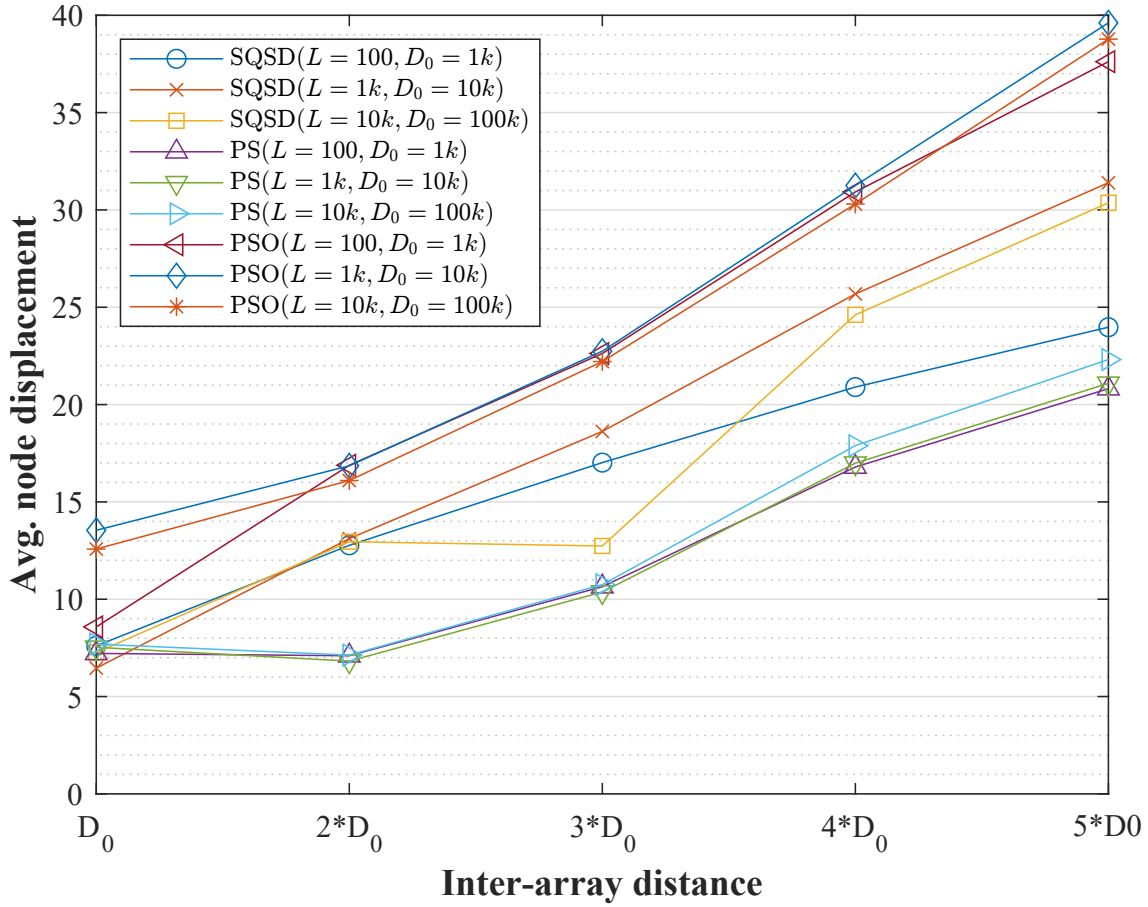

Fig. 34—Upper quartile values of average receiver node displacement versus interarray distance compared across different optimization algorithms and array volumes; $N = 16, M = 16, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

