Chapter

Convex Optimization and Array Orientation Diversity-Based Sparse Array Beampattern Synthesis

Hui Chen and Qun Wan

Abstract

The sparse array pattern synthesis (APS) has many important implications in some special situations where the weights, size, and cost of antennas are limited. In this chapter, the APS with a minimum number of elements problem is investigated from the perspective of sparseness constrained optimization. Firstly, to reduce the number of antenna elements in the array, the APS problem is formulated as sparseness constrained optimization problem under compressive sensing (CS) framework and solved by using the reweighted L1-norm minimization algorithm. Besides, to address left-right radiation pattern ambiguity problem, the proposed algorithm exploits the array orientation diversity in the sparsity constraint framework. Simulation results demonstrate the proposed method's validity of achieving the desired radiation beampattern with the minimum number of antenna elements.

Keywords: array beampattern synthesis, compressive sensing, array orientation diversity, convex optimization

1. Introduction

The objective of array pattern synthesis (APS) is to find the excitation of an array to produce a radiation beampattern which is close to the desired one. Dolph-Chebyshev method [1, 2] can be used to design an optimal pattern with the minimum sidelobe level and desired mainlobe width for a uniform linear array (ULA) with isotropic elements. While it is more difficult to solve the APS problem for an array of arbitrary geometric structures.

For nonuniformly spaced arbitrary arrays, there are several algorithms [3–6] that have been proposed to synthesize beampatterns. The design of thinned narrow-beam arrays has been well proposed in [3], which first fix element locations by eliminating the elements pair by pair according to the smallest possible sidelobe on the given interval and then optimize the weights via linear programming. For APS problem, which can also be formulated as a quadratic programming problem [4, 5], the objective function is to minimize the squared errors between the synthesized pattern and the desired pattern. Besides, additional linear constraints [4] or weighting functions [7] are also added to the quadratic objective function to minimize the peaks of the synthesis error. The challenge to weighting functions in the quadratic programming is that it has to be adjusted in an ad hoc manner. Besides, an inverse matrix has to be computed at each iteration for updating the weighting functions, which will result in high computation requirements, especially for large size of the array. The author of [8] proposed a recursive least squares method to solve the problem. Another kind of evolutionary algorithm, such as simulated annealing [9], particle swarm optimization [10], and genetic algorithm [11–13], has also been used for APS problem optimization.

Recently, second-order cone programming (SOCP) and semi-definite programming (SDP), as convex optimization techniques [14, 15], have been proposed to solve the APS problem readily by using SOCP solver and SDP solver, respectively. While a general nonuniform APS problem cannot be directly formulated as a convex problem. An iterative procedure [15] was proposed to optimize the array pattern by solving an SDP problem at each iteration. All the abovementioned approaches to design an optimal nonuniform array are to construct an objective function of minimizing the synthesis error or peak error. When the positions of elements are given, the nonuniformly spaced arrays can be optimized using convex programming like that for uniformly spaced arrays. While it is impossible to solve the APS problem by complex programming if the positions of the array elements are unknown. In addition, to solve the problem of occupying more elements to obtain the desired beampattern, the authors in [16] proposed a matrix pencil-based noniterative synthesis algorithm, which can efficiently save the number of elements in a very short computation time. Zhang et al. [17] formulated the APS problem as a sparseness constrained optimization problem and solved the problem by using Bayesian compressive sensing (BCS) inversion algorithm; the authors in [18] proposed an approach for APS of linear sparse arrays, and then the multitask BCS has been used to design 2D sparse synthesis problem [19], sparse conformal array synthesis problem [20–22], and another CS-based sparse array synthesis problem [23–26].

In this chapter, we proposed an array pattern synthesis algorithm [27] by using reweighted l_1-norm minimization [28] and convex optimization [29]. Then we extended our work to a new version [30] by using reweighted l_1-norm minimization and array orientation diversity. Merits of the algorithm include the following: (1) it does not need a thorough search in the multidimensional parameter space, and (2) it can achieve the same array performance with fewer antenna elements when the array size is given and thus reduces the array cost significantly. Regarding the notation of this chapter, $(\cdot)^T$ represents the transpose operation of a vector or matrix, $|\cdot|$ denotes the absolute value operator, and $\|\cdot\|_1$ and $\|\cdot\|_\infty$ represent the l_1-norm and l_∞-norm of a vector or matrix, respectively. And $\lceil x \rceil$ denotes the smallest integer not less than x, and $diag(x)$ means the diagonal matrix with the main diagonal elements equaled to the vector x.

2. Nonuniform array pattern synthesis using reweighted l_1-norm minimization

2.1 Problem formulation

Consider a narrowband linear array with M isotropic antennas located at $x_1, \ldots, x_M \in R^2$. Assume that a harmonic plane wave with wavelength λ propagates across the array with incident direction θ. The M signal outputs s_i are converted to the baseband, weighted by the weights w_i, and summed. Then the array response can represented as

$$G(\theta) = \sum_{i=1}^{M} w_i \exp\left(j2\pi x_i \sin\theta/\lambda\right) = \boldsymbol{w}^T \boldsymbol{a}(\theta) \tag{1}$$

where $\phi_i = 2\pi x_i \sin\theta/\lambda$ is the phase delay due to propagation, complex weight vector $\boldsymbol{w} = [w_1, \ldots, w_M]^T \in C^M$, and the steering vector $\boldsymbol{a}(\theta)$.

Let $G_d(\theta)$ be the desired array response at the direction θ. The APS problem is to find the complex weight vector $\boldsymbol{w}$ such that $G(\theta) = G_d(\theta)$ for all $\theta \in [-90^\circ, 90^\circ]$. For the array described above, how well $G(\theta)$ approximates $G_d(\theta)$ can be measured by using the peak error across θ, i.e.,

$$\min_{\boldsymbol{w}} \max_{\theta\in\Theta} |G(\theta) - G_d(\theta)| \tag{2}$$

where $\Theta \in [-90^\circ, 90^\circ]$ is a dense set of arrival angles that we are of interest. The goal of the proposed algorithm is to find both optimal antenna locations and corresponding weights that approach the desired array pattern as well as possible.

2.2 The proposed algorithm

The APS problem can be formulated as a following estimation problem:

$$\boldsymbol{w}^T \boldsymbol{a}(\theta) = G(\theta), \forall\theta \in \Theta \tag{3}$$

We try to find $\boldsymbol{w}$ in Eq. (3) such that Eq. (2) is satisfied.

The new solution of Eq. (3) can be summarized as follows:

2.2.1 Creating a virtual array

For a given array size, to obtain more elements than those of a conventional array with $\lambda/2$ inter-element spacing, we first create a dense uniformly spaced linear array with much smaller inter-element spacing than conventional array and initialize a weight matrix $\boldsymbol{Q}$ as an identity matrix to create a more sparse array in subsequent processing.

2.2.2 Finding the sparse weight vector

The specified synthesized pattern $G(\theta)$ is produced by a weight vector. The weight vector can be obtained by solving the following weighted l_1-norm minimization convex problem Eq. (4), which is subject to minimizing the peak of the error between the synthesized pattern $G(\theta)$ and the desired pattern $G_d(\theta)$:

$$\begin{aligned} &\text{Minimize } \ \|\boldsymbol{Qw}\|_1 \\ &\text{Subject to } \ \|G(\theta) - G_d(\theta)\|_\infty \le \varepsilon, \forall\theta \in [-90^\circ, 90^\circ] \end{aligned} \tag{4}$$

where ε is the fitting error between the synthesized pattern and desired pattern. Minimizing $\|\boldsymbol{Qw}\|_1$ makes the vector $\boldsymbol{Qw}$ sparse, which is useful to create a nonuniformly spaced array. According to the situation that some weights of the original weight vector $\boldsymbol{w} = [w_1, w_2, \ldots]^T$ from Eq. (4) are very small, they can be deleted without significantly decreasing the array performance. So a sparse weight vector can be obtained by retuning the small value elements of the original weight vector, that is, the w_i will be retained if $|w_i|/\|\boldsymbol{w}\|_\infty > \eta$ $(i = 1, 2, \ldots)$, otherwise $w_i = 0$. The η is a designed threshold whose value should make a trade-off between

APS performance and convergence rate. Because more elements of the original weight vector will be pruned if the threshold value η is increased, which make us probably cannot find the optimal array element positions of the array, correspondingly the array synthesis performance is not optimal for a given array element number. Conversely, if the threshold value is decreased, less elements of the original weight vector will be pruned in each iteration, which increases the algorithm complexity. So we should make a good balance between APS performance and convergence rate when setting the value of η.

