sensors

Article

Fast Low-Sidelobe Pattern Synthesis Using the Symmetry of Array Geometry

Ming Zhang [1,*], Yongxi Liu [1], Haidong Zhou [2] and Anxue Zhang [1]

[1] School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China; liuyongxi@stu.xjtu.edu.cn (Y.L.); anxuezhang@xjtu.edu.cn (A.Z.)
[2] Leihua Avionics Institute of AVIC, Wuxi 214062, China; zhouhd@avic.com
* Correspondence: ming20.zhang@xjtu.edu.cn

Abstract: Array pattern synthesis with low sidelobe levels is widely used in practice. An effective way to incorporate sensor patterns in the design procedure is to use numerical optimization methods. However, the dimension of the optimization variables is very high for large-scale arrays, leading to high computational complexity. Fortunately, sensor arrays used in practice usually have symmetric structures that can be utilized to accelerate the optimization algorithms. This paper studies a fast pattern synthesis method by using the symmetry of array geometry. In this method, the problem of amplitude weighting is formulated as a second-order cone programming (SOCP) problem, in which the dynamic range of the weighting coefficients can also be taken into account. Then, by utilizing the symmetric property of array geometry, the dimension of the optimization problem as well as the number of constraints can be reduced significantly. As a consequence, the computational efficiency is greatly improved. Numerical experiments show that, for a uniform rectangular array (URA) with 1024 sensors, the computational efficiency is improved by a factor of 158, while for a uniform hexagonal array (UHA) with 1261 sensors, the improvement factor is 284.

Keywords: array pattern synthesis; amplitude weighting; symmetry of array geometry; dynamic range ratio; second-order cone programming (SOCP)

check for updates

Citation: Zhang, M.; Liu, Y.; Zhou, H.; Zhang, A. Fast Low-Sidelobe Pattern Synthesis Using the Symmetry of Array Geometry. *Sensors* **2024**, *24*, 4059. https://doi.org/10.3390/s24134059

Academic Editors: Andrzej Stateczny and Christian Vollaire

Received: 27 March 2024
Revised: 10 May 2024
Accepted: 20 June 2024
Published: 21 June 2024

1. Introduction

Sensor arrays play a key role in modern information processing systems due to their powerful spatial filtering ability. They have been successfully applied to various fields such as radar [1], wireless communication [2], satellite navigation [3], radio astronomy [4], speech enhancement [5], and seismic exploration [6]. Array patterns with low sidelobes are very important in these applications to cope with unwanted interferences [7–9]. One of the most widely used methods to synthesize patterns with low sidelobes is amplitude weighting (also called amplitude taper). Conventional analytical methods such as Dolph–Chebychev weighting [10,11] and Taylor weighting [12,13] can be implemented very efficiently. However, these methods are limited to arrays with specific geometries and cannot take into account the sensor patterns. Numerical optimization-based techniques, which address the above issues, have attracted extensive attention in amplitude weighting. Apart from the ability to handle arrays with arbitrary geometries and sensor patterns, they can deal with additional constraints, such as forming nulls at some specified directions or controlling the dynamic range of the weighting coefficients. While heuristic algorithms [14,15] exhibit robust modeling and generalization capabilities, they are relatively time-consuming and cannot guarantee that optimal results will be obtained. Convex optimization shows supreme performance in finding the optimal solutions at the expense of modeling difficulty. Due to the quadratic nature of the array pattern expression, second-order cone programming (SOCP) and semidefinite programming (SDP) have gained widespread applications in array pattern synthesis [16–25]. Nevertheless, for convex optimization methods, the problem models vary with the optimization purposes, and most problems appear to be non-convex. Relaxation must be adopted to transform the original problems into their convex approximations. For example, in [17], the minimization of the l_1-norm, instead of the non-convex l_0-norm, is adopted to design sparse arrays via sequential convex optimizations. In [19], cross-polarization is considered in power pattern synthesis using semidefinite relaxation. In [24], phase-only beamforming with continuous or discrete shifters is also solved by semidefinite relaxation. However, the computational cost of these methods is very high, particularly for large-scale arrays used in modern radar and communication systems [26–28]. Consequently, there is an urgent need to reduce their computational burden.

Although many modeling methods have been proposed for array pattern synthesis using convex optimization, little attention has been paid to their efficient implementations. For example, most optimization problems are solved by available packages such as CVX [29] or SeDumi [30]. In some cases, however, these packages are not effective enough, since redundant variables will be introduced when transforming a given problem into a standard one. In addition, the computational complexities of SOCP and SDP are N^3 and $N^{3.5}$ per iteration, respectively, with N being the dimension of the problem [31,32], while for most problems, 20$\sim$40 iterations are required [33]. When semidefinite relaxation is employed [34], the dimension of the problem is increased from N to N^2, leading to very high computational complexity. As a result, for large-scale arrays with thousands of sensors and tens of thousands of constraints that are commonly encountered in military applications, the SOCP solvers will take several hours to find the solution on a personal computer, while the SDP solvers do not work. Therefore, the computational efficiency is very important for these problems.

Fortunately, sensor arrays as well as their patterns used in practice usually have symmetric structures [35]. For example, there is mirror symmetry in uniform linear arrays (ULAs) and uniform rectangular arrays (URAs), and rotational symmetry in uniform hexagonal arrays (UHAs) and concentric circular arrays (CCAs). Moreover, to simplify the feeding network, large-scale arrays are usually divided into several identical blocks [36]. Thus, the weighting coefficients also have symmetric distributions. Utilizing these symmetries, both the dimension of the optimization variables and the number of constraints in the optimization problems can be reduced. As a consequence, the computational efficiency can be improved dramatically. The conjugate weighting for ULAs with central symmetry is analyzed in [36,37]. However, a detailed discussion on the mirror symmetry and rotational symmetry is not presented. In [38], only the central symmetry is used to optimize a URA. Therefore, the mirror symmetry of the array and its beampattern is not fully exploited. In [39], a thinned array is optimized and the results show that the excitation coefficients exhibit rotational symmetry. Nevertheless, this symmetry is not considered in the design procedure and thereby the computational efficiency can be improved.

This paper studies a fast amplitude weighting method based on the SOCP by leveraging these symmetries. We mainly focus on the planar arrays with mirror symmetry (e.g., URAs) and rotational symmetry (e.g., UHsA). A URA located on the xy-plane is symmetric about the x-axis as well as the y-axis. Therefore, only the weighting coefficients located in the first quadrant need to be optimized. A UHA located on the xy-plane is symmetric about the x-axis as well as the lines passing through the origin with angles of $\phi = 60^\circ$ and $\phi = 120^\circ$. Therefore, only the weighting coefficients located in the region of $0^\circ \leq \phi < 60^\circ$ need to be optimized. Moreover, due to the symmetry of array pattern with respect to the azimuth angle, the number of constraints can also be reduced. Based on these considerations, the dimension of the optimization variables can be reduced by 75% for URAs and by 83% for UHAs. In addition, to control the dynamic range of array excitations that can simplify the feeding network and improve the robustness of the array when performing beam scanning [40–42], the lower and upper bounds of the weighting coefficients are taken into account in the proposed method.

In order to verify the efficiency of the proposed method, we implemented the algorithm described in [31,43] for SOCP problems without performing any code optimization. Two URAs with 256 sensors and 1024 sensors and two UHAs with 331 sensors and 1261 sensors are tested in the simulation experiments. The results show that the computational efficiency is improved by an order of magnitude for URA-256 and UHA-331, and by two orders of magnitude for URA-1024 and UHA-1261. Moreover, it is found that the direct implementation of the SOCP solver described in [31,43] is more efficient than the CVX package when solving the problems of amplitude tapering.

The main contributions of this paper are as follows:

1. We propose a fast low-sidelobe pattern synthesis method using the symmetry of array geometry, which reduces the dimension of the optimization variables and improves the computational efficiency significantly.
2. The lower and upper bounds of the weighting coefficients can also be specified in the proposed method, which can control the dynamic range of the weighting coefficients.

