Robust Beampattern Synthesis for UAV-Swarm-Based Distributed Beamforming

W. Jeremy Morrison*, Todd E. Humphreys*, Dao A. Ton-Nu*
* *Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin*

ABSTRACT

Distributed and collaborative beamforming (DCBF) combines the inputs or outputs of antennas that are not physically connected and do not share a common clock. In DCBF, the coherent combination of antenna signals hinges on fractional-wavelength positioning knowledge of participating antennas and precise timing synchronization of participating radios. When beamforming with antennas located on dynamic vehicles, the antennas' locations will not be perfectly known and will experience perturbations, motivating a characterization of the resulting beampattern's statistical properties. This statistical analysis has been lacking from prior work. In this paper, prior work's derivations for the expected value and variance of the array gain equation are expanded to include arbitrary Gaussian covariances for each antenna's position. A UAV-based DCBF simulation model is introduced as an example of dynamic distributed arrays and used for numeric statistical beampattern analysis. The difficulties of beampattern synthesis are discussed and an algorithm for beampattern synthesis is presented using particle swarm optimization. It is shown that beamforming is impossible if the standard deviation of antenna position measurement errors exceeds 0.4 wavelengths due to randomly induced phase shifts. If standard deviations are non-identical in the three dimensions, beampattern control will be greater along dimensions with more precise positioning measurements. The number of participating antennas also impacts the beampattern's stability, with the pattern becoming more stable as antennas are added to the array. It is also shown that using carrier-phase differential GNSS (CDGNSS) to determine the relative positions of antennas enables DCBF for C-band frequencies and below. A timing synchronization method is introduced that leverages the fractional-wavelength CDGNSS positioning estimates. This method's coupling of positioning and timing errors is discussed. Lastly, a mitigation technique for vehicle positional perturbations is presented and implemented in simulation, demonstrating successful stabilization of the array's beampattern despite such perturbations.

BIOGRAPHIES

W. Jeremy Morrison (B.S., The University of Texas at Austin) is a first year Ph.D. student in the Department of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin and a graduate research assistant in the UT Radionavigation Laboratory. His current research interests include radionavigation, optimal RF spectrum usage, and secure PNT methods.

Todd E. Humphreys (B.S., M.S., Utah State University; Ph.D., Cornell University) holds the Ashley H. Priddy Centennial Professorship in Engineering in the Department of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin. He is Director of the Wireless Networking and Communications Group and of the UT Radionavigation Laboratory, where he specializes in the application of optimal detection and estimation techniques to positioning, navigation, and timing. His awards include the UT Regents' Outstanding Teaching Award (2012), the NSF CAREER Award (2015), the ION Thurlow Award (2015), the PECASE (NSF, 2019), and the ION Kepler Award (2023). He is Fellow of the Institute of Navigation and of the Royal Institute of Navigation.

Dao A. Ton-Nu (B.S., The University of Texas at Austin) is a former undergraduate researcher of the UT Radionavigation Laboratory in the Department of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin. She is currently working at NASA's Johnson Space Center as a Mission Trajectory Software engineer on the Artemis and Gateway programs.

I. INTRODUCTION

Distributed and collaborative beamforming (DCBF) is a promising method for boosting communications across constellations of autonomous systems. DCBF combines the inputs or outputs of antennas attached to individual nodes in a wireless network to form a distributed phased antenna array — one whose nodes are neither physically connected nor share a common clock. By combining the signals sent or received by each node with an appropriate complex weighting, signal power is spatially boosted or attenuated depending on the direction vector signals' direction of arrival or departure. This distribution of antenna gain across azimuth and zenith angles is called the beampattern. Distributed arrays can form beampatterns advantageous to their tasks. For example, DCBF could enable a swarm of unmanned aerial vehicles (UAVs) to increase its transmit signal

Preprint of the 2024 ION GNSS+ Conference Baltimore, MA, September 17-20, 2024

power or increase the received signal-to-noise ratio from a given direction, compared to what one UAV alone could achieve, maintaining communication links that would have failed otherwise. This paper considers UAV-based distributed arrays as an example of DCBF with dynamic nodes. Many dynamic DCBF schemes have been proposed in research and implemented in practice, from robotic vehicle beamforming [Muralidharan and Mostofi, 2016] to satellite communications [Avellan and Jayasimha, 2018]. Dynamic DCBF differs from stationary distributed arrays in that the nodes are in motion and have positional uncertainty, resulting in unstable beampatterns. Dynamic distributed arrays are more flexible in application than conventional phased arrays or stationary distributed arrays, because antenna positions can be changed. Such flexibility can be of enormous value in communication environments that change over time.

As technologies emerge that increase the feasibility of forming distributed arrays with autonomous vehicles, the need for flexibly synthesizing suitable far-field beampatterns arises. Beamforming for phased arrays is well researched; however, conventional phased arrays have stationary antenna elements that are spaced in regular intervals. A distributed array is unlikely to follow these well-known geometries; rather its antenna geometry will be sparse and non-uniform. Sparse phased arrays, which have antenna elements that are spaced further than a half-wavelength apart, have spatial aliasing effects: signals constructively combine along more than one direction vector, creating a beampattern with potentially many grating lobes — directions along which the array gain is near its maximum value. Even if the beampattern has no grating lobes, a sparse array will always have higher sidelobe levels than what can be achieved by a uniform, dense array. It has, however, been shown that sparse arrays, with carefully-selected antenna positions and complex weightings, can improve upon beampatterns of uniform arrays. Sparsity allows a larger effective aperture to be achieved with fewer antennas, resulting in a sharp mainlobe. The authors of [Trucco and Murino, 1999] optimized positions and weightings for non-uniform sparse arrays in both one and two dimensions using simulated annealing algorithms. They showed improvements in tapering side lobe levels as well as a decrease in number of antenna elements needed to synthesize beampatterns comparable to those of a uniformly-spaced array. But the degree of sparsity in dynamic DCBF arrays is likely to far exceed those presented by [Trucco and Murino, 1999], since movement and safety constraints often separate antennas by multiple wavelengths.

In recent years an increased interest in UAV-based systems has manifested in DCBF literature. The authors of [Egarguin et al., 2020] proposed beamforming using a UAV swarm equipped with passive signal scattering devices. A horizontal beampattern slice towards the intended receiver was shaped by optimizing UAV positions. Since in that work the UAVs only scatter the signal, without producing signals of their own, the phase of emitted signals is tied solely to the distance of the UAVs from the transmitter and thus cannot be actively phase shifted. Researchers from Jilin University published a paper on the use of UAV-based collaborative beamforming to expand and secure terrestrial wireless networks [Sun et al., 2021]. The authors shape the beampattern of the distributed array to minimize the achievable bit-rate of known and unknown eavesdroppers while also optimizing other system variables, such as UAV energy consumption. Variations of swarm intelligence algorithms were used in their optimization. Most recently, [Xu et al., 2023] synthesized beampatterns for a UAV swarm. The UAV locations, heading angles, and complex weighting coefficients were optimized using particle swarm optimization. Nulls were placed along vectors of eavesdroppers, while array gain was maximized along a vector of intended reception.