2.2.3 Updating the weight matrix

After obtaining the original weight vector $\boldsymbol{w} = [w_1, w_2, \ldots]^T$, the weight matrix $\boldsymbol{Q}$ is updated as $\boldsymbol{Q} = diag([(|w_1| + \delta)^{-p},\ (|w_2| + \delta)^{-p},\ \ldots])$ (usually, p is an integer greater than 1; it was demonstrated experimentally that $p = 2$ is a better choice for our APS problem). To ensure that the weight matrix is effectively updated when a zero-valued component in $\boldsymbol{w}$, we introduce a parameter $\delta > 0$. It is empirically demonstrated that δ should be set slightly smaller than the expected nonzero magnitudes of $\boldsymbol{w}$.

2.2.4 Forming the nonuniform array

The sparse weight vector $\boldsymbol{w}_s$ is obtained by pruning the original weight vector, and then the antenna elements corresponding to nonzero-valued indices of $\boldsymbol{w}_s$ are retained to form a nonuniform array with fewer elements.

The above steps (A, B, C) are repeated until the final synthesized array performance is satisfactory or the specified maximum number of iterations is attained.

2.2.5 Optimizing the sparse weight vector

After obtaining the array antenna positions by steps (B, C, D), the optimal weight vector $\boldsymbol{w}_{opt}$ is further obtained by solving following convex optimization problem, which is to improve the performance of the array beampattern synthesized by the sparse weight vector:

$$\begin{aligned}&\text{Find } \boldsymbol{w}_{opt}\\&\text{Minimize } \|G(\theta) - G_d(\theta)\|_\infty, \theta \in [-90^\circ, 90^\circ]\end{aligned} \tag{5}$$

2.3 Computer simulation and discussions

Given the array aperture, the objective is to design an array with the beampattern as shown in **Figure 1**, where region $|\theta| \le \theta_s$ corresponds to the mainlobe and region $|\theta| \ge \theta_s$ corresponds to the sidelobe. We set $\theta_s = 2.5^\circ$, and the angle grid of the interval $[-90^\circ, 90^\circ]$ is 1°. We design a virtual ULA with the array aperture of 25.5λ having a uniform inter-element spacing of $\lambda/8$.

The beampattern of **Figure 2** is obtained by using our approach for a 19-element array, and the optimal beampattern exhibits the maximum sidelobe of −15.46 dB. The optimal antenna positions and the corresponding weights are displayed in **Table 1**. The designs proposed in [3, 17] describe a 25-element and a 29-element non-ULA with the approximate desired array pattern shown in **Figure 2**, respectively. The 25-element array beampattern described in [3] by eliminating the elements pair by pair has a maximum sidelobe −13.75 dB, while the maximum sidelobe of 29-element array beampattern obtained by the BCS algorithm [17] is −13.165 dB.

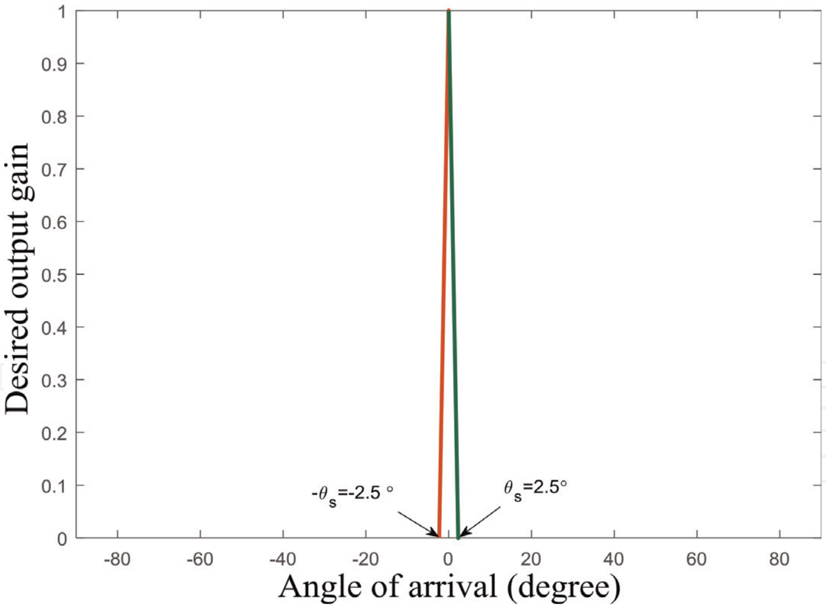


Figure 1.
The desired beampattern.

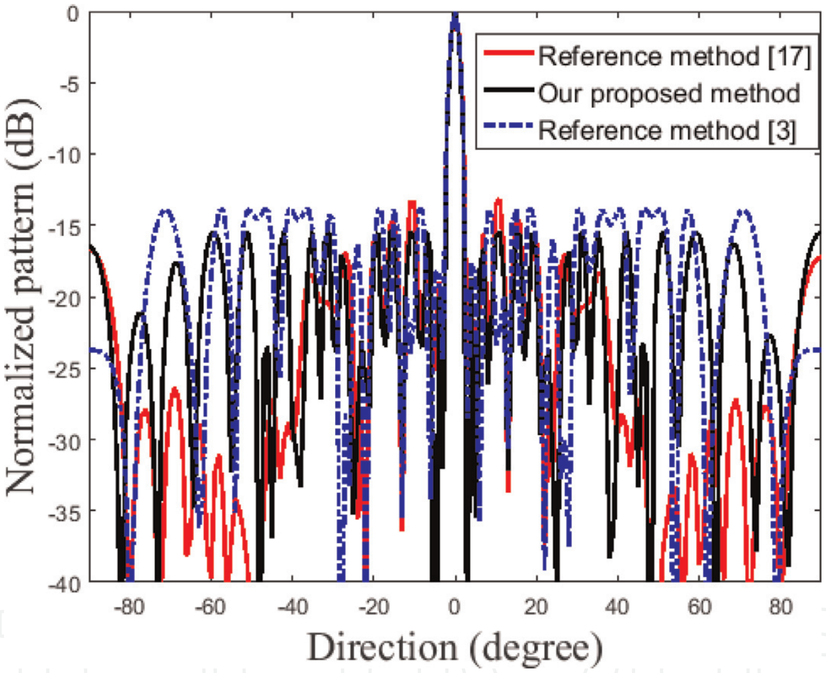


Figure 2.
A "25-element array beampattern obtained by [3] and a 29-element array beampattern obtained by [17]" vs. "our 19-element array beampattern."

Element indices	Position (λ)	Weight value	Element indices	Position (λ)	Weight value
1,19	±10.500	0.2289	6,14	±3.875	0.2677
2,18	±8.625	0.2583	7,13	±2.875	0.2195
3,17	±7.6250	0.2207	8,12	±1.875	0.1813
4,16	±6.750	0.2904	9,11	±1.000	0.2347
5,15	±4.750	0.1567	10	0	0.2427

Table 1.
Our element positions and weights in a 19-element antenna array.