The remainder of this paper is organized as follows. Section 2 formulates the amplitude weighting with dynamic range constraints as an SOCP problem. Section 3 describes the method that utilizes the symmetry of URAs and UHAs to reduce the dimension of the optimization problem obtained in Section 2. Simulation experiments are given in Section 4 to validate the effectiveness of the proposed method, followed by a conclusion and discussion in Section 5.

Notations: We use lowercase letters (e.g., a), lowercase boldface letters (e.g., $\mathbf{a}$), and uppercase boldface letters (e.g., $\mathbf{A}$) to represent scalars, vectors, and matrices, respectively. The superscripts $()^{\mathrm{T}}$ and $()^{\mathrm{H}}$ denote the transpose and Hermitian transpose, respectively. The real part and imaginary part of a complex number z are denoted by $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$, while æ $= \sqrt{-1}$ is the imaginary unit.

2. Problem Formulation

Consider a planar array located on the xy-plane as shown in Figure 1. The problem of amplitude weighting that minimizes the peak sidelobe level (PSLL) of the array pattern can be formulated as

$$\min_{\mathbf{w}\in\mathbb{R}^M} \quad \max_{(\theta,\phi)\in\Theta_{\mathrm{sl}}} \left|\mathbf{w}^{\mathrm{H}}\mathbf{v}(\theta,\phi)\right|, \tag{1a}$$

$$\text{s.t.} \quad \left|\mathbf{w}^{\mathrm{H}}\mathbf{v}(\theta_0,\phi_0)\right| = 1, \tag{1b}$$

where $\mathbf{w}$ is the weighting coefficients, M is the number of sensors, Θ_{sl} is the region of sidelobes, $\mathbf{v}(\theta,\phi)$ is the steering vector in direction (θ,ϕ), with (θ_0,ϕ_0) being the direction of the main beam. For planar arrays whose sensors lie in the xy-plane, the mth entry of $\mathbf{v}(\theta,\phi)$ is given by [37] (ch. 4)

$$f_m(\theta,\phi)e^{æ\frac{2\pi}{\lambda}[x_m\sin(\theta)\cos(\phi)+y_m\sin(\theta)\sin(\phi)]}, \tag{2}$$

where $f_m(\theta,\phi)$ and (x_m,y_m) are the pattern and location of the mth sensor, respectively, and λ is the wavelength of the operating frequency.

Because there are infinitely many directions in Θ_{sl}, one has to approximate Θ_{sl} by finite samples. For example, by sampling Θ_{sl} with N points $\{(\theta_n,\phi_n)\}_{n=1}^{N}$, Problem (1) can be approximated by [44]

$$\min_{g\in\mathbb{R}^+,\mathbf{w}\in\mathbb{R}^M} \quad g, \tag{3a}$$

$$\text{s.t.} \quad \mathrm{Re}\left(\mathbf{v}_0^{\mathrm{H}}\mathbf{w}\right) = 1, \tag{3b}$$

$$\left|\mathbf{v}_n^{\mathrm{H}}\mathbf{w}\right| \le g, \quad n = 1,\cdots,N, \tag{3c}$$

where $\mathbf{v}_n = \mathbf{v}(\theta_n, \phi_n)$. Note that the constraint $|\mathbf{v}_0^{\mathrm{H}}\mathbf{w}| = 1$ has been replaced by $\mathrm{Re}(\mathbf{v}_0^{\mathrm{H}}\mathbf{w}) = 1$, which does not change the optimal solution [42].

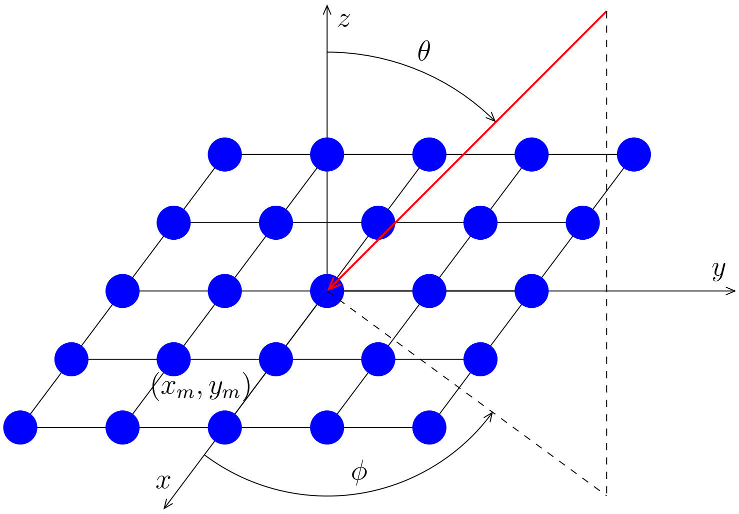


Figure 1. A planar array located on the *xy*-plane.

In addition, to prevent large weighting coefficients that make the synthesized pattern sensitive to array errors [37] (ch. 2), one can impose the range constraints as follows:

$$\min_{g\in\mathbb{R}^+,\mathbf{w}\in\mathbb{R}^M} \quad g, \tag{4a}$$

$$\text{s.t.} \quad \mathrm{Re}\left(\mathbf{v}_0^{\mathrm{H}}\mathbf{w}\right) = 1, \tag{4b}$$

$$\left|\mathbf{v}_n^{\mathrm{H}}\mathbf{w}\right| \le g, \quad n = 1, \cdots, N, \tag{4c}$$

$$\alpha_m \le w_m \le \beta_m, \quad m = 1, \cdots, M. \tag{4d}$$

There are two advantages by imposing the range constraints: reducing the complexity of feeding network and increasing the robustness of beam scanning (e.g., maintaining a low PSLL). The cost is adding more constraints in the optimization problem and thus increasing the computational complexity. Because M and N are very large for planar arrays (e.g., M is on the order of $10^2 \sim 10^3$ and N is on the order of 10^4), solving Problem (4) efficiently is very important.

Since the steering vectors $\mathbf{v}_n$ are complex, we have to convert them into real ones to represent Problem (4) in real variables. For example, let

$$\hat{\mathbf{v}}_n = \begin{bmatrix}\mathrm{Re}(\mathbf{v}_n)\\ \mathrm{Im}(\mathbf{v}_n)\end{bmatrix}, \ \tilde{\mathbf{v}}_n = \begin{bmatrix}\mathrm{Im}(\mathbf{v}_n)\\ -\mathrm{Re}(\mathbf{v}_n)\end{bmatrix}, \ \hat{\mathbf{w}} = \begin{bmatrix}\mathrm{Re}(\mathbf{w})\\ \mathrm{Im}(\mathbf{w})\end{bmatrix}. \tag{5}$$

Then we have [45]

$$\mathrm{Re}\left(\mathbf{v}_n^{\mathrm{H}}\mathbf{w}\right) = \hat{\mathbf{v}}_n^{\mathrm{T}}\hat{\mathbf{w}} \quad \text{and} \quad \left|\mathbf{v}_n^{\mathrm{H}}\mathbf{w}\right| = \left\|[\hat{\mathbf{v}}_n, \ \tilde{\mathbf{v}}_n]^{\mathrm{T}}\hat{\mathbf{w}}\right\|. \tag{6}$$

Because $\mathrm{Im}(\mathbf{w}) = \mathbf{0}$, Problem (4) is equivalent to

$$\min_{g\in\mathbb{R}^+,\mathbf{w}\in\mathbb{R}^M} \quad g, \tag{7a}$$

$$\text{s.t.} \quad [\mathrm{Re}(\mathbf{v}_0)]^{\mathrm{T}}\mathbf{w} = 1, \tag{7b}$$

$$\left\|[\mathrm{Re}(\mathbf{v}_n), \ \mathrm{Im}(\mathbf{v}_n)]^{\mathrm{T}}\mathbf{w}\right\| \le g, \tag{7c}$$

$$\alpha_m \le w_m \le \beta_m, \tag{7d}$$

where $n = 1, \cdots, N$ and $m = 1, \cdots, M$. In what follows, we use Problem (7) to illustrate how to utilize the symmetry of array geometry. To help understand the problem model, the meanings of variables and parameters in Problem (7) are summarized in Table 1.