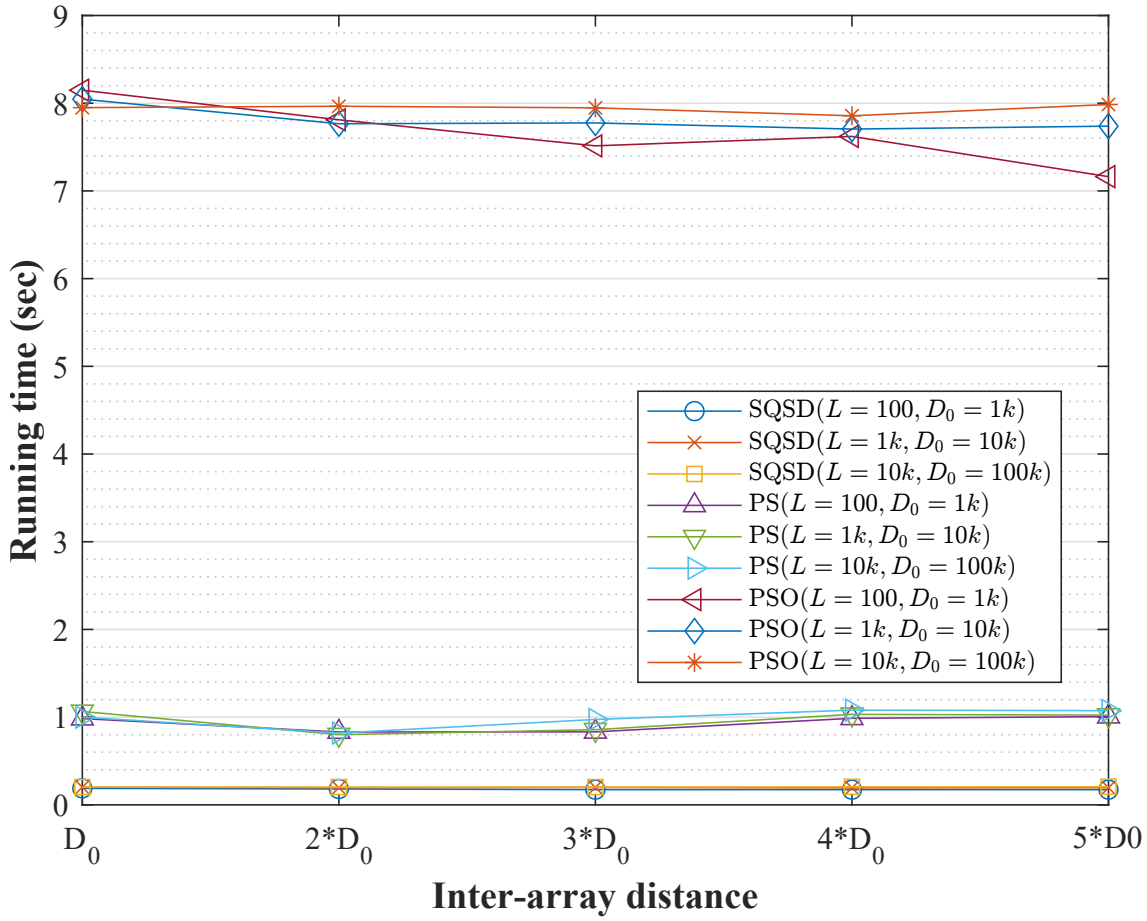

Fig. 35—Upper quartile values of algorithm running time versus interarray distance compared across different optimization algorithms and array volumes; $N = 16, M = 16, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

Next, the array volumes are fixed at $L_t = L_r = 1k$ and the number of nodes is varied in accordance with the starting CCG values shown in Section 7.2.1. Figure 36 displays curves of the lower quartile values of CCG improvement versus interarray distance. Across all three algorithms, the networks with $N = 32$ Tx nodes and $M = 4$ Rx nodes experienced the greatest improvement; these were also the networks that had the highest initial CCG due to greater contribution of transmitters to the system gain. Figure 37 shows the upper quartile values for average Tx node displacement versus D, and Fig. 38 shows the same for Rx nodes. SQSD had the greatest variation in average node displacement across the three different array scenarios, whereas PSO had almost no variation in average node displacement between the three scenarios. Figure 39 shows the upper quartile for algorithm running time across the different network sizes. As expected, PSO took the longest by far, with SQSD running about an order of magnitude faster. The $(N, M) = (8, 8)$ scenario was the fastest to optimize for all algorithms because it featured the smallest total number of nodes. This difference in running time is most apparent with PSO, which was 1.5–2 seconds faster optimizing the $(8, 8)$ scenario versus the $(32, 4)$ and $(4, 32)$ scenarios.