The antenna positions and the corresponding weights of the two methods in [3, 17] are listed in **Tables 2** and **3**, respectively. Compared with the method in [3], we can see from **Table 1** and **Figure 2** that our proposed algorithm saves six elements without reducing the array performance and the our minimum inter-element spacing of the non-ULA is 0.375λ larger than that of the method of eliminating the elements pair by pair [3]. Compared with the reference method in [17], our proposed method offers an economization of 10 elements as well as 2.3 dB performance improvement, and the minimum inter-element spacing of the sparse array designed by our approach is 0.75λ larger than that of the reference array [17]. We also emphasize that the reference array [17] has 4.5λ larger array aperture than that of our array.

The proposed APS algorithm based on convex optimization and reweighted l_1-norm minimization is proven to be effective in reducing array elements, suppressing the sidelobe, and reducing the aperture. This simple and effective design method can be extended to solving the 2D array synthesis problem.

Element indices	Position (λ)	Weight value	Element indices	Position (λ)	Weight value
1,25	±12.0	0.2100	8,18	±4.5	0.1924
2,24	±8.5	0.2605	9,17	±3.5	0.2296
3,23	±8.0	0.2276	10,16	±2.0	0.2282
4,22	±7.5	0.2554	11,15	±1.5	0.0876
5,21	±7.0	0.2103	12,14	±0.5	0.1143
6,20	±6.0	0.200	13	0	0.2084
7,19	±5.0	0.2037			

Table 2.
Element positions and weights obtained in a 25-element array [3].

Element indices	Position (λ)	Weight value
1,2	−1.375, −0.500	0.0876, 0.1178
3,4	0.375, 0.500	0.0532, 0.1025
5,6	1.250, 1.375	0.0497, 0.1844
7,8	2.125, 4.375	0.1895, 0.1086
9,10	4.500, 5.250	0.0778, 0.2679
11,12	6.125, 7.000	0.2440, 0.1098
13,14	7.125, 8.125	0.0643, 0.2400
15,16	8.875, 10.125	0.1953, 0.2249
17,18	11.000, 11.750	0.2297, 0.0720
19,20	12.000, 12.750	0.1480, 0.1810
21,22	13.750, 13.875	0.0761, 0.0554
23,24	14.875, 15.750	0.0840, 0.1833
25,26	16.625, 16.750	0.1860, 0.0516
27,28	17.375, 19.625	0.1625, 0.0745
29	24.125	0.0317

Table 3.
Element positions and weights obtained by the BCS inversion algorithm [17].

3. Beampattern synthesis using reweighted l_1-norm minimization and array orientation diversity

To address left-right radiation pattern ambiguity problem, we allow exploitation of the array orientation diversity in the CS framework.

3.1 Problem formulation

We assume that transmit signals and the array are coplanar, so the antenna array synthesis problem can be described as follows:

$$\min(DM) \quad \text{s.t.} \left\{ \min_{\substack{\{R_{\alpha i}, d_{\alpha i}\} \; \alpha = 1, \dots, D \\ i = 1, \dots, M}} \| F_d(\theta) - F(\theta) \|_{l_2} \right\} \le \xi \tag{6}$$

where $F(\theta) = \sum_{\alpha=1}^{D} \sum_{i}^{M} R_{\alpha i} e^{jkd_{\alpha i}\cos(\theta - \theta_\alpha)}$, $F_d(\theta)$ is the desired radiation pattern, M is the number of identical antenna elements in each linear array, $R_{\alpha i}$ is the excitation coefficient of the ith element located at $d_{\alpha i}$ in the αth array, k is the wavenumber in the free space, and D array orientations $\theta_\alpha (\alpha = 1, \dots, D)$. The objective of the problem is to synthesize the desired radiation pattern $F_d(\theta)$ with the minimum number of elements under a small tolerance error ξ. For one linear array at orientation θ_α to the incident plane wave from the bearing θ, the array factor is given by

$$F_\alpha(\theta) = \sum_{i}^{M} R_{\alpha i} e^{jkd_{\alpha i}\cos(\theta - \theta_\alpha)} \tag{7}$$

Suppose that all the antenna elements in each array orientation $\theta_\alpha (\alpha = 1, \dots, D)$ are symmetrically distributed within a range of $-d_s$ to d_s along the array orientation θ_α, respectively, the combination pattern of all the linear orientation arrays can be written as

$$F(\theta) = \sum_{\alpha=1}^{D} F_\alpha(\theta) \tag{8}$$

In order to solve Eqs. (7) and (8), we can assume that all the antenna elements are equally spaced from $-d_s$ to d_s with a small inter-element spacing Δd. Although it is supposed that there is one element at each position, not each antenna element is necessarily radiating waves or excited with current. All the antenna elements can be in two states: "on" states (when the element is in the supposed position or has an excitation) and "off" state (when there is no element in the supposed position or without an excitation). Through discretization, Eq. (8) can be written in a matrix form:

$$[F(\theta)]_{h \times 1} = [H]_{h \times n} [r]_{n \times 1} \tag{9}$$

where h is the number of sampled antenna radiation pattern, $n = D\left\lceil \frac{2d_s}{\Delta d} \right\rceil$, the sensing radiation pattern at different angles is contained in vector $\mathbf{F} = [F(\theta_1)\, F(\theta_2) \cdots F(\theta_h)]^T$, overcomplete dictionary $\mathbf{H}$ is an $h \times n$ matrix whose (i, l)th element is $\mathbf{H}_{il} = e^{jkd_{\alpha i}\cos(\theta_l - \theta_\alpha)}$, $l \in \left((\alpha - 1)\frac{n}{D} + 1, \frac{n}{D}\alpha\right)$ for $\alpha \in \{1, \dots, D\}$, and $h \ll n$. $\boldsymbol{r}$ is an excitation vector, $R_{\alpha i} = 0$ means the antenna in the lth position of the

αth array is absent from the supposed position, and the solution of sparse excitation vector $\boldsymbol{r}$ can be casted as the following convex optimization problem:

$$\begin{aligned} &\min \quad \|\boldsymbol{r}\|_1 \\ &\text{subject to} \quad \|\mathbf{F}\text{-}\mathbf{H}\boldsymbol{r}\|_\infty \le \xi \end{aligned} \tag{10}$$

In Eq. (10) the smallest number of nonzero elements in the excitation vector $\boldsymbol{r}$ can be obtained readily by using existing software package, such as CVX [31].

3.2 The proposed algorithm

In this subsection, the new solution of Eq. (6) can be summarized as follows:

3.2.1 Initializing a virtual array and a weight matrix

To place more antenna elements than those of a conventional array with the same array size, we first create D virtual linear orientation arrays with much smaller interspacing $\lambda/16$ (in general, the inter-element spacing of the conventional ULA is $\lambda/2$). Using the reweighted l_1-norm minimization in the following step, we set a $DM \times DM$ dimension weight matrix $\mathbf{Q}$ as a unit matrix.

3.2.2 Finding the sparse weight vector

Let $F(\theta)$ be a synthesized beampattern by using a weight vector, and the weight vector can be obtained by solving the following weighted l_1-norm minimization convex problem which is to try to minimize the peak value of the error between the synthesized pattern and the desired pattern:

$$\begin{aligned} &\text{Minimize} \quad \|\boldsymbol{Qw}\|_1 \\ &\text{Subject to} \quad \|F(\theta) - F_d(\theta)\|_\infty \le \zeta, \forall\theta \in [-180^\circ, 180^\circ] \end{aligned} \tag{11}$$

where ζ is the fitting error between the synthesized pattern and the desired one. Minimizing $\|\boldsymbol{Qw}\|_1$ makes the vector $\boldsymbol{Qw}$ sparse, which is useful to create D nonuniformly spaced linear orientation arrays. Here, let the weight vector $\boldsymbol{w} = [w_1, w_2, \ldots]^T$ obtained from Eq. (11) be the original weight vector for convenience. The weighted l_1-norm minimization will make some weights of the original weight vector be very small, so they can be adjusted to zero without significantly reducing the array performance. That is, if the absolute value of an element from the original weight vector is smaller than a threshold which is set according to the array performance requirement, the element will be assigned zero; otherwise, the element will be retained. Thus the sparse weight vector $\boldsymbol{w}_s$ is obtained.