There remains an important deficiency in the UAV-based DCBF beampattern synthesis literature: Almost all prior work assumes perfect knowledge of antenna locations, and the few papers that allow for probabilistic knowledge do so under the assumption of isotropic antenna position distributions. It has been shown that beampatterns quickly degrade as antenna positions become uncertain and phase noise is added [Moghaddam et al., 2019, Jayaprakasam et al., 2017]. This is especially pertinent to dynamic DCBF because the tracking filters, such as Kalman filters, used to determine the locations of moving vehicles have probabilistic outputs. To create statistically robust beampatterns, the wireless network must know the location of all participating antennas to within a small fraction of a wavelength. These findings motivate the creation of robust beampatterns and techniques to retain beampattern shape despite unknown or undesired antenna movement.

To gain further understanding of the stringent DCBF positioning requirements, it is helpful to think in terms of feasibility regimes. Since fractional-wavelength positioning accuracy is required to form a beampattern that retains its shape, the fraction of positioning error standard deviation σ_p over wavelength λ is a sensible feasibility indicator. If σ_p/λ is low, the statistical shape of the array gain pattern closely follows the one computed by using the mean antenna positions. If this fraction is too high, the gain pattern loses its shape. In the extreme case the carrier phase error becomes uniformly random, making beamforming impossible.

Fractional-wavelength positioning is not the only impediment to implementing DCBF. To successfully combine the signals transmitted or received by each antenna in the distributed array, the RF hardware onboard the vehicles must be synchronized to an exacting degree — to within the time it takes light to travel a small fraction of the signal's wavelength. Recent papers have introduced various protocols for such synchronization [Mohanti et al., 2019, Mohanti et al., 2022, Hanna and Cabric, 2022, Wentz and Chowdhury, 2023]. Almost all of these synchronization methods rely on an external sensor that monitors the carrier phase received from each node and transmits this information back to the nodes such that they can align their phases. Techniques using such an external feedback source are called closed-loop synchronization methods. Synchronizing in this manner has two drawbacks: (1) the array gain can only be controlled along vectors close to the feedback source, and (2) feedback is impossible in

many situations. An open-loop synchronization method for transmit beamforming that does not rely on such external feedback is presented in [Wentz and Chowdhury, 2023]. The authors are able to sufficiently synchronize the nodes in the array, but this technique, while eliminating the phase feedback loop, still requires an outside signal coming from the intended transmission direction, limiting the control over the full beampattern. A well-known choice for synchronizing and determining the location of individual nodes across a network of devices is GNSS. While conventional GNSS capabilities are "far too coarse in position, frequency, and time" [Nanzer et al., 2021] to satisfy the tight positioning and synchronization requirements of distributed arrays, GNSS might still hold the answer: carrier-phase differential GNSS (CDGNSS) satisfies the fractional wavelength positioning requirement for frequencies at C-band and below. CDGNSS also enables a synchronization technique herein called carrier-phase synchronization by GNSS, which is introduced in Section VI of this paper. In combination, CDGNSS and carrier-phase synchronization by GNSS are the most promising open-loop methods that provide the precision required to implement DCBF. This paper explores the statistical properties of DCBF beampatterns when using these methods and implements beampattern stabilizing techniques in simulation. Follow-on work will implement these techniques experimentally using UAVs.

This paper makes the following contributions:

1. Statistical far-field beampattern derivations are expanded to incorporate arbitrary antenna element covariance matrices, and the coupling of position errors with phase synchronization errors is detailed for DCBF with carrier-phase synchronization by GNSS.
2. An analysis of beampattern stability with respect to measurement positioning errors is conducted for dynamic distributed arrays that use CDGNSS for node localization.
3. An optimization technique is presented and implemented for stabilizing UAV-based DCBF beampatterns despite UAV control position errors.

II. UAV-BASED DCBF

Phased antenna arrays and their far-field power gain patterns have been well studied in literature. What follows is a quick introduction to the math of phased arrays and their application to dynamic distributed arrays. Van Trees's book on array processing [Van Trees, 2002] offers derivations of the equations introduced below and a more thorough introduction to the fundamentals of phased arrays.

Due to reciprocity, the gain pattern formed by an array of antennas is the same when receiving and transmitting. Consider an array of N antennas with indices $i \in \{1, \dots, N\}$. To compute the far-field gain along a chosen direction vector $\boldsymbol{v} = [\phi, \theta]^\mathsf{T}$, expressed in azimuth ϕ and zenith θ angles, each signal sent or received is weighted by a complex factor w_i and then summed across all antennas. This complex sum is called the array factor, and its squared magnitude is the signal power gain. Using vector notation, the power gain G can be written as a function of the antenna positions $\boldsymbol{r}_i \triangleq [x_i, y_i, z_i]^\mathsf{T}$, the direction vector $\boldsymbol{v}$, and the stacked vector of complex weights $\boldsymbol{w}$ by

$$G(\boldsymbol{r}_{1,\dots,N}, \boldsymbol{v}) \triangleq \left\| \boldsymbol{w}^\mathsf{H} \left[e^{j\boldsymbol{r}_1^\mathsf{T}\boldsymbol{k}(\boldsymbol{v})}, e^{j\boldsymbol{r}_2^\mathsf{T}\boldsymbol{k}(\boldsymbol{v})}, ..., e^{j\boldsymbol{r}_n^\mathsf{T}\boldsymbol{k}(\boldsymbol{v})} \right]^\mathsf{H} \right\|^2 \tag{1}$$

where $r_{1,\dots,N} = [\boldsymbol{r}_1, ..., \boldsymbol{r}_N]$ is the matrix of antenna positions. The wavenumber

$$\boldsymbol{k}(\boldsymbol{v}) = \frac{2\pi}{\lambda} [\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta)]^\mathsf{T} \tag{2}$$

is the phase-normalized direction vector that the gain is computed along. The complex weight can be split into a magnitude g and a phase shift ψ: $w_i = g_i e^{-j\psi_i}$.

For UAV-based DCBF, the antenna positions are dependent on the location and pose of the UAVs themselves, as well as the antenna's known placement on the UAV body. Consider a model in which there is one antenna per UAV. Let $\boldsymbol{r}^B$ be a vector in the body frame, $\boldsymbol{r}^N$ be a vector in the navigation frame, and $\boldsymbol{Q}^{NB}$ be the rotation matrix from the body frame to the navigation frame. Given a known antenna placement vector in the vehicle body frame $\boldsymbol{r}_A^B$ the antenna's location in the navigation frame $\boldsymbol{r}_A^N$ can be expressed as

$$\boldsymbol{r}_A^N = \boldsymbol{r}_{\text{UAV}}^N + \boldsymbol{Q}^{NB}\boldsymbol{r}_A^B \tag{3}$$

In general this rotation matrix should consider the UAV yaw, pitch, and roll, but if using hovering rotor-crafts, such as most micro aerial vehicles (MAVs) commercially available are, it may suffice to solely consider yaw. The measurement errors of

UAV position and UAV orientation propagate through (3) to the antenna's position error, which is approximated as Gaussian. The ith antenna's position in the navigation frame is distributed as $\boldsymbol{r}_{A,i}^{N} \sim \mathcal{N}(\boldsymbol{r}_i^c, \boldsymbol{R}_{ii})$ and its positional randomness propagates through (1), making the array gain a random variable. The effect of measurement positioning errors on the beampattern is considered in Section IV.