Table 1. Variable definitions for Problem (7).

Variable/Parameter	Meaning
g	the maximum array gain in sidelobe region
$\mathbf{v}_0$	steering vector of the scan angle
$\mathbf{w}$	weighting coefficients (optimization variables)
$\mathbf{v}_n$	steering vectors in sidelobe region
α_m	lower bounds of the weighting coefficients
β_m	upper bounds of the weighting coefficients

Problem (7) can be translated into an SOCP problem and solved by the primal–dual interior-point methods [31]. Because the SOCP problem is convex, its optimal solution is guaranteed. The dimensions of the primal and dual variables of (7) are $D_{\mathrm{p}} = 2M + 3N + 1$ and $D_{\mathrm{d}} = M + 1$, respectively. Because the main computations of the SOCP solver come from (i) constructing a $D_{\mathrm{d}} \times D_{\mathrm{d}}$ matrix from a $D_{\mathrm{d}} \times D_{\mathrm{p}}$ matrix and (ii) solving two linear systems of equations with dimension D_{d}, the computational complexity of (7) can be reduced if we can reduce D_{p} and D_{d}.

3. The Proposed Method

3.1. Fast Amplitude Weighting for URA Using Mirror Symmetry

Without loss of generality, consider a URA consisting of $M = M_x \times M_y$ sensors with M_x and M_y being even. The method developed in this section can be easily extended to the cases when M_x and M_y have different parities. The geometry of the URA is shown in Figure 2, where the same icon means that those sensors have the same pattern and the same weighting coefficient.

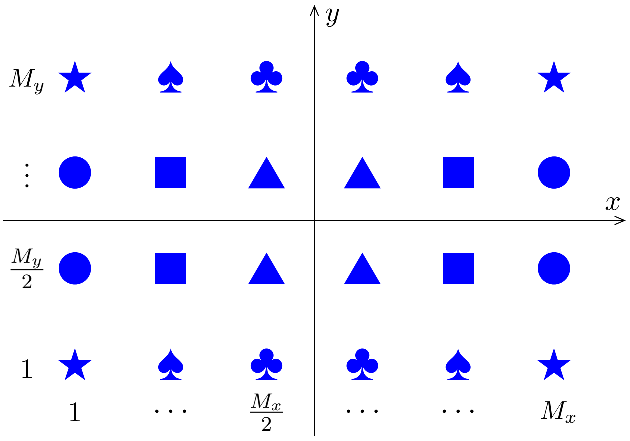


Figure 2. The symmetry of a URA with M_x and M_y being even.

The sensors of the URA are numbered in a consecutive way, starting with the bottom-left ($m = 1$) and proceeding in the x-direction and then in the y-direction. Let $\mathcal{I}_{m_y}^k$ be the indices of the sensors in the m_yth row in the $[k + 2\mathrm{sign}(2.5 - k)]$th quadrant, i.e.,

$$\mathcal{I}_{m_y}^1 = (m_y - 1)M_x + \left\{1, \cdots, \frac{M_x}{2}\right\}, \tag{8a}$$

$$\mathcal{I}_{m_y}^2 = m_y M_x - \left\{0, \cdots, \frac{M_x}{2} - 1\right\}, \tag{8b}$$

$$\mathcal{I}_{m_y}^3 = (M_y - m_y)M_x + \left\{1, \cdots, \frac{M_x}{2}\right\}, \tag{8c}$$

$$\mathcal{I}_{m_y}^4 = (M_y - m_y + 1)M_x - \left\{0, \cdots, \frac{M_x}{2} - 1\right\}, \tag{8d}$$

where $m_y = 1, \cdots, M_y/2$. For example, if $M_x = 6$ and $M_y = 4$, then

$$\mathcal{I}_1^1 = \{1, 2, 3\}, \qquad \mathcal{I}_2^1 = \{7, 8, 9\}, \tag{9a}$$

$$\mathcal{I}_1^2 = \{6, 5, 4\}, \qquad \mathcal{I}_2^2 = \{12, 11, 10\}, \tag{9b}$$

$$\mathcal{I}_1^3 = \{19, 20, 21\}, \qquad \mathcal{I}_2^3 = \{13, 14, 15\}, \tag{9c}$$

$$\mathcal{I}_1^4 = \{24, 23, 22\}, \qquad \mathcal{I}_2^4 = \{18, 17, 16\}. \tag{9d}$$

Furthermore, let

$$\mathcal{I}^k = \mathcal{I}_1^k \cup \mathcal{I}_2^k \cdots \cup \mathcal{I}_{M_y/2}^k, \quad k = 1, 2, 3, 4, \tag{10}$$

i.e., each $\mathcal{I}^k$ contains the indices of sensors in one quadrant. By symmetry, we have

$$\mathbf{w}(\mathcal{I}^1) = \mathbf{w}(\mathcal{I}^2) = \mathbf{w}(\mathcal{I}^3) = \mathbf{w}(\mathcal{I}^4). \tag{11}$$

Consequently,

$$\mathbf{v}_n^{\mathrm{H}}\mathbf{w} = \sum_{k=1}^{4} \left[\mathbf{v}_n(\mathcal{I}^k)\right]^{\mathrm{H}} \mathbf{w}(\mathcal{I}^k) = \mathbf{u}_n^{\mathrm{H}}\mathbf{w}(\mathcal{I}^1), \tag{12}$$

where

$$\mathbf{u}_n = \mathbf{v}_n\left(\mathcal{I}^1\right) + \mathbf{v}_n\left(\mathcal{I}^2\right) + \mathbf{v}_n\left(\mathcal{I}^3\right) + \mathbf{v}_n\left(\mathcal{I}^4\right). \tag{13}$$

Equation (12) indicates that the dimension of the optimization variable (i.e., g and $\mathbf{w}$) in Problem (7) can be reduced from $M + 1$ to $M/4 + 1$.

In addition, due to the symmetry of array geometry, the array pattern also has symmetry in the ϕ-direction. For example, when specifying the sidelobe region, we only need to impose the constraints for $\phi \in [0, 90^\circ]$ rather than $\phi \in [0, 360^\circ]$. Strictly speaking, this simplification requires that the sensor patterns are also symmetric about the x and y axes, which is satisfied in practice because the sensor patterns in a large array usually depend weakly on ϕ. For example, the microstrip patch antenna and pyramidal/conical horn antenna are widely used in antenna arrays [46] (ch. 13), whose patterns are commonly modeled as functions that do not depend on ϕ [47] (ch. 6).