3.2.3 Updating the weight matrix

After obtaining the original weight vector $\boldsymbol{w} = [w_1, w_2, \ldots]^T$ from step (2), the weight matrix $\boldsymbol{Q}$ is updated according to $\boldsymbol{Q} = diag([(|w_1| + \delta)^{-p},\ (|w_2| + \delta)^{-p},\ \ldots])$ in each iteration; usually, p is an integer greater than 1, while it was demonstrated experimentally that $p = 2$ is a better choice for our APS problem. To ensure regular update $\boldsymbol{Q}$ especially for zero-valued components in $\boldsymbol{w}$, we bring in the parameter $\delta > 0$ which should be set slightly smaller than the expected nonzero magnitudes of $\boldsymbol{w}$. Reweighted l_1 minimization can improve the signal reconstruction performance.

3.2.4 Creating the nonuniform arrays

After obtaining the sparse weight vector $\boldsymbol{w}_s$ from step (4), the antenna elements corresponding to nonzero-valued indices of the sparse weight vector are retained to create D sparse linear arrays with different orientations.

Repeat steps (2, 3, and 4) until the synthesized array beampattern performance is satisfactory or the specified maximum number of iterations or minimum antenna number is attained.

3.2.5 Finding the optimal weight vector

After optimizing the antenna element positions by the above steps, we introduce convex optimization to obtain the optimal weight vector which can further improve the performance of the array beampattern synthesized by the sparse weight vector:

$$\begin{aligned} &\text{Find } \boldsymbol{w}_{opt} \\ &\text{Minimize } \|F(\theta) - F_d(\theta)\|_\infty, \theta \in [-90^\circ, 90^\circ] \end{aligned} \tag{12}$$

The optimal sparse weight vector $\boldsymbol{w}_{opt}$ can be obtained from Eq. (12) readily.

3.3 Computer simulations and discussion

The objective is to design an array with the desired beampattern for given the array physical size, as shown in **Figure 1**, where region $|\theta| \le \theta_s$ belongs to the mainlobe and region $|\theta| \ge \theta_s$ corresponds to the sidelobe. We set $\theta_s = 2.3^\circ$, and the angle grid for the search area $[-180^\circ, 180^\circ]$ is 2°, that is, we take a "dense set" of $[-180^\circ, 180^\circ]$ with the angles sampled at 2° from -180° to 180° (**Figure 3**).

To show the performance of our beampattern synthesis, we will consider two cases, same element number array and same beampattern performance, since all formulated problems in Eqs. (6), (10), (11), and (12) are convex, so we adopt the optimization toolbox to solve the formulated problems.

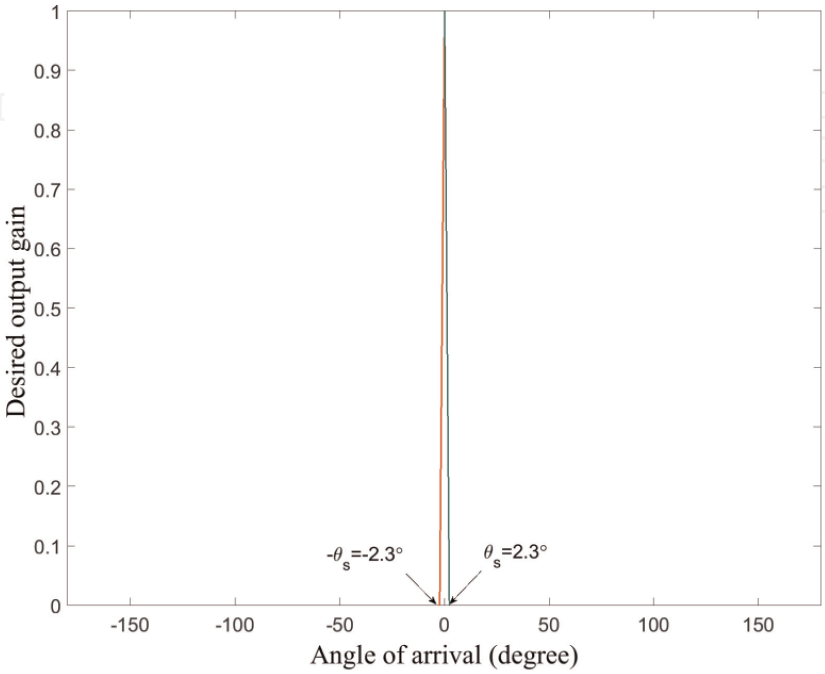
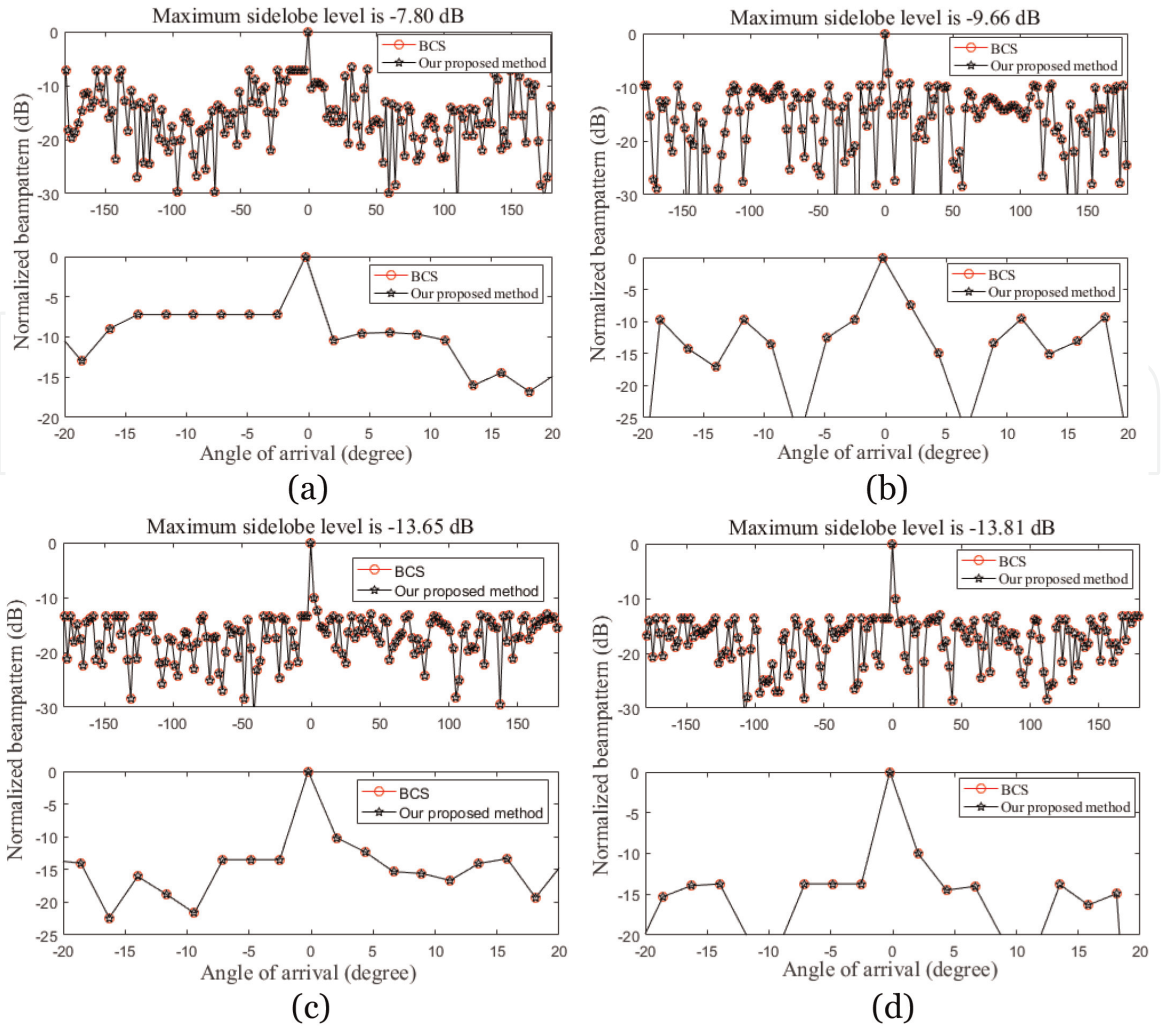


Figure 3.
Desired beampattern.