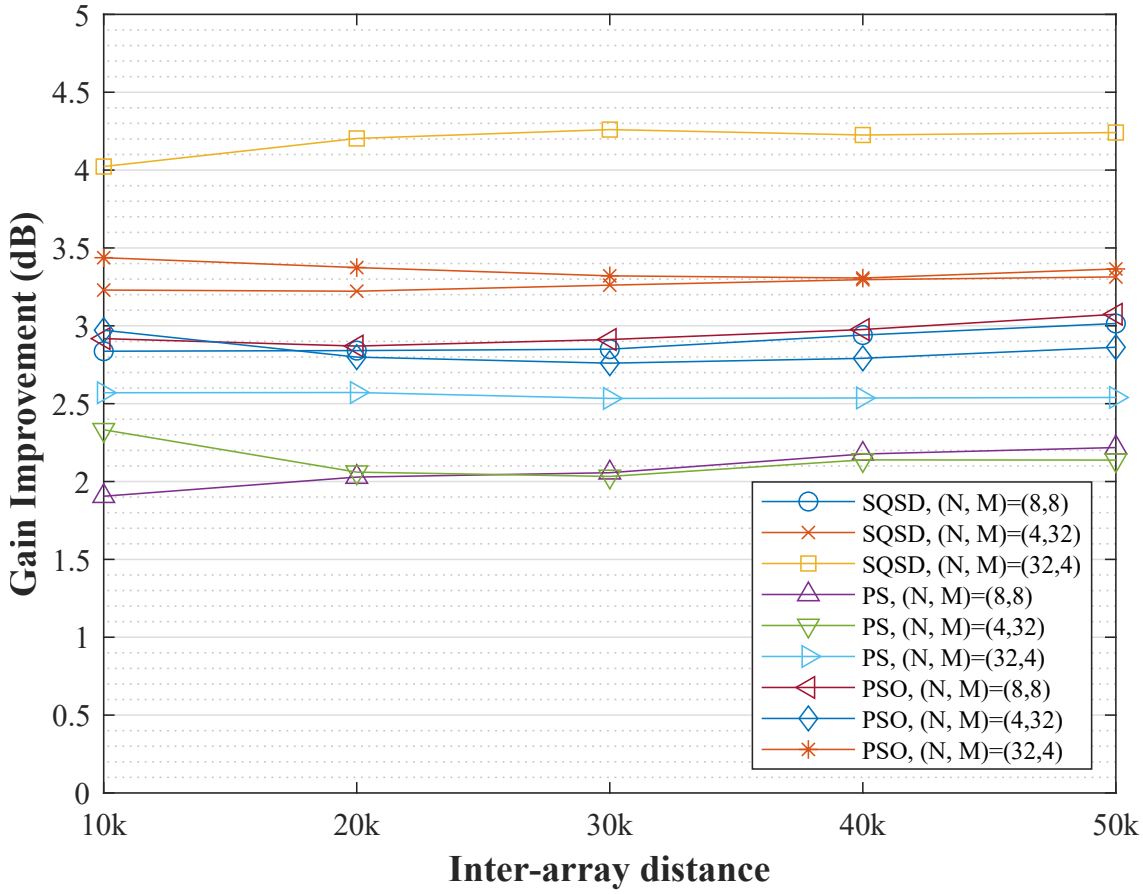

Fig. 36—Lower quartile values of improvement in coherent communication gain versus inter-array distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$