Now we can reformulate (7) as an SOCP problem whose primal and dual dimensions are $D_{\mathrm{p}} = M/2 + 3N/4 + 1$ and $D_{\mathrm{d}} = M/4 + 1$, respectively. Let

$$\mathbf{y} = \begin{bmatrix} g \\ \mathbf{w}(\mathcal{I}^1) \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}, \tag{14a}$$

$$\mathbf{a}_0 = \begin{bmatrix} 0 \\ \mathrm{Re}(\mathbf{u}_0) \end{bmatrix}, \qquad \mathbf{a}_n = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}, \tag{14b}$$

$$\mathbf{A}_n = \begin{bmatrix} 0 & 0 \\ \mathrm{Re}(\mathbf{u}_n) & \mathrm{Im}(\mathbf{u}_n) \end{bmatrix}, \qquad \mathbf{d}_m = \begin{bmatrix} 0 \\ \mathbf{e}_m \end{bmatrix} \tag{14c}$$

for $n = 1, \cdots, N/4$ and $m = 1, \cdots, M/4$, where $\mathbf{e}_m$ is the mth column of the identity matrix of order $M/4$. Then Problem (7) can be reformulated as

$$\min_{\mathbf{y} \in \mathbb{R}^{(M/4)+1}} \quad \mathbf{b}^{\mathrm{T}}\mathbf{y}, \tag{15a}$$

$$0 \leq \mathbf{a}_0^{\mathrm{T}}\mathbf{y} - 1, \tag{15b}$$

$$0 \leq \mathbf{d}_m^{\mathrm{T}}\mathbf{y} - \alpha_m, \tag{15c}$$

$$0 \leq -\mathbf{d}_m^{\mathrm{T}}\mathbf{y} + \beta_m, \tag{15d}$$

$$\left\| \mathbf{A}_n^{\mathrm{T}}\mathbf{y} \right\| \leq \mathbf{a}_n^{\mathrm{T}}\mathbf{y}, \tag{15e}$$

which is a standard (dual) SOCP problem [43]. Note that the constraint $\mathbf{a}_0^{\mathrm{T}}\mathbf{y} = 1$ has been replaced by $0 \leq \mathbf{a}_0^{\mathrm{T}}\mathbf{y} - 1$ in order to express (15) in the standard form. This replacement does not change the solution of (15) because $\mathbf{w}(\mathcal{I}^1)$ should become as small as possible to minimize the PSLL. Therefore, the inequality in (15b) will become an equality at the optimal point.

3.2. Fast Amplitude Weighting for UHA Using Rotational Symmetry

The geometry of the UHA is shown in Figure 3, where the same icon means that those sensors have the same pattern and the same weighting coefficient. It can be seen that the UHA is rotationally symmetric about the origin.

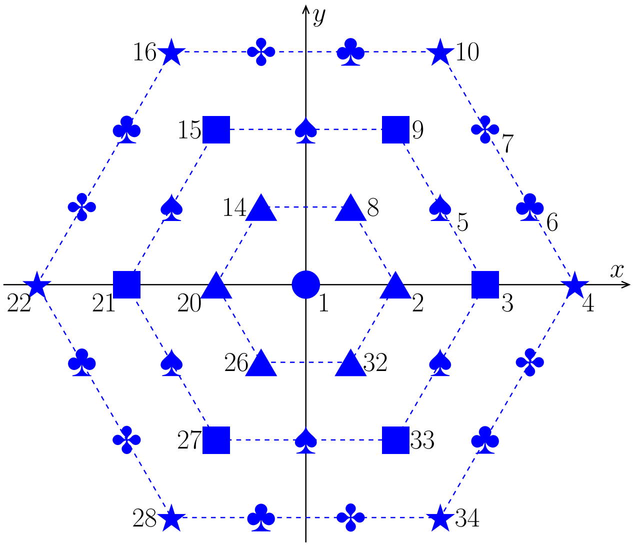


Figure 3. The symmetry of a UHA with 37 sensors.

Suppose the UHA consists of C hexagons (dashed lines in Figure 3); then, there are

$$M = 1 + 6 + 12 + \cdots + 6C = 3C(C+1) + 1 \tag{16}$$

sensors in the array. In the region of $0^\circ \leq \phi < 60^\circ$, there are (including the origin)

$$\frac{M-1}{6} + 1 = \frac{C(C+1)}{2} + 1 = H + 1 \tag{17}$$

sensors, where

$$H = \frac{M-1}{6} \quad \text{and} \quad M = 6H + 1\,. \tag{18}$$

For example, in Figure 3, $C = 3$, $M = 37$, and $H = 6$.

The numbering system of the UHA is as follows:

- The $H+1$ sensors in the region of $0^\circ \leq \phi < 60^\circ$ are numbered in a consecutive way, starting with the origin ($m = 1$) and proceeding in the x-direction and then in the y-direction ($m = 2, \cdots, H+1$).
- If the position of a sensor is obtained by rotating the mth sensor ($m = 2, \cdots, H+1$) with an angle of $k \times 60^\circ$ ($k = 1, \cdots, 5$), then this sensor is numbered by $m = 2 + kH$.

By using symmetry, we only have to optimize the weighting coefficients in the region of $0^\circ \leq \phi < 60^\circ$. Let

$$\mathcal{I}^k = \{kH+2, kH+3, \cdots, kH+(H+1)\} \tag{19}$$

for $k = 0, 1, \cdots, 5$, then

$$\mathbf{w}(\mathcal{I}^0) = \mathbf{w}(\mathcal{I}^1) = \cdots = \mathbf{w}(\mathcal{I}^5). \tag{20}$$

Consequently,

$$\begin{aligned} \mathbf{v}_n^{\mathrm{H}}\mathbf{w} &= v_{n1}^* w_1 + \sum_{k=0}^{5} \left[\mathbf{v}_n(\mathcal{I}^k)\right]^{\mathrm{H}} \mathbf{w}(\mathcal{I}^k) \\ &= v_{n1}^* w_1 + \mathbf{u}_n^{\mathrm{H}} \mathbf{w}(\mathcal{I}^1), \end{aligned} \tag{21}$$

where

$$\mathbf{u}_n = \mathbf{v}_n\left(\mathcal{I}^0\right) + \mathbf{v}_n\left(\mathcal{I}^1\right) + \cdots + \mathbf{v}_n\left(\mathcal{I}^5\right). \tag{22}$$

Equation (21) indicates that the dimension of $\mathbf{w}$ in Problem (7) can be reduced from $M+1$ to $(M-1)/6+2$.

Similar to the case of URA, we only need to impose the constraints for $\phi \in [0, 90^\circ]$ rather than $\phi \in [0, 360^\circ]$. And (7) can be reformulated as (15) with

$$\mathbf{y} = \begin{bmatrix} g \\ w_1 \\ \mathbf{w}(\mathcal{I}^1) \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}, \tag{23a}$$

$$\mathbf{a}_0 = \begin{bmatrix} 0 \\ \mathrm{Re}(v_{01}) \\ \mathrm{Re}(\mathbf{u}_0) \end{bmatrix}, \qquad \mathbf{a}_n = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}, \tag{23b}$$

$$\mathbf{A}_n = \begin{bmatrix} 0 & 0 \\ \mathrm{Re}(v_{n1}) & \mathrm{Im}(v_{n1}) \\ \mathrm{Re}(\mathbf{u}_n) & \mathrm{Im}(\mathbf{u}_n) \end{bmatrix}, \qquad \mathbf{d}_m = \begin{bmatrix} 0 \\ \mathbf{e}_m \end{bmatrix} \tag{23c}$$

for $n = 1, \cdots, N/4$ and $m = 1, \cdots, H/6+1$, where $\mathbf{e}_m$ is the mth column of the identity matrix of order $H+1$. The primal and dual dimensions of this simplified problem are $D_{\mathrm{p}} = 2H + 3N/4 + 2$ and $D_{\mathrm{d}} = H+1$, respectively.

4. Numerical Results and Analyses

4.1. Experiment 1: URA with Mirror Symmetry

A URA consisting of 16 × 16 isotropic sensors is used in the first experiment. Because the array pattern is symmetric about the xy-plane, only the upper half-space is shown when plotting the array pattern. Problems (7) and (15) can be solved by the CVX package. However, to provide a user-friendly interface, CVX may introduce auxiliary variables when transforming a given problem to a standard one. These redundant variables may reduce the computational efficiency of the solving processes. Therefore, we use the algorithm described in [31,43] to solve Problems (7) and (15), whose solutions are denoted by $\mathbf{w}_{(7)}$ and $\mathbf{w}_{(15)}$, respectively.