(a)

(b)

(c)

(d)

Figure 4.
A 19-element array performance obtained by BCS inversion algorithm [17] and our method with increasing array orientation diversity. (a) 1 array orientation, (b) 2 array orientations, (c) 3 array orientations, and (d) 4 array orientation.

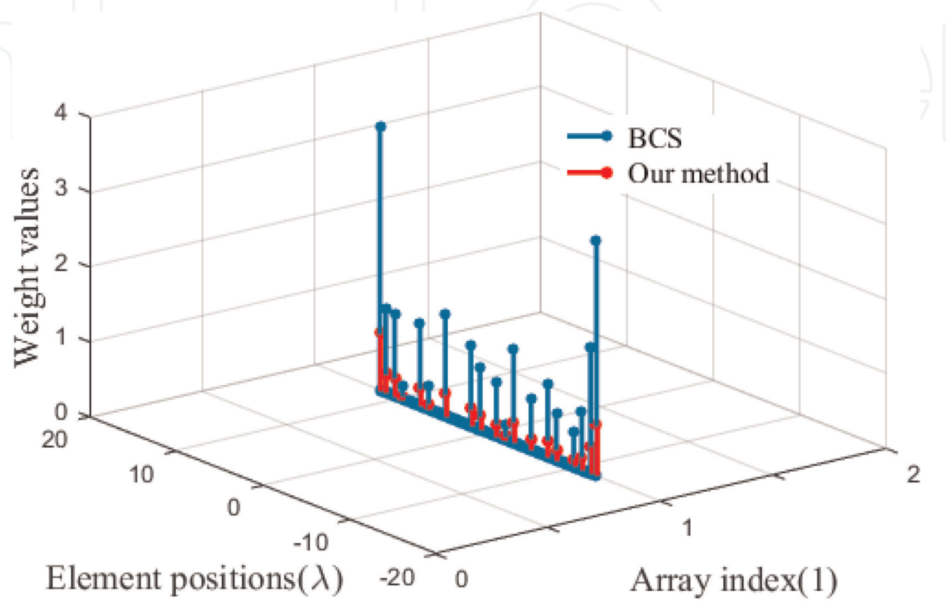


Figure 5.
Element positions and excitation amplitudes in a 19-element one-array orientation.

3.3.1 Same element number array with array orientation diversity

In this section, we analyzed the influence of the array orientation diversity on the beampattern synthesis by simulation results. We initialize four virtual ULAs (named Array 1, Array 2, Array 3, Array 4, with orientation -10°, 0°, 10°, 20°, respectively) with each subarray aperture of 25λ owning a uniform interspacing $\lambda/8$. Besides, we initialize $\boldsymbol{Q}$ as a unit matrix and choose $\delta = 10^{-4}$ and $p = 2$ in our simulations. **Figure 4** shows a 19-element beampattern synthesis performance in four cases with one-, two-, three-, and four-array orientations. From **Figure 4**, we can see that our proposed method and BCS algorithm can improve performance with increasing array orientation diversity (from 1 to 4); the optimal antenna positions and the corresponding excitation amplitudes of the four cases are displayed in **Figures 5**–**8**, respectively. Note that for the four cases of **Figures 5**–**8**, the required normalized radiated energies of BCS approach [17] are correspondingly bigger than that of our proposed method.

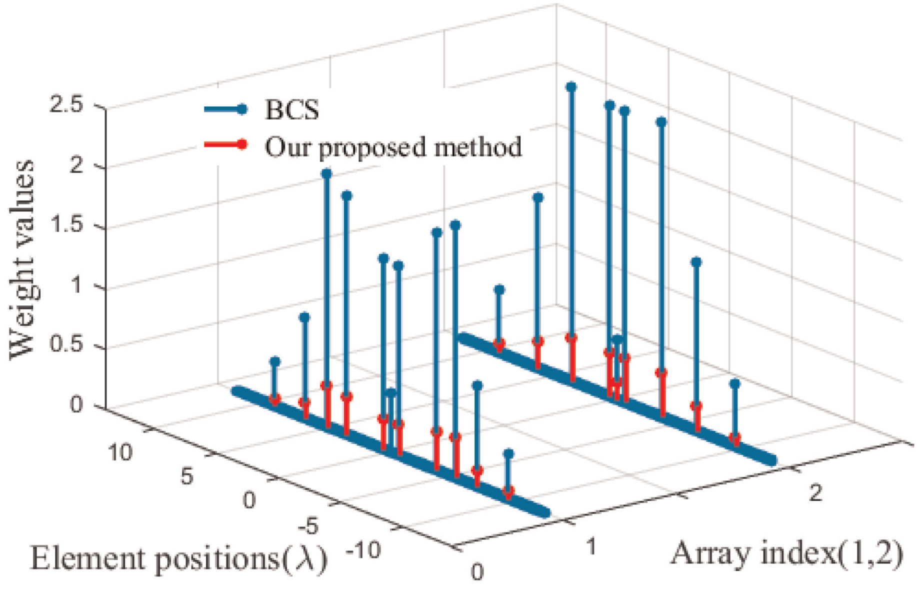


Figure 6.
Element positions and excitation amplitudes in a 19-element two-array orientation.

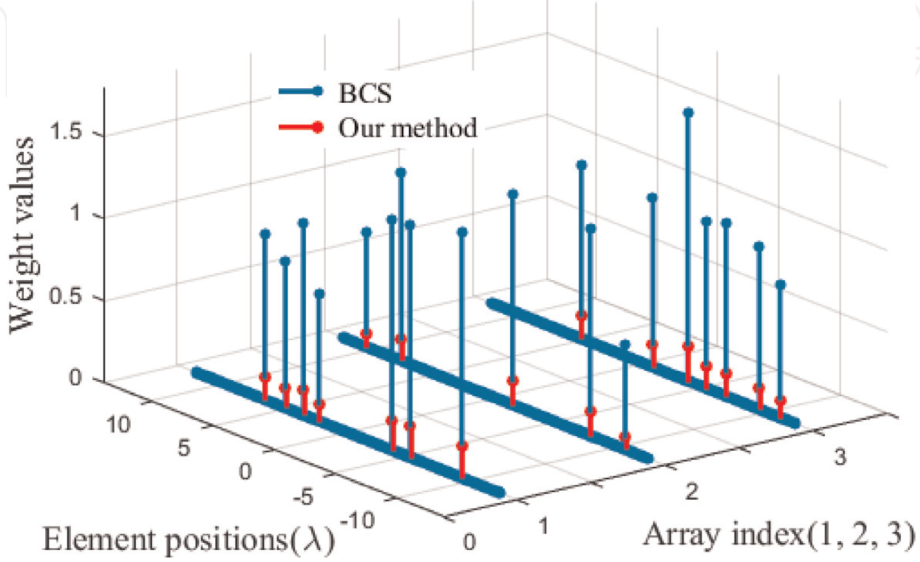


Figure 7.
Element positions and excitation amplitudes in a 19-element three-array orientation.