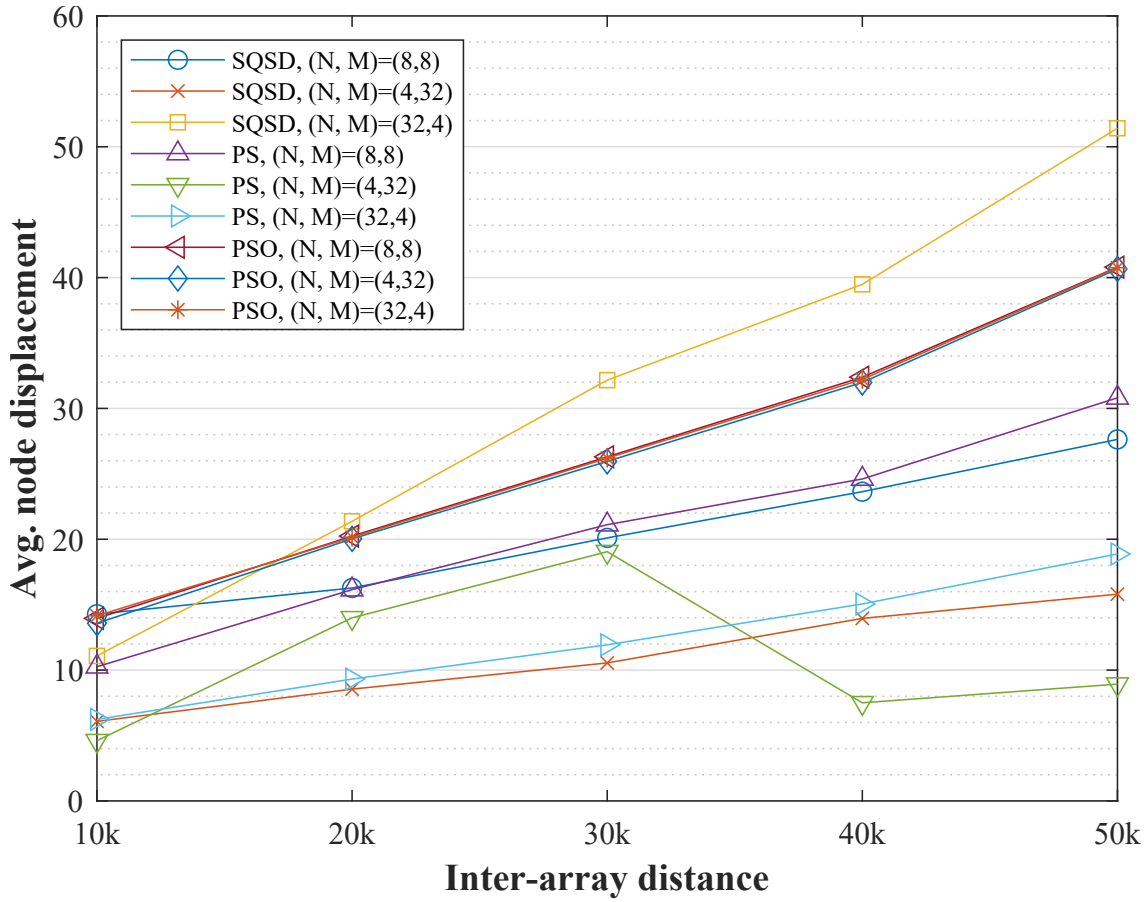

Fig. 37—Upper quartile values of average transmitter node displacement versus interarray distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