Implementing UAV-based DCBF hinges upon the availability of an accurate navigation and timing solution for all UAVs. The statistical shape of the beampattern depends on the combination of carrier wavelength and the precision of the measured UAV positions and orientations. If the antenna positions cannot be measured to within a fraction of the carrier wavelength, then the expected value of the beampattern will deviate from its desired shape. Another consideration are the control errors: As UAVs drift from their nominal positions and orientations, the array geometry will be perturbed. Just like the quality of the navigation solution, these geometry perturbations arising due to control errors can significantly impact the shape of the beampattern if not mitigated.

It follows that techniques are needed to ensure the beampattern retains its shape despite antenna position measurement and control errors. Furthermore, an understanding of statistical beampattern properties is vital to ensure effective communication links. Section IV gives insights into beampattern statistics and Section V proposes a strategy for retaining the beampattern shape despite these errors.

III. BEAMPATTERN SYNTHESIS

Beampattern synthesis is the process of finding the antenna positions and their complex weightings to create a desired beampattern shape. There are several factors that make dynamic DCBF beampattern synthesis difficult. First, the beampatterns produced by a distributed array are not likely to be symmetric as is the case with most conventional phased arrays. This asymmetry complicates the definition and computation of some beampattern characteristics like beamwidth. For symmetric phased arrays, the beamwidth is usually found by taking an azimuth and zenith slice of the mainlobe, but a distributed array's mainlobe can have a shape that is not fully captured by these two directions. Hence, more effort is needed to constrain and compute beampattern characteristics.

A second factor that complicates dynamic DCBF beampattern synthesis is the large dimensionality: assuming there are N total antennas, with one antenna per vehicle, there are a total of $8N$ parameters (3 from position, 3 from attitude, and 2 from weighting) that influence the beampattern. This order of dimensionality creates a large non-convex optimization problem [Jayaprakasam et al., 2017].

Another challenge is the need to respect the geometry constraints imposed by the vehicles, such as safety and aerodynamic constraints. In a swarm formation, UAVs cannot fly too close together or fly directly in each other's aerodynamic wake. As a result, the distance between antennas in the array will likely exceed the wavelength of the communications signal carrier, creating prominent sidelobes in the beampattern. Varying the complex weighting gains can help suppress these grating lobes, but decreases the absolute maximum gain achievable by the array – a trade-off between shape and gain.

For conventional phased arrays — ones with well-defined antenna geometries and appropriate complex weightings — the shape of the gain pattern is predictable. This is not the case for the arbitrary weights and antenna positions seen in dynamic distributed arrays. To determine the full shape of the beampattern, one must compute the gain along a discretized set of vectors filling 3D space. Some of the characteristics used to define an array's desired beampattern include the mainlobe beamwidth, local and global sidelobe levels, and nulling vectors. The optimal shape of the beampattern varies by application and environment. Consider the exemplary case of transmit beamforming. For a single transmission vector one might form a beampattern that maximizes the transmission gain and not enforce any further requirements. For multiple transmission vectors, a beampattern with multiple high-gain vectors is needed. For the case of eavesdroppers and interference concerns, the beampattern must incorporate areas of sidelobe tapering or nulling vectors. If there is appreciable uncertainty about the transmission or nulling vectors, the beampattern must be shaped to encompass the statistics of these vectors. Clearly, there are many different beampattern shapes that are desirable in different use cases and environments.

Due to the many degrees of freedom in beampattern synthesis, the optimization algorithm of choice must be able to search a vast combination of antenna positions and complex weightings on a non-convex landscape. Stochastic algorithms are a class of easily-accessible algorithms suited to the task. Similar to [Xu et al., 2023], the present work adopts particle swarm optimization (PSO) to mold beampatterns by optimizing the antenna positions and complex weights.

In PSO each particle is defined by a matrix $\boldsymbol{P}_j$ containing all optimization variables. Considering the case of N UAVs with one antenna per UAV participating in the array, $\boldsymbol{P}_j$ is defined as

$$\boldsymbol{P}_j \triangleq \begin{bmatrix} x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \\ z_1 & z_2 & \dots & z_N \\ \alpha_1 & \alpha_2 & \dots & \alpha_N \\ \beta_1 & \beta_2 & \dots & \beta_N \\ \gamma_1 & \gamma_2 & \dots & \gamma_N \\ g_1 & g_2 & \dots & g_N \\ \psi_1 & \psi_2 & \dots & \psi_N \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\b\\c\end{matrix}}\right\} \\ \left.\vphantom{\begin{matrix}a\\b\\c\end{matrix}}\right\} \\ \left.\vphantom{\begin{matrix}a\\b\end{matrix}}\right\} \end{matrix} \begin{matrix} \text{UAV position coordinates} \\ \text{UAV pose euler angles} \\ \text{complex weight gains and phases} \end{matrix} \tag{4}$$

PSO uses a large number of such particles to explore the cost landscape in an iterative process. During one iteration all particles are evaluated with a cost function and then adjusted based on each particle's previous configuration, personal best configuration, and the swarm's best configuration. Equation (5) shows how a particle is mutated by its velocity $\boldsymbol{V}_j$, which has the same dimensions as the particle, at the end of an iteration. Before being added to the current particle configuration, $\boldsymbol{V}_j$ is altered by information attained from the swarm and randomized. Let $\boldsymbol{P}_j^*$ be the best previously found configuration for the jth particle and $\boldsymbol{P}_\text{g}^*$ be the best particle configuration for the entire particle swarm. Then the update formulas for $\boldsymbol{V}_j$ and $\boldsymbol{P}_j$ are

$$\begin{aligned} \boldsymbol{V}_{j,\text{new}} &= c_\text{inertia}\boldsymbol{V}_j + c_\text{personal} r_1(\boldsymbol{P}_j^* - \boldsymbol{P}_j) + c_\text{global} r_2(\boldsymbol{P}_\text{g}^* - \boldsymbol{P}_j) \\ \boldsymbol{P}_{j,\text{new}} &= \boldsymbol{P}_j + \boldsymbol{V}_{j,\text{new}} \end{aligned} \tag{5}$$

Pseudocode for PSO, as applied to this paper's distributed beamforming case, is shown in Algorithm 1. At the start of the algorithm $\boldsymbol{V}_j$ and $\boldsymbol{P}_j$ are initialized with randomized values within set ranges. Performance of the algorithm depends on the hyperparameters $c_\text{inertia}, c_\text{personal}, c_\text{global}, \delta_{c_\text{inertia}}$, coefficients that affect the balance of exploration and exploitation during the optimization. Each iteration, the random generated numbers $r_1, r_2 \sim \mathcal{U}(0,1)$ ensure the search is stochastic, aiding in the exploration of the search space. The number of iterations N_iter and swarm size N_swarm need to be chosen such that a near-optimal solution is likely to be returned.