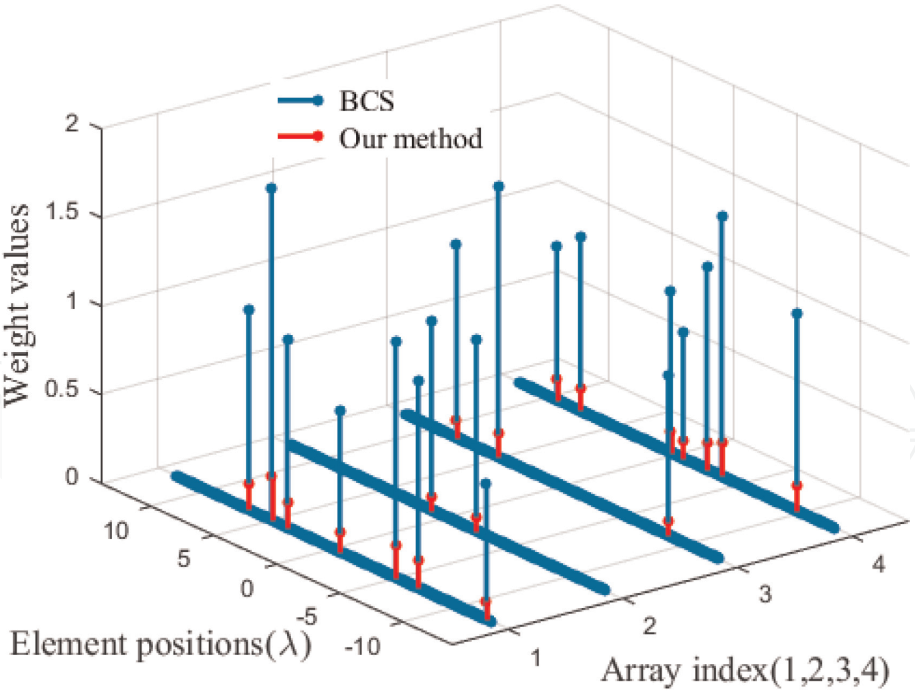


Figure 8.
Element positions and excitation amplitudes in a 19-element four-array orientation antenna.

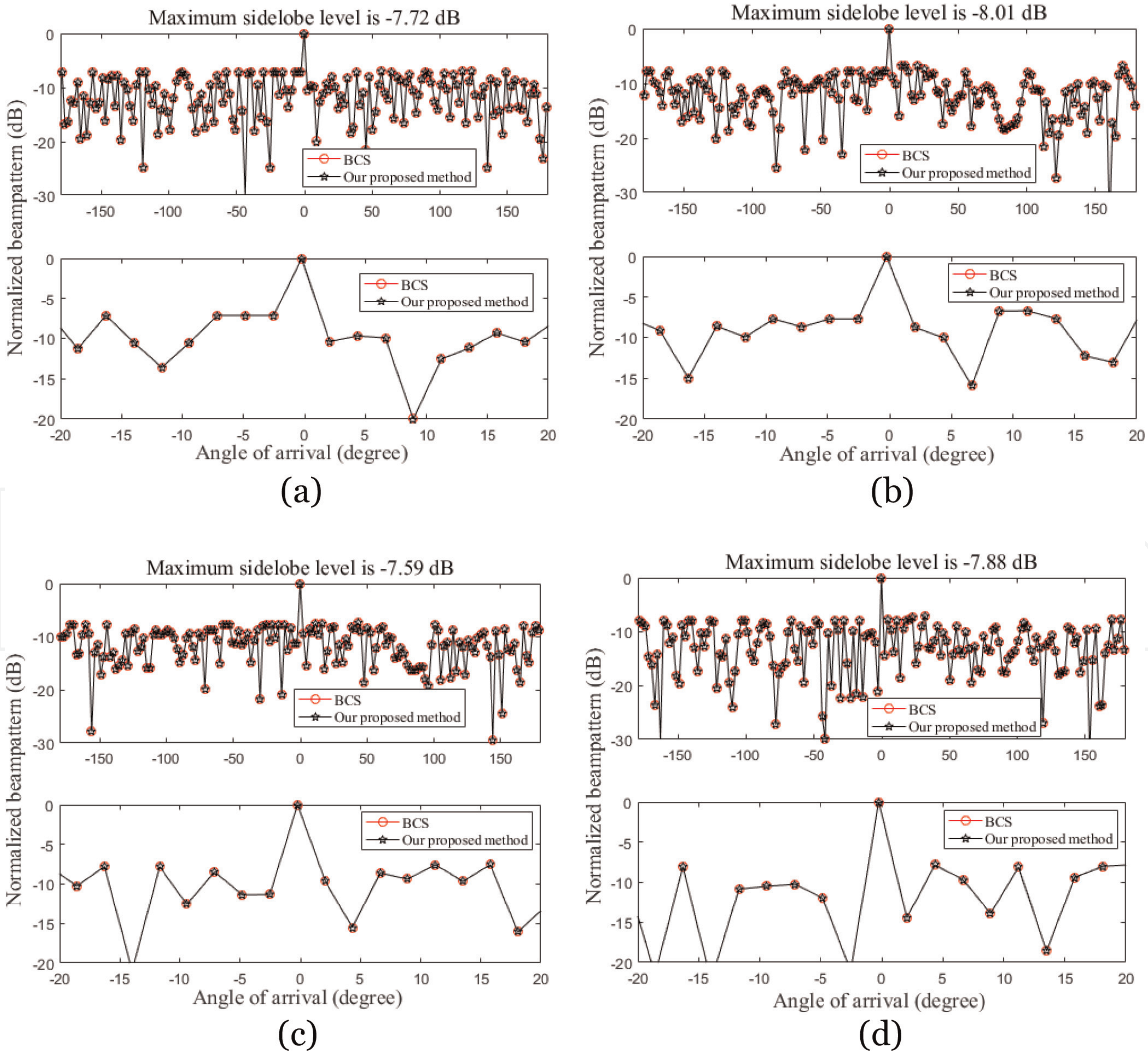


Figure 9.
Optimal beampattern of different element number array by using "BCS inversion algorithm [17]" vs. "our method." (a) 1 array orientation, (b) 2 array orientations, (c) 3 array orientations, and (d) 4 array orientations.

3.3.2 Approximate beampattern performance with array orientation diversity

To demonstrate another advantage of array orientation diversity, we examine the beampattern synthesis of an 18-element array, 11-element array, 10-element array, and 10-element array correspondingly with one orientation, two orientations, three orientations, and four orientations using BCS algorithm and our method, respectively. The optimal beampatterns exhibit maximal sidelobes of −7.72, −8.01, −7.59, and −7.88 dB, respectively, which are shown in **Figure 9**. **Figures 10**–**13** provide all the corresponding antenna positions and excitation amplitudes for all the four cases mentioned above. Obviously, given the array size, using orientation diversity can economize seven (or eight) elements without reducing the array performance. But more diversity is not always better enough, as shown in **Figures 11**–**13**. Besides, the excitation amplitudes in **Figures 10**–**13** show that our proposed method needs less radiation energy for all four cases.