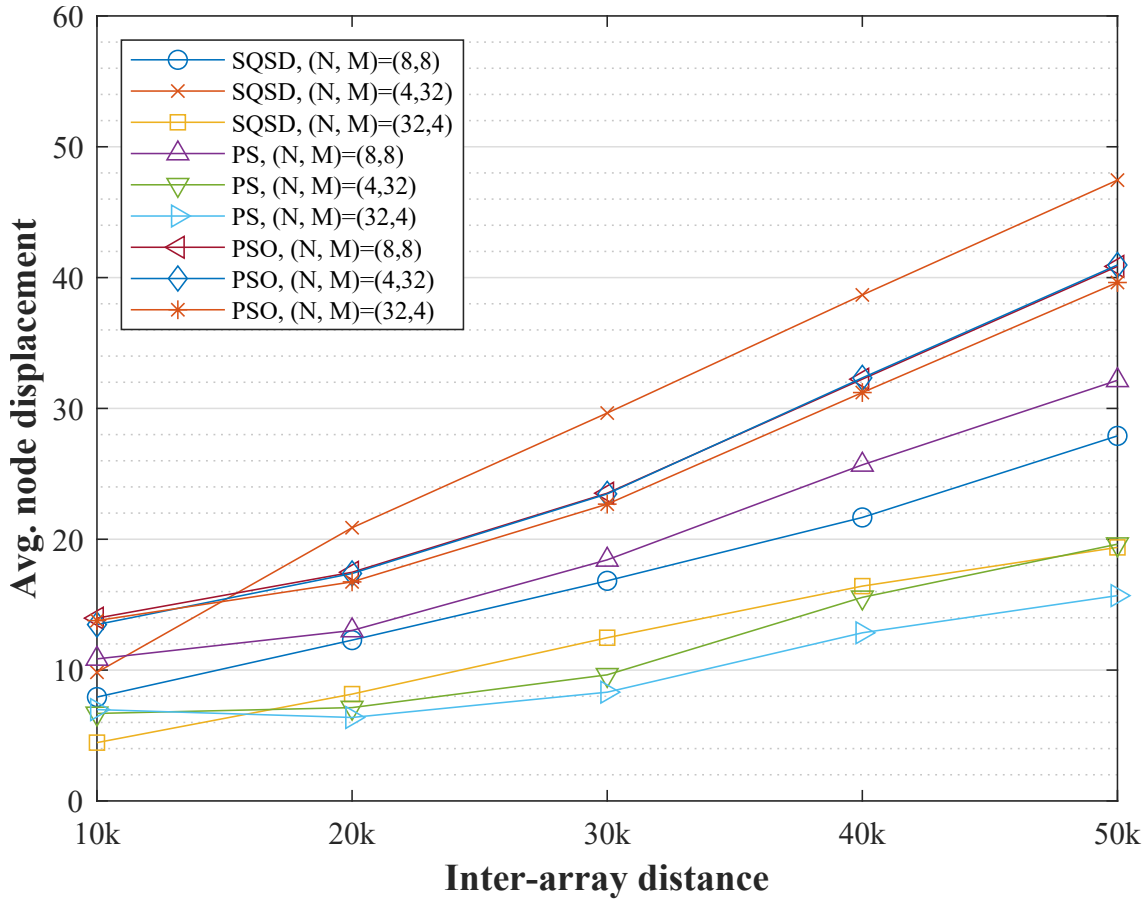

Fig. 38—Upper quartile values of average receiver node displacement versus interarray distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_\phi \approx 10^{-6}$