Algorithm 1 Particle Swarm Beampattern Synthesis

Input: $N_\text{swarm}, N_\text{iter}, c_\text{inertia}, c_\text{personal}, c_\text{global}, \delta_{c_\text{inertia}}$
Output: $\boldsymbol{P}_\text{g}^*$

$\mathcal{P} \leftarrow \{\boldsymbol{P}_1, \boldsymbol{P}_2, \dots, \boldsymbol{P}_{N_\text{swarm}}\}$; $\mathcal{P}^* \leftarrow \mathcal{P}$ ▷ **Randomly initialize particles**
$\mathcal{V} \leftarrow \{\boldsymbol{V}_1, \boldsymbol{V}_2, \dots, \boldsymbol{V}_{N_\text{swarm}}\}$
$f_i^* \leftarrow \infty$; $f_\text{g}^* \leftarrow \infty$
for iter $= 1 \rightarrow N_\text{iter}$ **do**
  $c_\text{inertia} \leftarrow \delta_{c_\text{inertia}} c_\text{inertia}$
  for $j = 1 \rightarrow N_\text{swarm}$ **do** ▷ **Evaluate and update best particles**
    $f_j \leftarrow$ `costFunction`$(\mathcal{P}\{j\})$
    if $f_j < f_j^*$ **then**
      $f_j^* \leftarrow f_j$
      $\mathcal{P}^*\{j\} \leftarrow \mathcal{P}\{j\}$
      if $f_j < f_\text{g}^*$ **then**
        $f_\text{g}^* \leftarrow f_j$
        $\boldsymbol{P}_\text{g}^* \leftarrow \mathcal{P}\{j\}$
      end if
    end if
  end for
  for $j = 1 \rightarrow N_\text{swarm}$ **do** ▷ **Permute particles**
    $r_1 \leftarrow \text{rand}(\mathcal{U}(0,1))$; $r_2 \leftarrow \text{rand}(\mathcal{U}(0,1))$
    $\mathcal{V}\{j\} \leftarrow c_\text{inertia}\mathcal{V}\{j\} + c_\text{personal} r_1(\mathcal{P}^*\{j\} - \mathcal{P}\{j\}) + c_\text{global} r_2(\boldsymbol{P}_\text{g}^* - \mathcal{P}\{j\})$
    $\mathcal{P}\{j\} \leftarrow \mathcal{P}\{j\} + \mathcal{V}\{j\}$
  end for
  iter $\leftarrow$ iter + 1
end for
return $\boldsymbol{P}_\text{g}^*$

As with all meta-heuristic algorithms, the formulation of a suitable cost function is pivotal to the success of the optimization.

The cost function reduces each particle's beampattern shape to a scalar value based on the desired beampattern requirements. Common requirements include mainlobe directions, half-power mainlobe width, regional or global sidelobe levels, and nulling directions. All of these requirements involve sampling the array gain in (1) along a selection of vectors. More extensive requirements, such as global sidelobe levels, are computationally expensive because the gain must be computed along many vectors with a sufficient spatial resolution. Unless the array is large it is recommended to only attempt local sidelobe tapering, since the sparsity of the distributed array naturally creates prominent sidelobes which are difficult to suppress in all directions. Furthermore, a cost function must weight features in an appropriate manner. If, for example, the cost for a null eclipses that of the mainlobe it might prohibit the formation of a suitable mainlobe beam in the optimization process.

Other system constraints, such as maximum UAV range or minimal inter-UAV spacing, may be easily added to the cost as additional terms, nudging the explored array geometries away from invalid configurations. Other constraints should be added to the PSO algorithm itself. In fact, if the array geometry is not sufficiently constrained, there are infinitely many solutions: A uniform translation of all antennas in the array does not affect the far-field beampattern. Hence, it is prudent to constrain one antenna to be stationary, which constrains one UAV in the swarm in both translation and rotation, by setting the appropriate PSO velocity terms to zero. There is also the possibility of the ith antenna having a weighting magnitude g_i near zero, effectively removing its influence on the array's beampattern. Any combination of positions and phases assigned to this antenna again yields an infinite solution space.

IV. THE BEAMPATTERN AS A RANDOM VARIABLE

Dynamic distributed arrays are formed with antennas attached to vehicles whose positions are tracked by navigation algorithms. The positions and orientations of the vehicles, and by extension the positions of the antennas onboard, are described statistically by the outputs of such navigation solutions. Therefore, it is important to understand and quantify the impact of the uncertainty in the array antennas' positions on the beampattern. This section quantifies the mean and variance of DCBF beampatterns under antenna position measurement uncertainty and analyzes how both mainlobe gain and nulling gain are impacted by changing uncertainty parameters.

1. Analytical Mean and Variance for Gaussian Position Errors

By examining (1) it is evident that an array's beampattern can be perturbed in two ways: by fluctuations in the antenna complex weights and deviations in antenna positions. Due to the dynamic nature of UAV movement and the statistical output of position-tracking algorithms, this paper focuses on the effect of antenna position errors on the array's beampattern.

The mean [Van Trees, 2002] and the variance [Moghaddam et al., 2019] of the array gain function (1) have been derived for Gaussian errors in complex weights and positions. In these works the authors assumed identically distributed errors in the x, y, and z dimensions for the entire array. However, these equations cannot account for position covariance matrices that are commonly seen by GNSS based positioning algorithms – ones that have a larger variance in the vertical than the horizontal. Since the proposed CDGNSS positioning technique for the UAVs has such covariance matrices as its output, this paper expands the derivations in [Van Trees, 2002, Moghaddam et al., 2019] by assuming arbitrary covariance matrices for the array's antenna positions. Further, the presented equations allow for each antenna to have a unique covariance matrix, enabling the analysis of arrays with non-homogeneous uncertainties across the vehicles. In the following two equations the covariance matrix of the ith antenna's position measurement error is denoted as $\boldsymbol{R}_{ii}$, the mean 3D antenna positions are denoted $\boldsymbol{r}^c$, and δ_{ij} is the Kronecker delta equaling one if $i = j$ and zero otherwise. The wavenumber $\boldsymbol{k}(\boldsymbol{v})$ is written as $\boldsymbol{k}$ for notational brevity.

$$\mathbb{E}\left[G(\boldsymbol{r}_{1,\ldots,N}, \boldsymbol{v})\right] = \sum_{i=1}^{N}\sum_{l=1}^{N} w_i w_l^* e^{-j\boldsymbol{k}^\mathsf{T}(\boldsymbol{r}_l^c - \boldsymbol{r}_i^c)} e^{-\frac{1}{2}\boldsymbol{k}^\mathsf{T}\{(\boldsymbol{R}_{ii}+\boldsymbol{R}_{ll})(1-\delta_{il})\}\boldsymbol{k}} \tag{6}$$

$$\begin{aligned}\mathbb{V}\left[G(\boldsymbol{r}_{1,\ldots,N}, \boldsymbol{v})\right] = &\sum_{i=1}^{N}\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} w_i w_l^* w_m w_n^* e^{-j\boldsymbol{k}^\mathsf{T}(\boldsymbol{r}_i^c - \boldsymbol{r}_l^c + \boldsymbol{r}_m^c - \boldsymbol{r}_n^c)} \\ &e^{-\frac{1}{2}\boldsymbol{k}^\mathsf{T}\{\boldsymbol{R}_{ii}(1-\delta_{il}-\delta_{in}+\delta_{im})+\boldsymbol{R}_{ll}(1-\delta_{il}-\delta_{lm}+\delta_{ln})+\boldsymbol{R}_{mm}(1-\delta_{lm}-\delta_{mn}+\delta_{im})+\boldsymbol{R}_{nn}(1-\delta_{in}-\delta_{mn}+\delta_{ln})\}\boldsymbol{k}} \\ &- \mathbb{E}\left[G(\boldsymbol{r}_{1,\ldots,N}, \boldsymbol{v})\right]^2\end{aligned} \tag{7}$$

Note that the cross-covariance between antenna position errors $\boldsymbol{R}_{ij}$ is assumed to be zero, but no further assumptions on the covariances are made. Also note that these equations ideally produce real valued gain numbers, because complex components cancel out. Due to computational precision they might produce values with small complex components, which can be neglected.