The region of sidelobes is $[10, 90]^\circ \times [0, 360]^\circ$, which is sampled by $\delta_\theta = 2^\circ$ and $\delta_\phi = 4^\circ$. To prevent negative weighting coefficients, α_m are usually set to 0. Because $w_m = 1/M$ for uniform weighting, β_m are usually set to μ/M, where μ is near 2. In this example, $\beta_m = 2.1/M$. The pattern synthesized by (15) is shown in Figure 4, which is the same as that synthesized by (7)). The weighting errors between $\mathbf{w}_{(15)}$ and $\mathbf{w}_{(7)}$ are shown in Figure 5. It can be seen that the difference between $\mathbf{w}_{(15)}$ and $\mathbf{w}_{(7)}$ is negligible. The relative error is

$$\frac{\|\mathbf{w}_{(15)} - \mathbf{w}_{(7)}\|}{\|\mathbf{w}_{(7)}\|} = 0.07\%, \tag{24}$$

which means that Problem (7) also gives symmetric weighting coefficients.

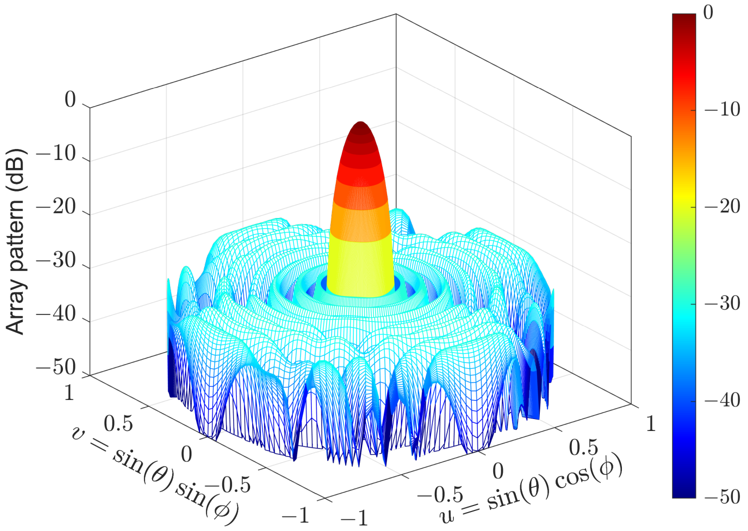


Figure 4. Array pattern of the 16 × 16 URA synthesized by (15).

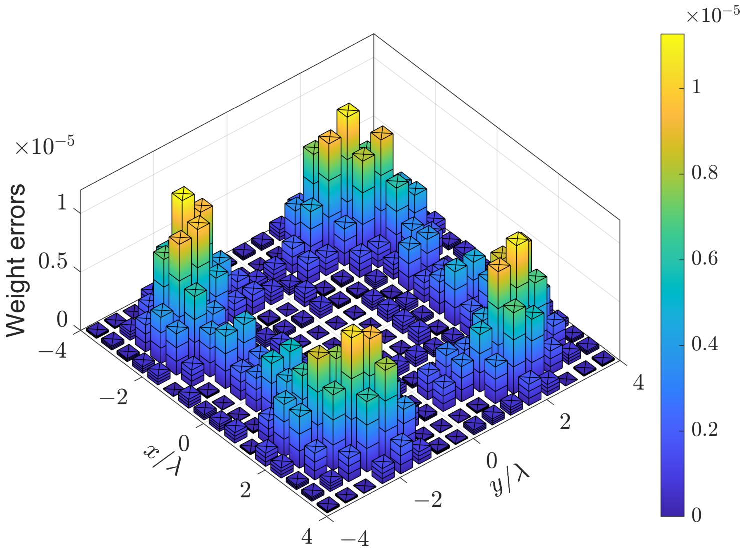


Figure 5. Weighting errors of the 16 × 16 URA synthesized by (15) and (7).

The CPU time T_{cpu} of solving (7) and (15) on a computer with four 2.4 GHz CPUs as well as the PSLLs of the synthesized patterns are listed in Table 2. It can be seen that (7) and (15) produce the same PSLL. However, solving (15) is 12 times faster than solving (7). The results when (7) and (15) are solved by CVX [29] are also provided, which indicates that solving (15) is 28 times faster than solving (7). In addition, the CVX package spends more CPU time than the solver based on [31,43]. The dimensions of the primal and dual variables involved in (7) and (15) are also listed in Table 2, showing that the dimension of the optimization problem is reduced by 75% using the mirror symmetry of the URA.

Another URA consisting of 32 × 32 isotropic sensors is used in the second experiment. The region of sidelobes is $[5, 90]^\circ \times [0, 360]^\circ$, which is sampled by $\delta_\theta = 1^\circ$ and $\delta_\phi = 2^\circ$. The parameters α_m and β_m are set to 0 and $1.9/M$, respectively. The pattern synthesized by (15) is shown in Figure 6, while the CPU time and PSLL are listed in Table 2. It can be seen that (15) produces the same PSLL as that of (7). However, solving (15) is 158 times faster than solving (7). On the other hand, solving (15) is 83 times faster than solving (7) by using the CVX [29] toolbox.

Table 2. The PSLL and CPU time for URA.

Problem	Solver	$M_x \times M_y$	D_p	D_d	PSLL (dB)	T_cpu (s)
(7)	[31] + [43]	16×16	11,706	257	−29.6	15.3
(7)	CVX [29]	16×16	/	/	−29.6	61.5
(15)	[31] + [43]	16×16	2958	65	−29.6	1.3
(15)	CVX [29]	16×16	/	/	−29.6	2.2
(7)	[31] + [43]	32×32	48,747	1025	−30.3	2884.6
(7)	CVX [29]	32×32	/	/	−30.3	4088.1
(15)	[31] + [43]	32×32	12,381	257	−30.3	18.3
(15)	CVX [29]	32×32	/	/	−30.3	49.1

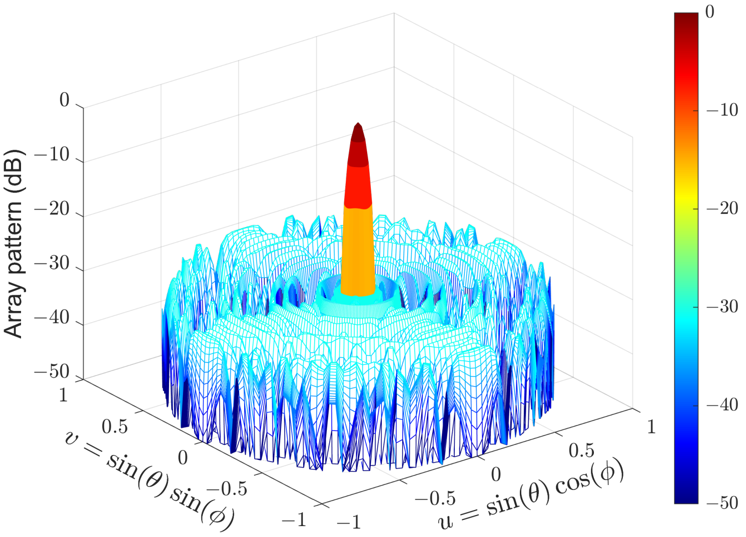
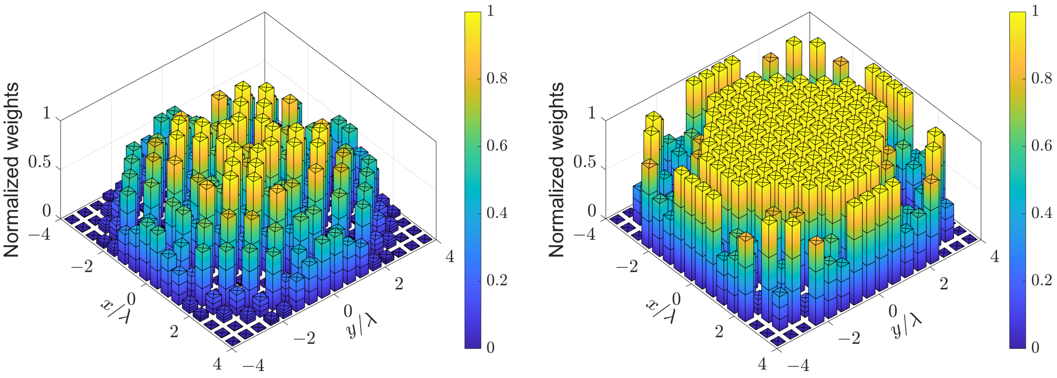