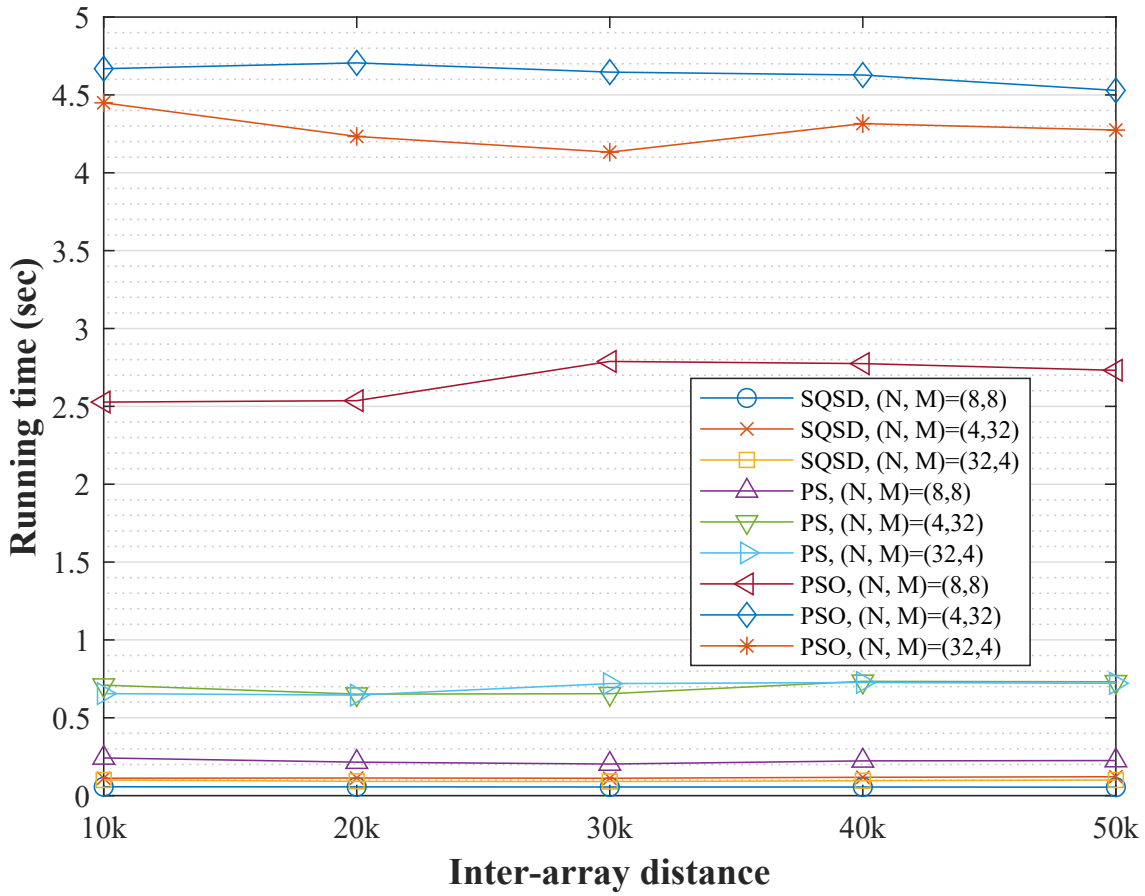

Fig. 39—Upper quartile values of algorithm running time versus interarray distance compared across different optimization algorithms and number of participating nodes; $L_t = L_r = 1k, \rho_{\mathbf{B}} = 1, \rho_{\phi} \approx 10^{-6}$

8. CONCLUSION

In this work, a system model for a combined transmit-and-receive coherent distributed array system is built up. The transmit-and-receive distributed arrays are volumetric aperiodic arrays and are defined by their own respective cubic volumes wherein their constituent array nodes are distributed in a uniform random manner. Array pattern and gain properties are explored via analytical expressions and MATLAB simulations.

First, the distributed transmit array is considered by itself, and its relevant metric is array factor or intensity, which is the normalized array factor squared. Properties of aperiodic volumetric arrays with uniform node distributions are discussed, specifically beamwidth and peak width as well as average intensity. Simulations are used to empirically validate expressions for average intensity and beam width. Periodic cubic arrays that have their array nodes arranged in a regularly spaced three-dimensional matrix, similar to a volumetric phased array but with much greater interelement spacing, are simulated as well.

Next, the receive-side array is added back into the system model, and the relevant metric is coherent communication gain, or CCG, which is essentially the local transmit array intensity summed across all participating receivers. Theoretical upper bound for gain of the combined distriubted array system with N transmitters and M receivers is N^2M, although typical values obtained in simulations are less than half of this upper bound. When there is more than one receiver and they are close enough to the transmit array such that they do not appear as a single point from the perspective of the transmitter, then the beam-pointing direction of the transmit array becomes a relevant variable. The transmit array's beam direction is defined by the azimuth and elevation angles. Discussion focuses on the problem of improving CCG using the positions of the transmit and receive nodes as well as the beam angles of the transmit array. Gradient descent is an attractive approach for the problem in the defined context because of the relative simplicity of gradient descent methods and its ability to find the locally optimal solution, which is superior for minimizing the displacement of network nodes from their initial positions. Square quadratic steepest descent (SQSD), a first-order gradient descent approach that features adaptive step lengths, is modified into a suitable algorithm.

SQSD was tested in simulations alongside pattern search and particle swarm optimization. First, baseline distributions for CCG were collected from uniform random distributions of nodes within array volumes for different array volumes, numbers of nodes, and interarray distances. For simulations with arbitrary node placement, baseline CCG was typically somewhere in the range of $0.1(N^2M)$ to $0.5(N^2M)$. Then, one of the three algorithms under test would be applied to adjust the positions of the nodes and the beam angles in order to maximize CCG for a fixed number of iterations. For the distributed array system configurations that were tested, SQSD performed better than the two global heuristic algorithms in terms of improvement in CCG in the vast majority of circumstances. System gain improvements as a result of node placement optimization ranged from 2 dB up to 6 dB in some circumstances. SQSD was also the fastest-running algorithm of all three, and typically came in second behind pattern search in terms of smallest average node displacement.