Equations (6) and (7) have been validated by comparison to Monte Carlo simulations across multiple combinations of array geometries, gain computation vectors, and antenna position covariance matrices. An alternative form for the mean, which eliminates complex numbers and reduces computation complexity, is shown below. Recall that complex weights may be separated into magnitude and phase $w_i = g_i e^{-j\psi_i}$, and let $\boldsymbol{g}$ be the stacked vector of weighting magnitudes.

$$\mathbb{E}\left[G(\boldsymbol{r}_{1,\ldots,N}, \boldsymbol{v})\right] = \boldsymbol{g}^{\mathsf{T}}\boldsymbol{g} + 2\sum_{i=1}^{N}\sum_{l>i}^{N} g_i g_l e^{-\frac{1}{2}\boldsymbol{k}^{\mathsf{T}}\{\boldsymbol{R}_{ii}+\boldsymbol{R}_{ll}\}\boldsymbol{k}} \cos(\psi_i - \psi_l - \boldsymbol{k}^{\mathsf{T}}(\boldsymbol{r}_i - \boldsymbol{r}_l)) \tag{8}$$

It is also important to note that while the antenna position errors are Gaussian distributed, the gain errors are not. Hence, the first two moments of the gain do not suffice to fully describe its distribution. The confidence intervals, shown in results of numerical simulations in a later subsection, were found from kernel density estimates of the gain's distributions.

2. Simulation Model

The simulation case study in this paper is modeled to parallel a real-world experimental setup that may be explored in follow-on work. The simulation assumes each UAV has one antenna and adapts the MAV model, which uses the simplified version of (3) that only considers the MAV's yaw.

To simulate CDGNSS position measurement errors, Gaussian errors are added to each MAV's position. Each MAV in the swarm is assigned errors described by the same covariance matrix. The impact of varying these measurement errors in magnitude and distribution is explored in the next subsection. Real-world empirical flight data of testbed UAVs shows the following average values for estimation error covariance using CDGNSS:

$$\boldsymbol{R}_{ii} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} = \begin{bmatrix} 0.005^2 & 0 & 0 \\ 0 & 0.005^2 & 0 \\ 0 & 0 & 0.011^2 \end{bmatrix} \text{m} \tag{9}$$

Beampatterns were synthesized with two features: a desired mainlobe direction with a corresponding beamwidth and a nulling vector. This beampattern structure might be desired in environments with one main noise or interference source to maximize received SINR. Using the simplified MAV orientation model of yaw α only, each PSO particle in the beampattern synthesis optimization takes the following form:

$$\boldsymbol{P}_i = \begin{bmatrix} x_1 & x_2 & \ldots & x_N \\ y_1 & y_2 & \ldots & y_N \\ z_1 & z_2 & \ldots & z_N \\ \alpha_1 & \alpha_2 & \ldots & \alpha_N \\ g_{11} & g_{12} & \ldots & g_{1N} \\ \psi_{11} & \psi_{12} & \ldots & \psi_{1N} \end{bmatrix} \tag{10}$$

The cost function used in Algorithm 1 was a weighted squared sum of a mainlobe beamwidth cost term f_{HPBW} and a null cost term f_{null}.

$$f = \left[c_1 f_{\text{HPBW}} + c_2 f_{\text{null}}\right]^2 \tag{11}$$

Here f_{HPBW} is computed by bisecting the mainlobe six times and finding the difference between the desired beamwidth angles and the actual beamwidth angles, while the null cost is computed by finding the gain along the nulling vector. To give these terms comparable importance on the overall optimization cost, they are weighted by the coefficients c_1 and c_2.

3. Analysis of Mainlobe and Nulling Gain

As discussed in the introduction, the normalized standard deviation of antenna positions σ_p/λ is a valuable indicator of DCBF feasibility. Consider a beampattern shape in which a desired mainlobe beamwidth is set and a null is placed. Further, let the variance for all antennas in the array be identical and each antenna have a perfectly spherical uncertainty – that is the same standard deviation along the x, y, and z axes. How does the beampattern stability evolve when varying the magnitude of σ_p/λ? Figure 1 shows the mean array gain, normalized such that 0 dB indicates perfect constructive phase alignment, along the nulling vector and the mainlobe vector as a function of σ_p/λ. As the uncertainty in position increases, the difference between null gain and mainlobe gain shrinks until they finally converge, validating the intuition that the mean beampattern is effectively isotropic

when the phase shifts induced by antenna position measurement errors overpower the beampattern-shaping complex weights. The asymptote of the convergence, which varies with the number of antennas in the array and their relative weighting magnitude, was found to be $10\log_{10}(\boldsymbol{w}^{\mathsf{H}}\boldsymbol{w})$, implying that the double summation in (8) goes to zero with large covariance matrices. Indeed, the term $e^{-\frac{1}{2}\boldsymbol{k}^{\mathsf{T}}\{\boldsymbol{R}_{ii}+\boldsymbol{R}_{ll}\}\boldsymbol{k}}$ will dominate as antenna variances increase, driving the summation towards zero.

The mainlobe gain is less sensitive to changes in variance than the null gain, which can be explained by examining the gain in a linear scale. In a linear scale, the difference between a power gain from -5dB to 0dB is 6.8×10^{-1}, while it is merely 6.8×10^{-3} from -25 dB to -20 dB. Therefore, the uncertainty of antenna locations affects null gain much more than the mainlobe gain on a logarithmic scale. It is difficult to set a feasibility limit value of σ_p/λ, since gain requirements vary by application. Some notable values are $\sigma_p/\lambda\approx0.13$, at which the mean gain in the direction of the mainlobe vector falls below the -3dB used to find the half power beamwidth, and $\sigma_p/\lambda\approx0.4$, at which the beampattern is effectively isotropic. Setting $\sigma_p=0.005$, an empirical value measured from CDGNSS positioning for UAVs as in (9), the maximum carrier frequency such that $\sigma_p/\lambda<0.13$ is 8.8 GHz, covering a valuable range of the spectrum.

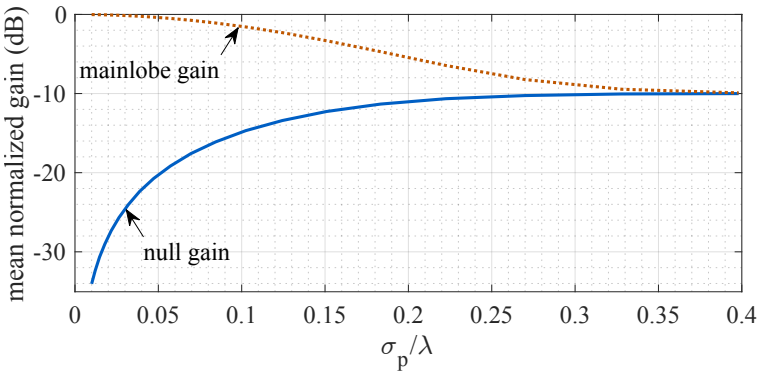

Figure 1: Plot of the mainlobe array gain and the null array gain as a function of normalized antenna positional standard deviation for an array with $N=10$. Null gain is more sensitive changes in positional uncertainty than the mainlobe gain when $\sigma_p/\lambda<0.15$. Gain for mainlobe and null converges at $\sigma_p/\lambda>0.4$, indicating an isotropic beampattern.