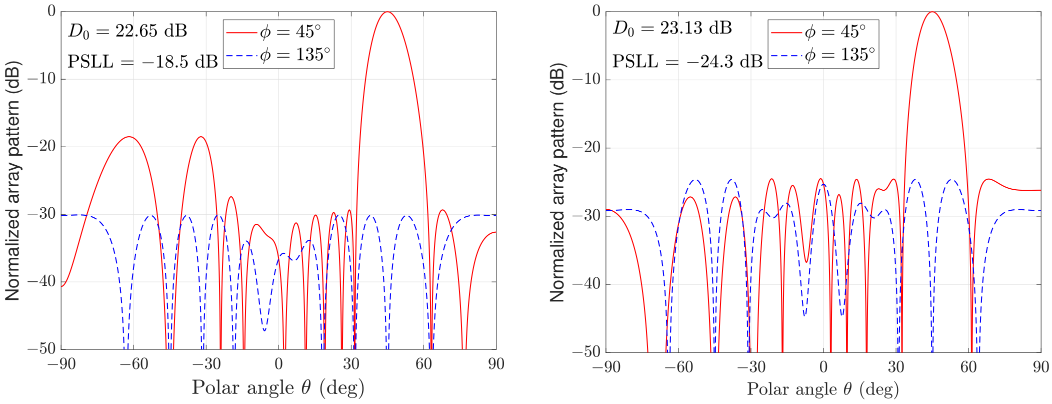
Figure 6. Array pattern of the 32 × 32 URA synthesized by (15).

The effect of dynamic range constraints $\alpha_m \leq w_m \leq \beta_m$ is shown in Figure 7, where the synthesized patterns of the 16 × 16 URA with and without the dynamic range constraints are compared. The lower and upper bounds of the weighting coefficients are $\alpha_m = 0$ and $\beta_m = 1.5/M$ respectively. It can be seen that the weighting coefficients have a smaller fluctuation when the dynamic range constraints are imposed, which can enhance the robustness of array patterns [48]. For example, when the array is steered to $(45°, 45°)$, the pattern with dynamic range constraints has a higher directivity (improved by 0.5 dB) and a lower PSLL (improved by 5.8 dB) than the pattern without dynamic range constraints.



(**a**) Weights without range constraints. (**b**) Weights with range constraints.

Figure 7. *Cont.*



(**c**) Pattern cuts without range constraints. (**d**) Pattern cuts with range constraints.

Figure 7. The effect of dynamic range constraints when the array is steered to (45°, 45°).

4.2. Experiment 2: UHA with Rotational Symmetry

A UHA consisting of 10 hexagons (331 sensors) is used in the third experiment. All sensors have the same pattern, given by

$$f_m(\theta, \phi) = \left[\frac{1 + \cos(\theta)}{2}\right]^2 = \cos^4(\theta/2)\,, \tag{25}$$

which is shown in Figure 8. Because this pattern is not symmetric about the *xy*-plane, the full space is shown when plotting the array patterns.

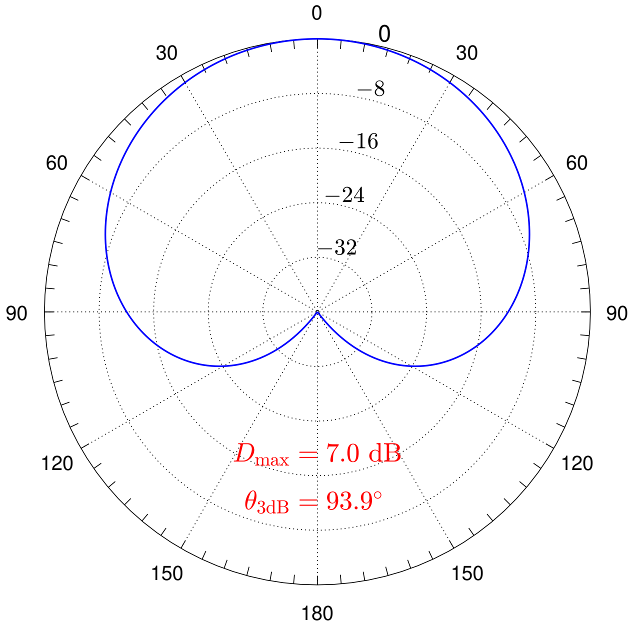
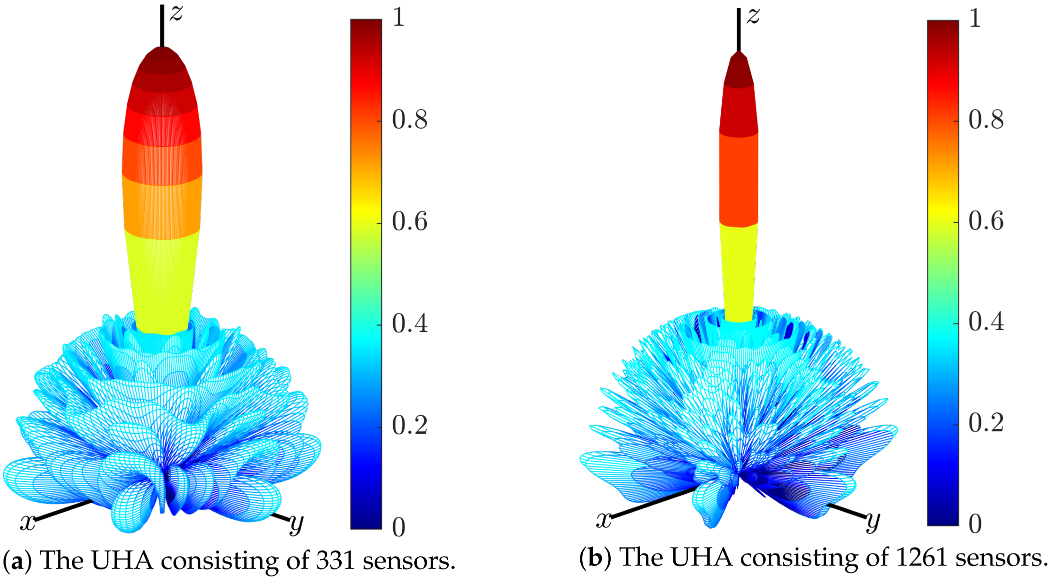


Figure 8. The pattern of $\cos^4(\theta/2)$ with directivity $D_{\max} = 7.0$ dB and 3 dB beamwidth $\theta_{3\mathrm{dB}} = 93.9°$.

The region of sidelobes is $[9, 180]° \times [0, 360]°$, which is sampled by $\delta_\theta = 2°$ and $\delta_\phi = 4°$. The parameters α_m and β_m are set to 0 and $1.8/M$, respectively. The pattern synthesized by (15) is shown in Figure 9a, which is the same as that synthesized by (7)). The weighting errors between $\mathbf{w}_{(15)}$ and $\mathbf{w}_{(7)}$ are shown in Figure 10. It can be seen that the difference between $\mathbf{w}_{(15)}$ and $\mathbf{w}_{(7)}$ is negligible. The relative error is

$$\frac{\|\mathbf{w}_{(15)} - \mathbf{w}_{(7)}\|}{\|\mathbf{w}_{(7)}\|} = 0.02\%\,, \tag{26}$$

which means that Problem (7) also gives symmetric weighting coefficients. The CPU time T_{cpu} of solving (7) and (15) as well as the PSLL are listed in Table 3. It can be seen that (7) and (15) produce the same PSLL. However, solving (15) is 19 times faster than solving (7). The results when (7) and (15) are solved by CVX [29] are also provided, which indicates that solving (15) is 49 times faster than solving (7). The dimensions of the primal and dual variables involved in (7) and (15) are also listed in Table 3. The results indicate that the number of constraints rather than the number of sensors dominates the dimension of the optimization problem.