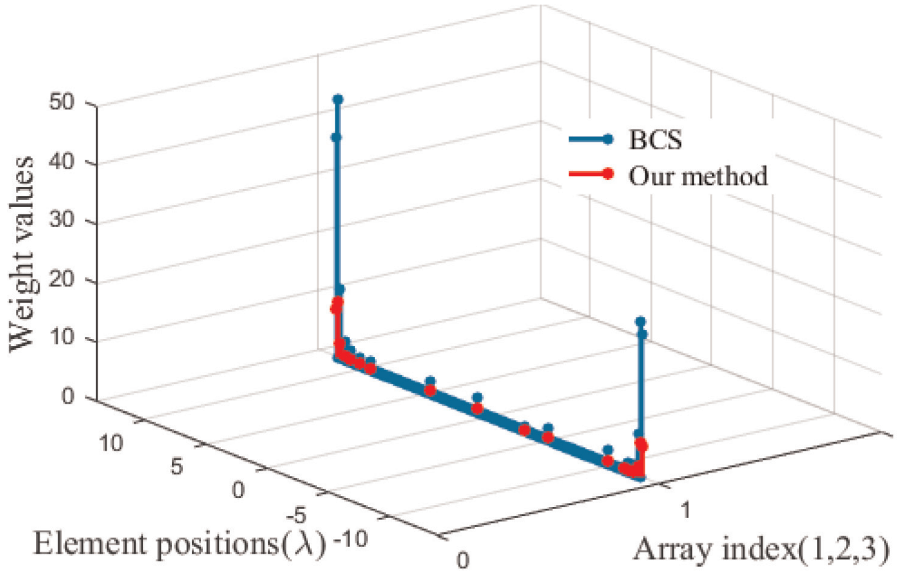


Figure 10.
Element positions and excitation amplitudes in an 18-element one-array orientation antenna.

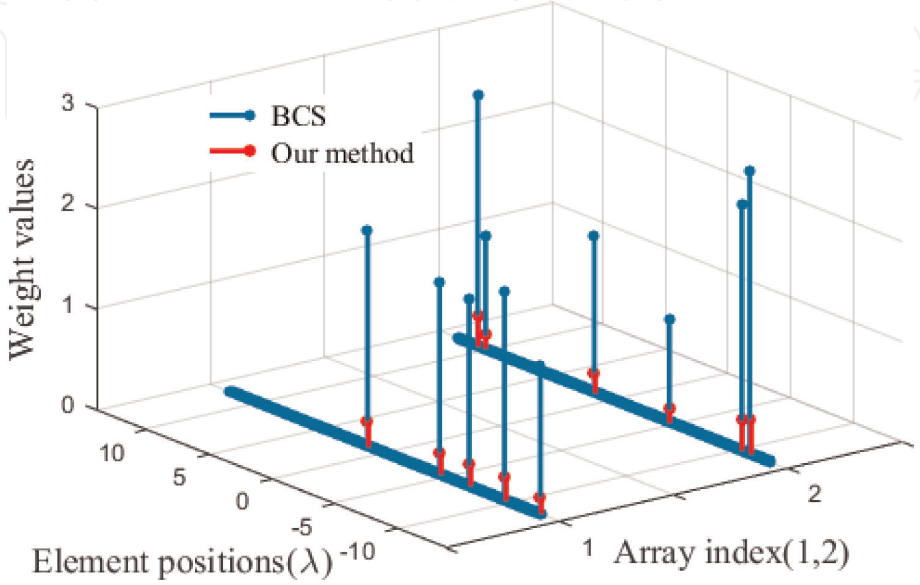


Figure 11.
Element positions and excitation amplitudes in an 11-element two-array orientation antenna.

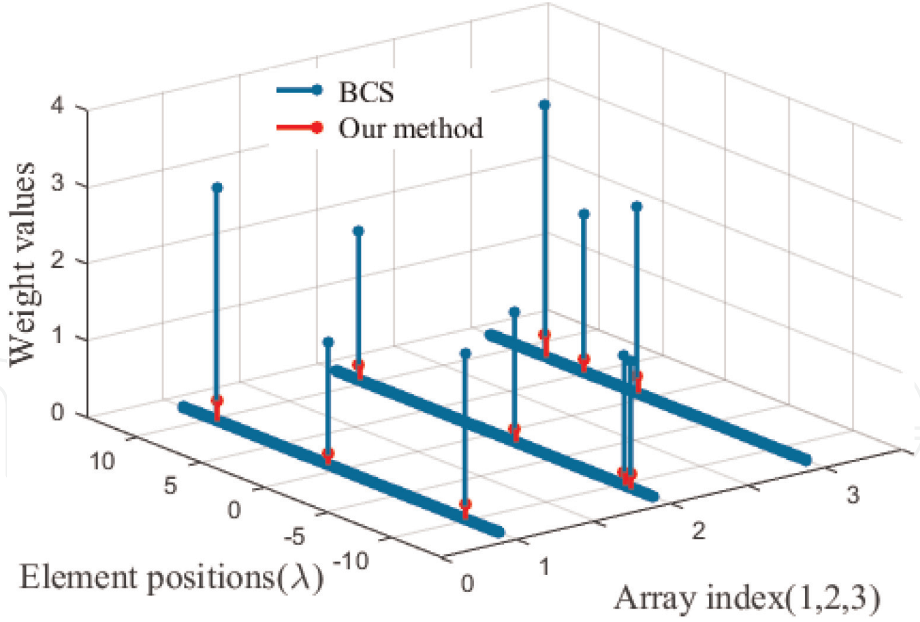


Figure 12.
Element positions and excitation amplitudes in a 9-element three-array orientation antenna.

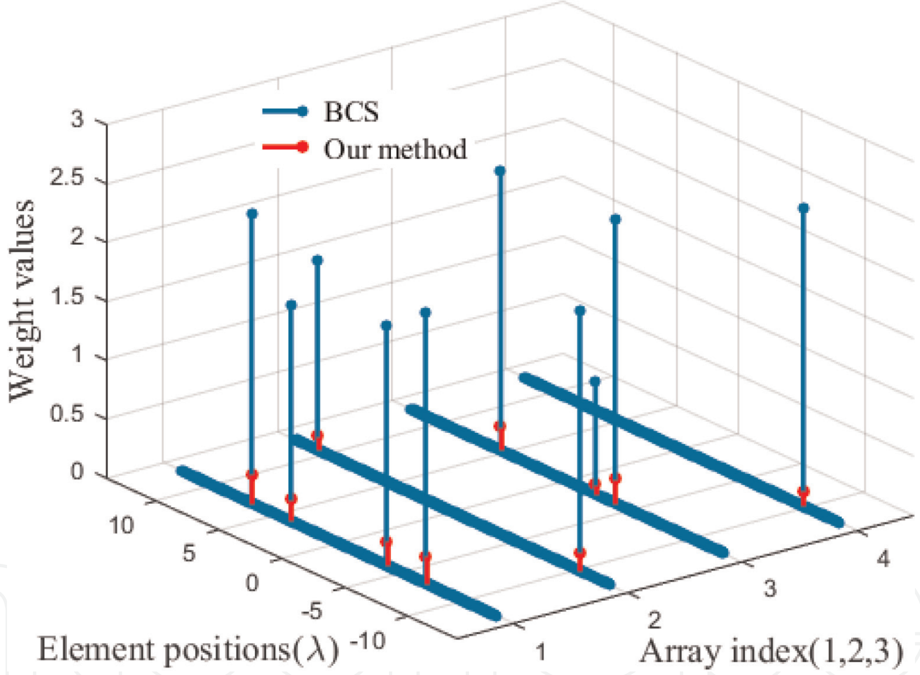


Figure 13.
Element positions and excitation amplitudes in a 10-element four-array orientation antenna.

The proposed APS algorithm based on reweighted l_1-norm minimization and array orientation diversity is demonstrated to be effective in reducing array elements, suppressing the sidelobe, and reducing the energy consumption to some extent.

4. Conclusions

This chapter focuses on the APS problem with sparse antenna array, which has practical applications, especially for massive antenna array. By using array orientation diversity and solving reweightedl_1-norm minimization convex optimization problem, the proposed APS algorithm shows the superiority in reducing array elements, suppressing the sidelobe, and reducing the energy consumption to some extent, and the robustness of the proposed design tool in real-life application will also be considered in our further work.