An array's beampattern stability also is affected by the number of antennas participating in the array: More antennas make nulls statistically deeper. This relationship seems to arise from the law of large numbers. In a larger array with many antennas there may be a greater likelihood that for each antenna that is not at an optimal position there are other antennas that cancel out the induced phase shift. Adding more antennas also decreases the influence of each individual antenna on the array's beampattern, causing statistical antenna position outliers to affect the beampattern less, which would also contribute to deeper nulls. Figure 2 shows the decrease in mean null gain as more antennas are added. Due to the high optimization cost at large number of antennas the figure only shows results for up to 100 antennas, but note that the relationship between null gain and number of antennas seems to be linear on a log-log scale.

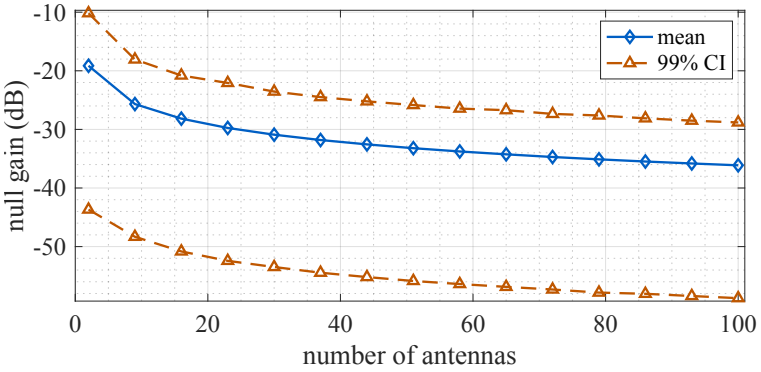

Figure 2: Normalized null array gain as a function of number of antennas participating in the distributed array. With increasing number of antennas the null gets statistically deeper.

A natural question one might ask is if there exist antenna geometries and weighting coefficients that are better than others for robustness against antenna position measurement errors. Synthesizing beampatterns by Algorithm 1 with the same requirements

on mainlobe and null revealed the existence of many unique near-optimal combinations of weighting coefficients and antenna geometries. Of the many near-optimal antenna geometries, there do not seem to be ones that are inherently more robust to antenna positional uncertainty than others. However, the weighting gains do impact beampattern stability: identical weighting gains across the array result in the most stable pattern. When antennas are weighted with different magnitudes the array is effectively using less antennas in forming the beampattern, which, per the findings in Figure 2, decreases the beampattern stability. An extreme example of this is when one antenna is completely deweighted and does not contribute to the beampattern at all.

If one is using GNSS-based positioning, the standard deviation is almost always higher in the vertical than in the horizontal. The possibility arises, depending on the quality of the positioning solution and the beamforming wavelength, of the vertical normalized standard deviation being outside the feasibility range while the horizontal one is inside the feasibility range. Beamforming near the horizontal plane is still possible! Movement of the array's antennas in the plane orthogonal to the beamforming vector does not affect the stability of the beampattern's gain along that vector. This disassociation of the array's gain from the orthogonal antenna movement is due to far-field assumptions: signals are modeled as plane waves. Antennas see the same signal phase anywhere in the signal propagation plane at the instant it is passing through, therefore any movement, or uncertainty in position, of antennas in this plane does not affect the coherent combination of the antenna's inputs or outputs. This is also visible in (1): positions of antennas and their associated errors, are projected onto the phase-normalized vector of gain computation $\boldsymbol{r}_i^{\mathsf{T}}\boldsymbol{k}(\boldsymbol{v})$, rendering any movement orthogonal to $\boldsymbol{v}$ irrelevant. Hence, an array's beampattern in which the antenna's are located using GNSS might be stable along vectors near the horizontal plane but loose stability along vectors with a significant vertical component. Fig. 3 showcases how the stability of the same null varies when the null is placed along the x, y, and z axes in a system with non-uniformly distributed positional uncertainty. In this example nulls pointing along or near the x-axis are much deeper than nulls near the z-axis due to the smaller variance in the x direction.

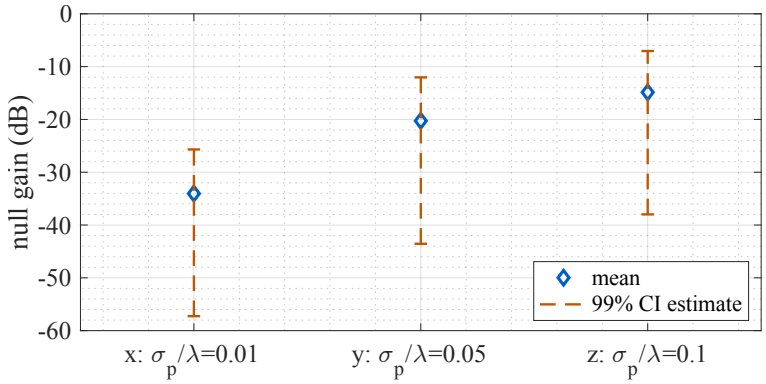

Figure 3: Normalized array gain for a null along each of the principal axes of a ten UAV array with non-spherical antenna uncertainties. Along axes with less positional uncertainty, the null gain is deeper, showing that the beampattern's stability along a vector is dependent on the antenna's positional uncertainties in the vectors direction.

V. MITIGATION OF VEHICLE CONTROL ERRORS

During UAV-based beamforming, and other dynamic beamforming scenarios, vehicle positions drift over time based on the vehicle's control system and environmental disturbances. For the MAVs used in this paper's simulation, these factors are reflected in their hovering behavior and the ability to measure the positions of the vehicles far exceeds the ability to control them. This section quantifies the effect of UAV hovering on the beampattern's stability and evaluates the efficacy of a proposed mitigation technique.

A simulation was devised to quantify the effect of hovering on the beampattern: Similarly to Section IV.3 a beampattern was synthesized with mainlobe and nulling vectors. To realistically simulate hovering behavior, estimated position and pose data of real-world hovering MAVs was collected and mean-shifted to the synthesized vehicle positions and poses in the simulation. During the simulation, 3dB mainlobe beamwidth and null gain were recorded while still accounting for antenna positional uncertainties in (9). Using ten UAVs the simulation covered 66 seconds of hovering, during which, with moderate winds, the MAVs drifted as far as one wavelength from the commanded position.

In an effort to mitigate the effects of the hovering motion on the array's beampattern, a re-optimization of the array's complex weighting vector is introduced. This re-optimization is performed at each time newly-estimated position and pose data is provided by the position-tracking filter and uses the same particle swarm optimization algorithm 1 that was used for the original beampattern synthesis. By setting the starting particle values for the UAV positions and poses equal to the estimates of the

position-tracking filter and setting the corresponding particle velocity components to zero, the algorithm only optimizes the complex weighting vector.