(**a**) The UHA consisting of 331 sensors. (**b**) The UHA consisting of 1261 sensors.

Figure 9. Array patterns of the UHA synthesized by (15).

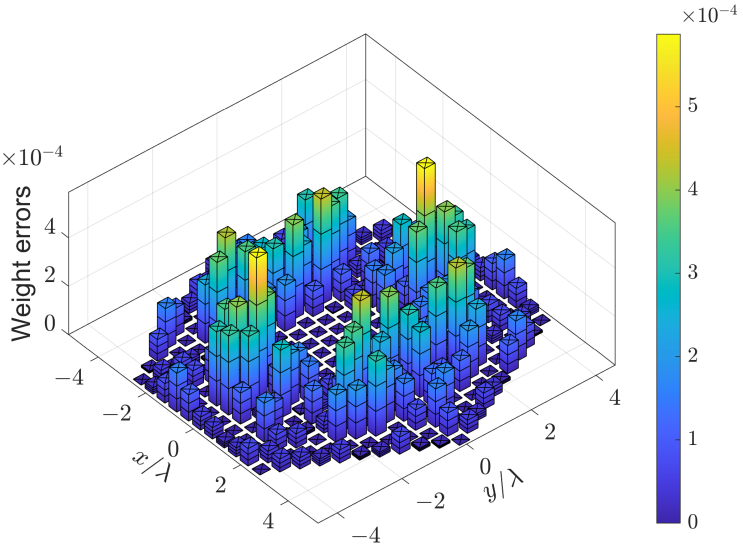


Figure 10. Weighting errors of the UHA with 10 hexagons synthesized by (15) and (7).

Another UHA consisting of 20 hexagons (1261 sensors) is used in the fourth experiment. The region of sidelobes is $[5, 90]^\circ \times [0, 360]^\circ$, which is sampled by $\delta_\theta = 1^\circ$ and $\delta_\phi = 2^\circ$. The parameters α_m and β_m are set to 0 and $1.7/M$, respectively. The pattern synthesized by (15) is shown in Figure 9b, while the CPU time and PSLL are listed in Table 3. It can be seen that (15) produces the same PSLL as that of (7). However, solving (15) is 284 times faster than solving (7). On the other hand, solving (15) is 133 times faster than solving (7) by using the CVX [29] toolbox.

Table 3. The PSLL and CPU time for UHA.

Problem	Solver	M	D_p	D_d	PSLL (dB)	T_cpu (s)
(7)	[31] + [43]	331	11,856	332	−30.9	20.7
(7)	CVX [29]	331	/	/	−30.9	92.3
(15)	[31] + [43]	331	2942	57	−30.9	1.1
(15)	CVX [29]	331	/	/	−30.9	1.9
(7)	[31] + [43]	1261	49,221	1262	−31.2	3859.0
(7)	CVX [29]	1261	/	/	−31.2	5480.6
(15)	[31] + [43]	1261	12,291	212	−31.2	12.8
(15)	CVX [29]	1261	/	/	−31.2	41.1

5. Conclusions and Discussion

A fast low-sidelobe pattern synthesis method was studied in this paper, which utilizes the symmetry of array geometry to reduce the dimension of the optimization variables as well as the number of constraints. Different from the analytical approaches, the proposed method was modeled by an SOCP problem which can take into account the sensor pattern and the dynamic range of the weighting coefficients. Simulation experiments showed that, for a URA with 1024 sensors, the computational efficiency was improved by a factor of 158, while for a UHA with 1261 sensors, the improvement factor was 284. In addition, the proposed method can be easily extended to other symmetric arrays.

The improvement of computational efficiency in the second experiment is larger than that in the first experiment. This is because the primal and dual dimensions of (7) in the first experiment are $D_\mathrm{p} = 11{,}706$ and $D_\mathrm{d} = 257$, while in the second experiment, they are $D_\mathrm{p} = 48{,}747$ and $D_\mathrm{d} = 1025$. Therefore, constructing the $D_\mathrm{p} \times D_\mathrm{p}$ matrix in the interior-point method dominates the computations in the first experiment, while solving the $D_\mathrm{p} \times D_\mathrm{p}$ linear systems of equations dominates the computations in the second experiment. For the same reason, the improvement of computational efficiency in the fourth experiment is larger than that in the third experiment.

It is worth remarking that there may be different types of symmetry in one array (e.g., the mirror symmetry and rotational symmetry in a square array). In such cases, either symmetry can be exploited depending on the array pattern to be synthesized. Finally, Tables 2 and 3 show that the direct implementation of the SOCP solver described in [31,43] is faster than the CVX toolbox [29]. By checking the interior parameters, it is found that the dimension of the optimization problem in CVX is larger than that of the standard form. Therefore, redundant variables are introduced in CVX when solving a convex problem that is not in its standard form. Another possible reason is that before solving the given problem, pre-examination is performed by the CVX toolbox to select a proper algorithm, which takes extra time.

Applications of array symmetries discussed in this paper are not limited to amplitude weighting; they can also be utilized in other array pattern synthesis problems such as sparse array optimization. Apart from convex optimization, other numerical optimization-based methods can also leverage this geometry property. By exploiting array symmetries, advantages arise not only on the improvement of computational efficiency; from an engineering point of view, symmetry also eases the difficulty in manufacture of the sensor arrays. With the development of extremely large-scale sensor arrays, we believe that the applications of array symmetries will attract more and more attention. In addition, future works could focus on their extension to semi-symmetric arrays, where partial symmetry exists in the array geometry.