REFERENCES

1. J. A. Nanzer, R. L. Schmid, T. M. Comberiate, and J. E. Hodkin, "Open-loop coherent distributed arrays," *IEEE Transactions on Microwave Theory and Techniques* **65**(5), 1662–1672 (2017), doi:10.1109/TMTT.2016.2637899.

2. R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Transactions on Wireless Communications* **6**(5), 1754–1763 (2007), doi:10.1109/TWC.2007.360377.

3. H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Transactions on Signal Processing* **53**(11), 4110–4127 (2005), doi:10.1109/TSP.2005.857028.

4. S. Jayaprakasam, S. K. A. Rahim, and C. Y. Leow, "Distributed and collaborative beamforming in wireless sensor networks: Classifications, trends, and research directions," *IEEE Communications Surveys & Tutorials* **19**(4), 2092–2116 (2017).

5. F. Quitin, A. Irish, and U. Madhow, "Distributed receive beamforming: A scalable architecture and its proof of concept," Proceedings of the 2013 IEEE 77^{th} Vehicular Technology Conference (VTC Spring), Dresden, Germany (IEEE), 2013, pp. 1–5. doi:10.1109/VTCSpring.2013.6692513.

6. F. Quitin, A. T. Irish, and U. Madhow, "A Scalable Architecture for Distributed Receive Beamforming: Analysis and Experimental Demonstration," *IEEE Transactions on Wireless Communications* **15**(3), 2039–2053 (2016), doi:10.1109/TWC.2015.2497687.

7. D. R. Brown III, U. Madhow, M. Ni, M. Rebholz, and P. Bidigare, "Distributed Reception with Hard Decision Exchanges," *IEEE Transactions on Wireless Communications* **13**(6), 3406–3418 (2014), doi:10.1109/TWC.2014.051314.131763.

8. R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Scalable feedback control for distributed beamforming in sensor networks," Proceedings of the Proceedings. International Symposium on Information Theory, 2005. ISIT 2005., 2005, pp. 137–141. doi:10.1109/ISIT.2005.1523309.

9. R. Mudumbai, B. Wild, U. Madhow, and K. Ramchandran, "Distributed beamforming using 1 bit feedback: from concept to realization," Proceedings of the Proceedings of the 44th Allerton conference on communication, control and computation, volume 8, 2006, pp. 1020–1027.

10. R. Mudumbai, D. R. Brown Iii, U. Madhow, and H. V. Poor, "Distributed transmit beamforming: challenges and recent progress," *IEEE Communications Magazine* **47**(2), 102–110 (2009), doi:10.1109/MCOM.2009.4785387.

11. R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Distributed transmit beamforming using feedback control," *IEEE Transactions on Information Theory* **56**(1), 411–426 (2009).

12. N. Chatzipanagiotis, Y. Liu, A. Petropulu, and M. M. Zavlanos, "Controlling groups of mobile beamformers," Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control (CDC) (IEEE), 2012, pp. 1984–1989.

13. F. Quitin, M. M. U. Rahman, R. Mudumbai, and U. Madhow, "A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept," *IEEE Transactions on Wireless Communications* **12**(3), 1418–1428 (2013).

14. D. R. Brown III and H. V. Poor, "Time-slotted round-trip carrier synchronization for distributed beamforming," *IEEE Transactions on Signal Processing* **56**(11), 5630–5643 (2008).

15. G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," Proceedings of the Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004 (IEEE), 2004, pp. 81–88.

16. H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "On the effects of phase estimation errors on collaborative beamforming in wireless ad hoc networks," Proceedings of the Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., volume 3 (IEEE), 2005, pp. iii–657.

17. Y. Shi and Y. E. Sagduyu, "Coherent Communications in Self-Organizing Networks With Distributed Beamforming," *IEEE Transactions on Vehicular Technology* **69**(1), 760–770 (2020), doi:10.1109/TVT.2019.2954653.