Figure 4 plots the null gain and mainlobe beamwidth statistics during the hovering simulation for both the original and continuously re-optimized weighting vectors. The null gain when re-optimizing weights shows much improvement in both mean depth and confidence interval range from the original weighting. Looking at the bore-sight view of the mainlobe it is clear that the 3dB beamwidth for the original weighting fluctuates a lot, even dipping below the -3dB mark used to compute it. By comparison, the re-optimization procedure stabilized the mainlobe such that its mean is near the requested two degree diameter and is tightly bounded by its 99% confidence intervals.

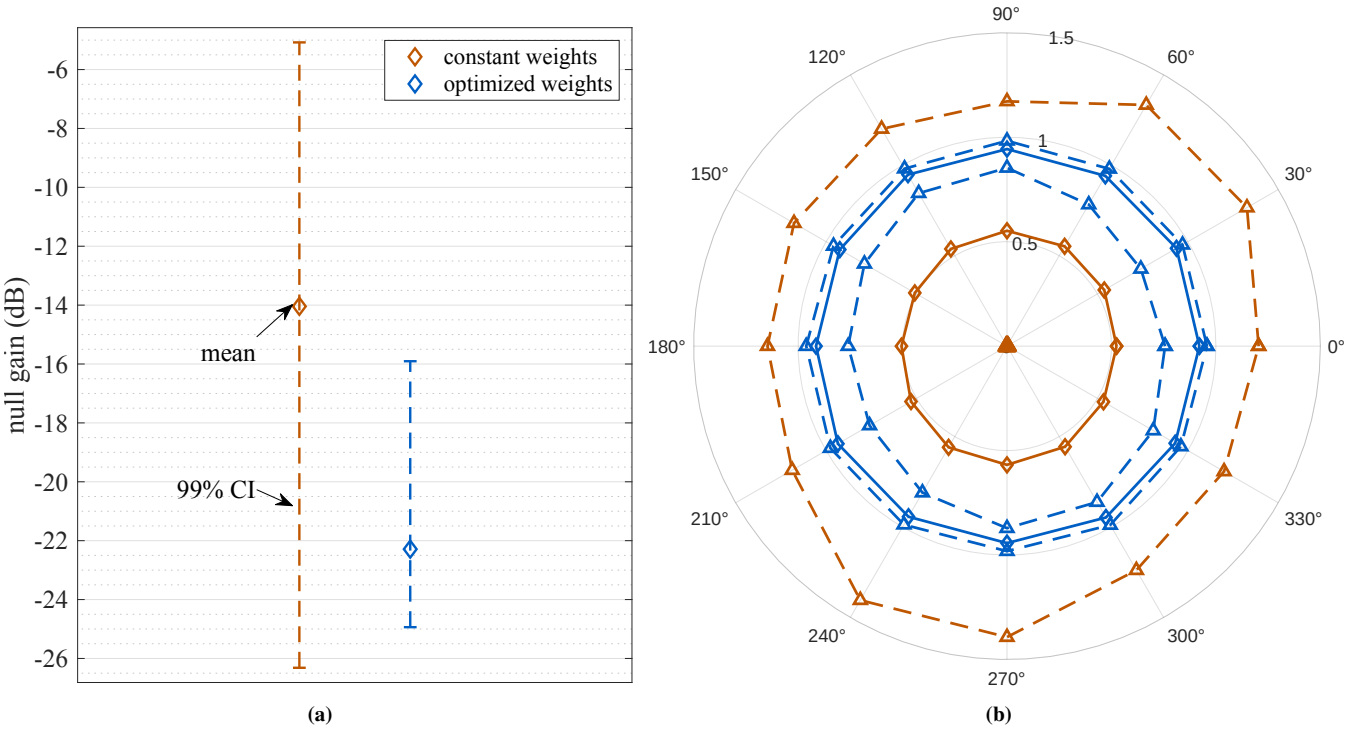

Figure 4: Beampattern stability during hovering: (a) The nulling gain's mean and 99% confidence interval and (b) A bore sight view of the mainlobe, showing the mean and 99% confidence interval 3dB beamwidths.

VI. TIMING SYNCHRONIZATION

In Section IV the beampattern's stability was analyzed with respect to the uncertainty of the antenna's positions, while assuming perfect timing synchronization across the array. In this section a timing synchronization method titled carrier-phase synchronization by GNSS is introduced and the impact of using the proposed synchronization method on the beampattern's stability is discussed.

To synchronize the timing and phases across the distributed array's radios a method called carrier phase synchronization by GNSS is used. In this technique, a GNSS satellite's signal is designated as the 'clock signal' which all radios synchronize to. By using the fractional-wavelength positioning estimate achieved by CDGNSS to compute an expected beat carrier-phase of the clock signal $\bar{\theta}_i$ at each antenna and taking the difference to the beat carrier-phase measured by each antenna $\tilde{\theta}_i$, a phase adjustment for the ith antenna is found, from which a timing adjustment to the local clock is derived.

$$\tilde{\theta}_i - \bar{\theta}_i = 2\pi(f_c + f_{Di})\Delta t_i \tag{12}$$

Here f_c and f_{Di} are, respectively, the clock signal's center frequency and the Doppler frequency shift seen by the ith antenna. Note that the clock signal should not be used in computing the CDGNSS position, since this would introduce complicated cross-errors between position and time.

The computed timing adjustment Δt_i is not able to differentiate between phase errors caused by drift of the local clock and those due to deviation from the estimated position of the antenna; Phase shift errors that are coupled to antenna position errors

have been introduced. To further understand the nature of this coupling consider Figure 5, where $\boldsymbol{e}$ is the position error of the antenna, $\hat{\boldsymbol{c}}$ is the direction of the clock signal, and $\hat{\boldsymbol{v}}$ is the direction array gain is computed along. Applying the timing adjustment to the antenna's radio corrects its clock drift, but also introduces a phase shift proportional to the projection of $\boldsymbol{e}$ onto $\hat{\boldsymbol{c}}$. Defining $\boldsymbol{k}_c \triangleq \frac{2\pi(f_c+f_{Di})}{v_{\text{signal}}}\hat{\boldsymbol{c}}$ to be the wavenumber for the clock signal, the effects of this phase shift can be modeled by adjusting (8) as follows.