Acknowledgements

This work was supported in part by Sichuan Science and Technology Program (No. 18ZDYF2551) and in part by Fundamental Research Funds for the Central Universities (Program No. ZYGX2018J005).

Author details

Hui Chen* and Qun Wan
School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, PR China

*Address all correspondence to: huichen0929@uestc.edu.cn

References

[1] Dolph CL. A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level. Proceedings of the IRE. 1946;**34**(6):335-348. DOI: 10.1109/JRPROC.1946.225956

[2] Sarkar TK, Pereira O. Using the matrix pencil method to estimate the parameters of a sum of complex exponentials. IEEE Antennas and Propagation Magazine. 1995;**37**(1): 48-55. DOI: 10.1109/74.370583

[3] Jarske P, Saramaki T, Mitra SK, Neuvo Y. On properties and design of nonuniformly spaced linear arrays. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1988;**36**(3):372-380. DOI: 10.1109/29.1534

[4] Tseng C, Griffiths LJ. A simple algorithm to achieve desired patterns for arbitrary arrays. IEEE Transactions on Signal Processing. 1992;**40**: 2737-2746. DOI: 10.1109/78.165660

[5] Ng BP, Er MH, Kot C. A flexible array synthesis method using quadratic programming. IEEE Transactions on Antennas and Propagation. 1993;**41**: 1541-1550. DOI: 10.1109/8.267354

[6] Haupt RL. Antenna Arrays: A Computational Approach. New Jersey: Wiley; 2010. DOI: 10.1002/9780470937464.ch5

[7] Zhou P, Ingram M. Pattern synthesis for arbitrary arrays using an adaptive array method. IEEE Transactions on Antennas and Propagation. 1999;**47**: 862-869. DOI: 10.1109/8.774142

[8] Wang F, Yang R, Frank C. A new algorithm for antenna array pattern synthesis using recursive least square method. IEEE Transactions on Signal Processing Letter. 2003;**10**(8):235-238. DOI: 10.1109/lsp.2003.814398

[9] Murino V, Trucco A, Regazzoni CS. Synthesis of unequally spaced arrays by simulated annealing. IEEE Transactions on Signal Processing. 1996;**44**(1): 119-123. DOI: 10.1109/78.482017

[10] Khodier MM, Christodoulou CG. Linear array geometry synthesis with minimum side lobe level and null control using particle swarm optimization. IEEE Transactions on Antennas and Propagation. 2005;**53**(8): 2674-2679. DOI: 10.1109/TAP.2005.851762

[11] Rattan M, Patterh MS, Sohi BS. Synthesis of aperiodic liner antenna arrays using genetic algorithm. In: Proceedings of the 2007 19th International Conference on Applied Electromagnetics and Communications; 24-26 September 2007; Dubrovnik, Croatia: IEEE; 2008. pp. 1-4

[12] Rahmat-Samii Y, Michielssen E. Electromagnetic Optimization by Genetic Algorithms. New Jersey: Wiley; 1999. DOI: 10.1364/IPR.1999.RTuE4

[13] Haupt RL, Werner DH. Genetic Algorithms in Electromagnetics. New Jersey: Wiley; 2007. DOI: 10.1109/APS.1996.549878

[14] Lebret H, Boyd S. Antenna array pattern synthesis via convex optimization. IEEE Transactions on Signal Processing. 1997;**45**(3):526-532. DOI: 10.1109/78.558465

[15] Wang F, Balakrishnan V, Zhou PY, Chen J, Yang R, Frank C. Optimal array pattern synthesis using semidefinite programming. IEEE Transactions on Signal Processing. 2003;**51**(5):1172-1183. DOI: 10.1109/TSP.2003.810308

[16] Liu Y, Nie Z, Liu Q. Reducing the number of elements in a linear antenna array by the matrix pencil method. IEEE

Transactions on Antennas and Propagation. 2008;**56**(9):2955-2962. DOI: 10.1109/tap.2008.928801

[17] Zhang WJ, Li L, Li F. Reducing the number of elements in linear and planar antenna arrays with sparseness constrained optimization. IEEE Transactions on Antennas and Propagation. 2011;**59**(8): 3106-3111. DOI: 10.1109/ TAP.2011.2158943

[18] Caratelli D, Vigano MC. A novel deterministic synthesis technique for constrained sparse array design problems. IEEE Transactions on Antennas and Propagation. 2011;**59**(11): 4085-4093. DOI: 10.1109/ TAP.2011.2164193

[19] Viani F, Oliveri G, Massa A. Compressive sensing pattern matching techniques for synthesizing planar sparse arrays. IEEE Transactions on Antennas and Propagation. 2013;**61**(9): 4577-4587. DOI: 10.1109/ TAP.2013.2267195

[20] Oliveri G, Bekele ET, Robol F, Massa A. Sparsening conformal arrays through a versatile BCS based method. IEEE Transactions on Antennas and Propagation. 2014;**62**(4):1681-1689. DOI: 10.1109/tap.2013.2287894

[21] Long-Jun LI, Wang BH, Xia CH. Synthesis of sparse conformal array antennas pattern. Acta Electronica Sinica. 2017;**45**(1):104-111. DOI: 10.3969/j.issn.0372-2112.2017.01.015

[22] Giorgio G, Luca T, Nicola A, Giacomo O, Paolo R. Sparse conformal array design for multiple patterns generation through Multi-Task Bayesian Compressive Sensing. In: Proceedings of 2017 IEEE International Symposium on Antennas and Propagation & USNC/ URSI National Radio Science Meeting; 9-14 July 2017; San Diego, CA, USA: IEEE; 2017. pp. 429-430

[23] Bencivenni C, Ivashina MV, Maaskant R, Wettergren J. Synthesis of maximally sparse arrays using compressive sensing and full-wave analysis for global earth coverage applications. IEEE Transactions on Antennas and Propagation. 2016; **64**(11):4872-4877. DOI: 10.1109/ TAP.2016.2594840

[24] D'Urso M, Prisco G, Tumolo RM. Maximally sparse, steerable, and non super directive array antennas via convex optimizations. IEEE Transactions on Antennas and Propagation. 2016;**64**(9):3840-3849. DOI: 10.1109/TAP.2016.2586490

[25] Tao H, Xiao-Pan S, Xue-Song L. Synthesis of sparse linear array for directional modulation via convex optimization. IEEE Transactions on Antennas and Propagation. 2018;**66**(8): 3959-3972. DOI: 10.1109/ TAP.2018.2835641

[26] Chen H, Shao H-Z, Wang W-Q. Joint sparsity-based range-angle-dependent beampattern synthesis for frequency diverse array. IEEE Access. 2017;**5**(99):15152-15161. DOI: 10.1109/ ACCESS.2017.2731973

[27] Chen H, Wan Q. Non-uniform array pattern synthesis using reweighted L1-norm minimization method. AEUE-International Journal of Electronics and Communications. 2013;**67**(9):795-798. DOI: 10.1016/j.aeue.2013.03.010

[28] Candès EJ, Wakin MB, Boyd SP. Enhancing sparsity by reweighted L1 minimization. Journal of Fourier Analysis and Applications. 2008;**14**(5–6):877-905. DOI: 10.1007/s00041-008-9045-x

[29] Boyd S, Vandenberghe L, Faybusovich L. Convex optimization. IEEE Transactions on Automatic Control. 2006;**51**(11):1859-1859. DOI: 10.1109/TAC.2006.884922

[30] Chen H, Wan Q. Rong fan: Beampattern synthesis using reweighted L1-norm minimization and array orientation diversity. Radioengineering. 2013;**22**(2):602-609

[31] Grant M, Boyd S. CVX: Matlab software for disciplined convex programming, version 2.0 beta. Available from: http://cvxr.com/cvx