18. S. M. Ellison and J. A. Nanzer, "High-Accuracy Multinode Ranging For Coherent Distributed Antenna Arrays," *IEEE Transactions on Aerospace and Electronic Systems* **56**(5), 4056–4066 (2020), doi:10.1109/TAES.2020.2985251.

19. S. M. Ellison, S. Mghabghab, J. J. Doroshewitz, and J. A. Nanzer, "Combined Wireless Ranging and Frequency Transfer for Internode Coordination in Open-Loop Coherent Distributed Antenna Arrays," *IEEE Transactions on Microwave Theory and Techniques* **68**(1), 277–287 (2020), doi:10.1109/TMTT.2019.2943292.

20. J. A. Nanzer, S. R. Mghabghab, S. M. Ellison, and A. Schlegel, "Distributed Phased Arrays: Challenges and Recent Advances," *IEEE Transactions on Microwave Theory and Techniques* **69**(11), 4893–4907 (2021), doi:10.1109/TMTT.2021.3092401.

21. Hanna, Samer and Cabric, Danijela, "Distributed Transmit Beamforming: Design and Demonstration From the Lab to UAVs," *IEEE Transactions on Wireless Communications* **22**(2), 778–792 (2023), doi:10.1109/TWC.2022.3198102.

22. S. Hanna and D. Cabric, "Distributed Transmit Beamforming: Analyzing the Maximum Communication Range," Proceedings of the 2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM), 2022, pp. 136–140. doi:10.1109/SAM53842.2022.9827897.

23. S. Hanna, E. Krijestorac, and D. Cabric, "Destination-Feedback Free Distributed Transmit Beamforming Using Guided Directionality," *IEEE Transactions on Mobile Computing* pp. 1–13 (2022), doi:10.1109/TMC.2022.3188602.

24. F. Rottenberg, M. C. Lee, T. Choi, J. Zhang, and A. F. Molisch, "Robust Non-Coherent Beamforming for FDD Downlink Massive MIMO," Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1–5. doi:10.1109/VTC2020-Spring48590.2020.9128902.

25. G. Sklivanitis, K. Alexandris, and A. Bletsas, "Testbed for non-coherent zero-feedback distributed beamforming," Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 2563–2567. doi:10.1109/ICASSP.2013.6638118.

26. M. V. Lipski, S. Kompella, and R. M. Narayanan, "A Model for Coherent Communication Gain in Distributed Wireless Networks," Proceedings of the 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), 2021, pp. 207–215. doi:10.1109/MASS52906.2021.00036.

27. C. A. Balanis, *Antenna Theory: Analysis and Design*, 4^{th} ed. (John Wiley and Sons, Hoboken, New Jersey, 2016).

28. Y. Shi, S. Ponnaluri, Y. E. Sagduyu, and E. Van Doorn, "Distributed Coherent Group Communications," Proceedings of the MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), 2018, pp. 798–803. doi:10.1109/MILCOM.2018.8599756.

29. K. Buchanan, *A study of aperiodic (random) arrays of various geometries*, PhD thesis (Texas A & M University, 2012).

30. J. Snyman and A. Hay, "The spherical quadratic steepest descent (SQSD) method for unconstrained minimization with no explicit line searches," *Computers & Mathematics with Applications* **42**(1), 169–178 (2001), doi:10.1016/S0898-1221(01)00141-9.

31. J. A. Snyman and D. N. Wilke, *Practical Mathematical Optimization: Basic Optimization Theory and Gradient-Based Algorithms*, 2^{nd} ed. (Springer Int., Cham, Switzerland, 2018).

32. R. Hooke and T. A. Jeeves, "'Direct Search' solution of numerical and statistical problems," *J. of the ACM (JACM)* **8**(2), 212–219 (1961), doi:10.1145/321062.321069.

33. J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of the Proceedings of ICNN'95 - International Conference on Neural Networks, volume 4, 1995, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.

34. R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intelligence* **1**(1), 33–58 (2007), doi:10.1007/s11721-007-0002-0.