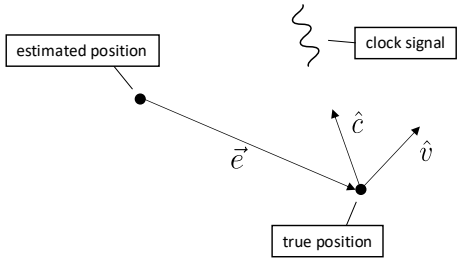

Figure 5: True and estimated positions during carrier-phase GNSS synchronization. Here $\hat{\boldsymbol{c}}$ is the direction of the clock signal, $\boldsymbol{e}$ is the position error vector of the antenna, and $\hat{\boldsymbol{v}}$ is the direction of array gain computation.

$$\mathbb{E}\left[G(\boldsymbol{r}_{1,\ldots,N}, \boldsymbol{v})\right] = \boldsymbol{g}^{\mathsf{T}}\boldsymbol{g} + 2\sum_{i=1}^{N}\sum_{l>i}^{N} g_i g_l e^{-\frac{1}{2}(\boldsymbol{k}-\boldsymbol{k}_c)^{\mathsf{T}}\{\boldsymbol{R}_{ii}+\boldsymbol{R}_{ll}\}(\boldsymbol{k}-\boldsymbol{k}_c)} \cos(\psi_i - \psi_l - \boldsymbol{k}^{\mathsf{T}}(\boldsymbol{r}_i - \boldsymbol{r}_l)) \tag{13}$$

Expanding the difference of wavenumbers $(\boldsymbol{k} - \boldsymbol{k}_c) = 2\pi(\frac{\hat{\boldsymbol{v}}}{\lambda_{\text{BF}}} - \frac{\hat{\boldsymbol{c}}}{\lambda_c'})$, where λ_c' is the clock signal's wavelength after accounting for Doppler shift, shows that if $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{c}}$ are close together the carrier phase synchronization by GNSS helps eliminate uncertainty in the gain along $\hat{\boldsymbol{v}}$ due to position errors. Conversely, if the vectors are far apart the beampattern's stability along $\hat{\boldsymbol{v}}$ suffers. It is also evident that if $\hat{\boldsymbol{v}}$ is not parallel to $\hat{\boldsymbol{c}}$ the positional uncertainty of antennas in directions other than $\hat{\boldsymbol{v}}$ starts affecting the beampattern's stability along $\hat{\boldsymbol{v}}$.

It is therefore advantageous to select a clock signal in that direction which requires the most stability in the array's beampattern. As discussed, nulling is one such example that requires a high degree of stability – even marginal increases in beampattern stability along a nulling vector can yield a noticeable improvement in null depth.

VII. CONCLUSION

Dynamic distributed arrays are formed with RF hardware located on multiple vehicles, whose positional uncertainty and positional perturbations impact the beampattern of the array. UAV-based distributed and collaborative beamforming was introduced as an example of dynamic distributed arrays. A discussion on the difficulties of beampattern synthesis, the process of finding antenna positions and complex weights that form a suitable beampattern, revealed that stochastic optimization algorithms are best suited to the task. An algorithm for beampattern synthesis by particle swarm optimization and ways of including constraints, such as inter-vehicle spacing, were presented.

The unknown and dynamic nature of antenna positions in a dynamic distributed array impact the beampattern stability, motivating a statistical representation of the beampatterns. Analytical derivations for the expected value and variance of the array gain equation were expanded to account for arbitrary antenna position measurement covariance matrices. A statistical study of dynamic distributed array beampatterns with respect to Gaussian positional measurement errors of antenna positions was presented. The simulation study showed beamforming is impossible if the positions of the antennas cannot be determined to a fraction of the wavelength. When antenna positional measurement error magnitude exceeds $\sigma_p/\lambda > 0.4$, the expected value of the array gain converged on a constant value for all directions, creating an isotropic mean beampattern. With the positioning precision of carrier-phase differential GNSS (CDGNSS) beamforming was shown to be feasible at and below C-band frequencies. Beampatterns formed by many antennas exhibited greater stability than those formed by few antennas, which was traced back to the statistical law of large numbers. An analysis of beamforming performance with antennas who's positional measurements have uneven magnitudes in each dimension showed that higher beampattern control can be achieved along dimensions with lower positional uncertainty. For vehicles located by CDGNSS this indicates that the beampattern will always be more stable near the horizontal plane.

A technique for mitigating the effect of vehicle perturbations on the array's beampattern, for situations where the positioning measurement accuracy is greater than the vehicle's control error magnitude, was presented. This technique was implemented in a UAV-based array hovering simulation. By re-synthesizing the complex weighting vector at each time newly-estimated

UAV positions were available from the carrier-phase differential positioning algorithm, the beampattern remained more stable throughout the UAV hovering motions.

Finally, a timing synchronization method, carrier-phase synchronization by GNSS, was introduced. The impact of the synchronization signal's angle of arrival on the beampattern was discussed and it was found that the beampattern will be more stable along angles that are near the synchronization signal's arrival direction. Lastly it was shown that, when using carrier-phase synchronization by GNSS, position errors are coupled to phase errors.

Further research efforts will include a statistical characterization of beampatterns formed with multiple antennas per vehicle, which introduces non-zero cross-covariance matrices between antenna positions, as well as experimental demonstration of CDGNSS enabled receive beamforming at the L_1 GNSS wavelength.

REFERENCES

[Avellan and Jayasimha, 2018] Avellan, A. and Jayasimha, S. (2018). System and method for high throughput fractionated satellites (HTFS) for direct connectivity to and from end user devices and terminals using flight formations of small or very small satellites. US 9973266 B1.

[Egarguin et al., 2020] Egarguin, N. J. A., Jackson, D. R., Onofrei, D., Leclerc, J., and Becker, A. (2020). Adaptive beamforming using scattering from a drone swarm. In *2020 IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)*, pages 1–6.

[Hanna and Cabric, 2022] Hanna, S. and Cabric, D. (2022). Distributed transmit beamforming: Design and demonstration from the lab to UAVs. *IEEE Transactions on Wireless Communications*, 22(2):778–792.

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

[Moghaddam et al., 2019] Moghaddam, M. H., Aghdam, S. R., and Eriksson, T. (2019). Statistical analysis of antenna array systems with perturbations in phase, gain and element positions. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE.

[Mohanti et al., 2019] Mohanti, S., Bocanegra, C., Meyer, J., Secinti, G., Diddi, M., Singh, H., and Chowdhury, K. (2019). Airbeam: Experimental demonstration of distributed beamforming by a swarm of UAVs. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 162–170. IEEE.

[Mohanti et al., 2022] Mohanti, S., Bocanegra, C., Sanchez, S. G., Alemdar, K., and Chowdhury, K. R. (2022). SABRE: swarm-based aerial beamforming radios: Experimentation and emulation. *IEEE Transactions on Wireless Communications*, 21(9):7460–7475.

[Muralidharan and Mostofi, 2016] Muralidharan, A. and Mostofi, Y. (2016). Distributed beamforming using mobile robots. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6385–6389. IEEE.

[Nanzer et al., 2021] Nanzer, J. A., Mghabghab, S. R., Ellison, S. M., and Schlegel, A. (2021). Distributed phased arrays: Challenges and recent advances. *IEEE Transactions on Microwave Theory and Techniques*, 69(11):4893–4907.

[Sun et al., 2021] Sun, G., Li, J., Liu, Y., Liang, S., and Kang, H. (2021). Time and energy minimization communications based on collaborative beamforming for UAV networks: A multi-objective optimization method. *IEEE Journal on Selected Areas in Communications*, 39(11):3555–3572.

[Trucco and Murino, 1999] Trucco, A. and Murino, V. (1999). IEEE stochastic optimization of linear sparse arrays. *IEEE Journal of Oceanic Engineering*, 24(3):291–299.

[Van Trees, 2002] Van Trees, H. L. (2002). *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons.

[Wentz and Chowdhury, 2023] Wentz, M. and Chowdhury, K. R. (2023). Intra-network synchronization and retrodirective distributed transmit beamforming with UAVs. *IEEE Transactions on Vehicular Technology*.

[Xu et al., 2023] Xu, Z., Zheng, X., and Zhou, J. (2023). Optimization design of collaborative beamforming for heterogeneous UAV swarm. *Physical Communication*, 61:102202.