Author Contributions: Conceptualization, M.Z. and A.Z.; methodology, M.Z. and Y.L.; software and validation, M.Z. and H.Z.; writing–original draft preparation, M.Z. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Shaanxi S&T Innovation Team under Grant No. 2023-CX-TD-03 and in part by the Shaanxi Key Laboratory of Deep Space Exploration Intelligent Information Technology under Grant No. 2021SYS-04.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Richards, M.A. *Fundamentals of Radar Signal Processing*, 3rd ed.; McGraw-Hill Education: Singapore, 2022.
2. Heath, R.W.; Lozano, A. *Foundations of MIMO Communication*; Cambridge University Press: Cambridge, UK, 2019.
3. Kaplan, E.D.; Hegarty, C.J. *Understanding GPS/GNSS: Principles and Applications*, 3rd ed.; Artech House: London, UK, 2019.
4. Warnick, K.F.; Maaskant, R.; Ivashina, M.V.; Davidson, D.B.; Jeffs, B.D. *Phased Arrays for Radio Astronomy, Remote Sensing, and Satellite Communications*; Cambridge University Press: Cambridge, UK, 2018.
5. Rafaely, B. *Fundamentals of Spherical Array Processing*, 2nd ed.; Springer: Cham, Switzerland, 2019.
6. Aster, R.; Borchers, B.; Thurber, C. *Parameter Estimation and Inverse Problems*, 3rd ed.; Elsevier: Cambridge, MA, USA, 2019.
7. Aslan, Y.; Puskely, J.; Roederer, A.; Yarovoy, A. Phase-Only Control of Peak Sidelobe Level and Pattern Nulls Using Iterative Phase Perturbations. *IEEE Antennas Wirel. Propag. Lett.* **2019**, *18*, 2081–2085. [CrossRef]
8. Cai, L.; Chu, R.; Ding, Z.; Zou, Y.; Li, H. Robust low-sidelobe transmit beamforming under peak-to-average-power ratio constraint. *Sensors* **2023**, *23*, 4468. [CrossRef]
9. Li, M.; Chen, S.; Liu, Y.; Guo, Y.J. Wide-angle beam scanning phased array antennas: A review. *IEEE Open J. Antennas Propag.* **2023**, *4*, 695–712. [CrossRef]
10. Dolph, C.L. A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level. *Proc. IRE* **1946**, *34*, 335–348. [CrossRef]
11. Riblet, H.J. Discussion on 'a current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level'. *Proc. IRE* **1947**, *35*, 489–492.
12. Taylor, T.T. Design of line-source antennas for narrow beamwidth and low sidelobes. *IRE Trans. Antennas Propag.* **1955**, *3*, 16–28. [CrossRef]
13. Villeneuve, A.T. Taylor patterns for discrete arrays. *IEEE Trans. Antennas Propag.* **1984**, *32*, 1089–1093. [CrossRef]
14. Trucco, A. Weighting and thinning wide-band arrays by simulated annealing. *Ultrasonics* **2002**, *40*, 485–489. [CrossRef]
15. Haupt, R.L. Antenna Design with a Mixed Integer Genetic Algorithm. *IEEE Trans. Antennas Propag.* **2007**, *55*, 577–582. [CrossRef]
16. Lebret, H.; Boyd, S. Antenna array pattern synthesis via convex optimization. *IEEE Trans. Signal Process.* **1997**, *45*, 526–532. [CrossRef]
17. Prisco, G.; D'Urso, M. Maximally sparse arrays via sequential convex optimizations. *IEEE Antennas Wirel. Propag. Lett.* **2012**, *11*, 192–195. [CrossRef]
18. Fuchs, B. Application of convex relaxation to array synthesis problems. *IEEE Trans. Antennas Propag.* **2014**, *62*, 634–640. [CrossRef]
19. Liu, Y.; Bai, J.; Xu, K.D.; Xu, Z.; Han, F.; Liu, Q.H.; Guo, Y.J. Linearly polarized shaped power pattern synthesis with sidelobe and cross-polarization control by using semidefinite relaxation. *IEEE Trans. Antennas Propag.* **2018**, *66*, 3207–3212. [CrossRef]
20. Lee, Y. Adaptive interference suppression of phase-only thinned arrays via convex optimization. *IEEE Trans. Antennas Propag.* **2020**, *68*, 4583–4592. [CrossRef]
21. Miao, Y.; Liu, F.; Lu, J.; Li, K. Synthesis of unequally spaced arrays with uniform excitation via iterative second-order cone programming. *IEEE Trans. Antennas Propag.* **2020**, *68*, 6013–6021. [CrossRef]
22. Gu, B.; Chen, Y.; Jiang, R.; Liu, X. Optimization of sparse cross array synthesis via perturbed convex optimization. *Sensors* **2020**, *20*, 4929. [CrossRef] [PubMed]
23. Lu, Q.; Cui, G.; Liu, R.; Yu, X. Beampattern synthesis via first-order iterative convex approximation. *IEEE Antennas Wirel. Propag. Lett.* **2021**, *20*, 1493–1497. [CrossRef]
24. Lee, Y. Adaptive beamforming with continuous/discrete phase shifters via convex relaxation. *IEEE Open J. Antennas Propag.* **2022**, *3*, 557–567. [CrossRef]
25. Fuchs, L.; Maki, A.; Gallstrom, A. Optimization method for wide beam sonar transmit beamforming. *Sensors* **2022**, *22*, 7526. [CrossRef]
26. Zhang, M.; Wang, X.; Zhang, A. An efficient broadband adaptive beamformer without presteering delays. *Sensors* **2021**, *21*, 1100. [CrossRef]
27. Shao, X.; Hu, T.; Zhang, J.; Li, L.; Xiao, M.; Xiao, Z. Efficient beampattern synthesis for sparse frequency diverse array via matrix pencil method. *Sensors* **2022**, *22*, 1042. [CrossRef]

28. Whipple, A.; Ruzindana, M.W.; Burnett, M.C.; Kunzler, J.W.; Lyman, K.; Jeffs, B.D.; Warnick, K.F. Wideband array signal processing with real-time adaptive interference mitigation. *Sensors* **2023**, *23*, 6584. [CrossRef]
29. CVX: Matlab Software for Disciplined Convex Programming, Version 2.2. Available online: http://cvxr.com/cvx (accessed on 1 January 2020).
30. Sturm, J.F. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optim. Methods Software.* **1999**, *11–12*, 625–653. [CrossRef]
31. Andersen, E.D.; Roos, C.; Terlaky, T. On implementing a primal-dual interior-point method for conic quadratic optimization. *Math. Program. Ser. B.* **2003**, *95*, 249–277. [CrossRef]
32. Alizadeh, F.; Haeberly, J.P.A.; Overton, M.L. Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM J. Optm.* **1998**, *8*, 746–768. [CrossRef]
33. Wright, S.J. *Primal-Dual Interior-Point Methods*; SIAM: Philadelphia, PA, USA, 1997.
34. Luo, Z.; Ma, W.; So, A.M.; Ye, Y.; Zhang, S. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Process. Mag.* **2010**, *27*, 20–34. [CrossRef]
35. Gregory, M.D.; Namin, F.A.; Werner, D.H. Exploiting rotational symmetry for the design of ultra-wideband planar phased array layouts. *IEEE Trans. Antennas Propag.* **2013**, *61*, 176–184. [CrossRef]
36. Mailloux, R.J. *Phased Array Antenna Handbook*; Artech House: London, UK, 2018.
37. Van Trees, H.L. *Optimum Array Processing*; John Wiley & Sons: New York, NY, USA, 2002.
38. Nai, S.E.; Ser, W.; Yu, Z.L.; Chen, H. Beampattern synthesis for linear and planar arrays with antenna selection by convex optimization. *IEEE Trans. Antennas Propag.* **2010**, *58*, 3923–3930. [CrossRef]
39. Cheng, M.; Wu, Q.; Yu, C.; Wang, H.; Hong, W. Synthesis of a thinned prephased electronically steered phased array using excitation control of both the small amplitude dynamic range ratio and low-resolution phase. *IEEE Trans. Antennas Propag.* **2024**, *72*, 600–613. [CrossRef]
40. Fan, X.; Liang, J.; Zhang, Y.; So, H.C.; Zhao, X. Shaped power pattern synthesis with minimization of dynamic range ratio. *IEEE Trans. Antennas Propag.* **2019**, *67*, 3067–3078. [CrossRef]
41. Zhang, X.; Wang, X.; So, H.C. Linear arbitrary array pattern synthesis with shape constraints and excitation range control. *IEEE Antennas Wirel. Propag. Lett.* **2021**, *20*, 1018–1022. [CrossRef]
42. Zhang, M.; Li, J.; Zhang, A. The robustness of pencil beam synthesis without considering sensor uncertainties. *IEEE Trans. Antennas Propag.* **2022**, *70*, 8608–8613. [CrossRef]
43. Alizadeh, F.; Goldfarb, D. Second-order cone programming. *Math. Program. Ser. B.* **2003**, *95*, 3–51. [CrossRef]
44. Isernia, T.; Iorio, P.D.; Soldovieri, F. An effective approach for the optimal focusing of array fields subject to arbitrary upper bounds. *IEEE Trans. Antennas Propag.* **2000**, *48*, 1837–1847. [CrossRef]
45. Zhang, M.; Li, J.; Zhu, S.; Chen, X. Fast and simple gradient projection algorithms for phase-only beamforming. *IEEE Trans. Veh. Technol.* **2021**, *70*, 10620–10632. [CrossRef]
46. Balanis, C.A. *Antenna Theory: Analysis and Design*, 4th ed.; John Wiley & Sons: New York, NY, USA, 2016.
47. Wang, J.; Zheng, Y.; He, Z. *Antenna Array Theory and Engineering Applications*; Publishing House of Electronics Industry: Beijing, China, 2015.
48. Zhang, M.; Chen, X.; Zhang, A. A simple tridiagonal loading method for robust adaptive beamforming. *Signal Process.* **2019**, *157*, 103–107. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.