Digital Signal Processing 117 (2021) 103152

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Beampattern synthesis for large-scale antenna array via accurate array response control

Weilai Peng, Xuejing Zhang *, Zishu He, Julan Xie, Chunlin Han

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

ARTICLE INFO

Article history:
Available online 29 June 2021

Keywords:
Beampattern synthesis
Large-scale antenna array
Low complexity
Array signal processing

ABSTRACT

In this paper, two low computational complexity accurate array response control algorithms for large-scale antenna array are presented. The first proposed algorithm is an Improved Weight vector ORthogonal Decomposition (I-WORD) approach. We extend the WORD method by selecting the non-negative coefficient as the ultimate solution, and thus the ultimate weight vector will be in an exact form. Therefore, the proposed I-WORD algorithm has no longer a selection procedure and has low computational complexity compared with the WORD algorithm. Moreover, to achieve multi-point accurate array response control, we further develop a Multi-point control based on I-WORD (MI-WORD) algorithm. The MI-WORD algorithm is able to control multi-point response simultaneously by finding the weight vector from the intersection of weight vector sets with a new matrix constructing manner, which is different from the multi-point accurate array response control (MA^2RC) method. Both the proposed algorithms can be applied to adjust the response accurately to synthesize the beampattern. Furthermore, the proposed MI-WORD method has the advantage of low computational complexity, especially when the number of array antennas is large. Simulation results show the effectiveness of the two algorithms and the property of low complexity of the MI-WORD algorithm for beampattern synthesis.

1. Introduction

Array signal processing has attracted significant attention from researchers for decades. An important branch of array signal processing is beampattern synthesis [1–3]. The problem of beampattern synthesis is to design a weight vector for antenna array, which can make the array beampattern meet some specific requirements. For example, the low uniform sidelobe is designed to against the interferences in the radar system. Moreover, antenna arrays with a narrow main beam and high gain are urgently demanded in modern applications. Large phased antenna arrays have been effectively used in realizing highly directional beamforming [4]. Thus, beampattern synthesis for large-scale antenna arrays of which the number of elements is large also has been researched [5].

Plenty of works have been devoted to beampattern synthesis in recent years. A weight vector calculated with the theory of Chebyshev polynomials was presented in [6], which can achieve the same level for all the sidelobes when the beam width is minimum. However, its applications are limited to uniform antenna arrays. As a result, several algorithms have been proposed to solve the beampattern synthesis problem for nonuniform antenna arrays. For example, a simple iterative solution of linearly constrained least squares method was developed in [7]. In [8], the problem of beampattern synthesis was formed as a quadratic program problem, which could synthesize the pattern of arbitrary array to any appropriate beampattern. The element beampattern in an unequally spaced array was considered as the beampattern radiated by a subarray of some equally spaced virtual elements in [9]. The synthesis of arbitrary sparse planar arrays based on the two-dimensional unitary matrix pencil was studied in [10].

Besides, a number of approaches based on global optimization have been developed for nonuniform arrays [11–13]. A genetic algorithm was applied to the beampattern synthesis for thinned arrays in [11]. In [12], the particle swarm optimization was implemented to handle arbitrary nonlinear cost functions. To synthesize beampattern for an unequally spaced array, simulated annealing was applied in [13]. However, all the global search algorithms bring heavy computation loads to the problem of beampattern synthesis.

Different from the global search approaches, a class of methods based on convex optimization techniques [14–16] have been presented in the past several years. In [17], Lebret and Boyd have expressed some antenna array beampattern synthesis problems as

* Corresponding author.
E-mail addresses: lestinpwl@163.com (W. Peng), 784650567@qq.com (X. Zhang), zshe@uestc.edu.cn (Z. He), julanxie@uestc.edu.cn (J. Xie), hancl@uestc.edu.cn (C. Han).

https://doi.org/10.1016/j.dsp.2021.103152
1051-2004/

convex optimization problems, which can be solved by interior-point method. A method based on the second-order cone programming (SOCP) was developed in [18], which was applicable to general beampattern synthesis problem for arbitrary geometry array. In [19], a semidefinite programming has been presented to design the robust array beampattern, which can synthesize beampattern with the uncertainties, for example, array gain uncertainties. In addition, a novel beampattern synthesis method based on the semidefinite relaxation was developed to focus energy in the desired range-angle domain in [20].

Recently, some quite comfortable and effective methods have been researched. The fast Fourier transform (FFT) technique has been applied to synthesize unequally spaced arrays in [9,21]. Then, an extend FFT method as iterative spatiotemporal Fourier transform has been used to design filter coefficients for generating frequency-invariant beam pattern in [22]. The modified iterative FFT technique was also used to synthesize thinned massive array for 5G communications in [23]. Besides, synthesizing the beampattern utilizing the differential evolution method also was researched in [24,25]. Lately, [26] has presented a differential evolution algorithm to achieve shaped power pattern of a linear dipole array. With new encoding mechanism and Cauchy mutation, the differential evolution algorithm was applied to synthesize large unequally spaced planar arrays in [27].

Unfortunately, all algorithms mentioned above cannot flexibly control the array response. It has to be completely redesigned the weight vector even if only a slight change of the desired pattern is needed. When the response at a given direction is needed to be adjusted accurately, the weight vector needs to be resolved from the beginning. To solve this problem, a scheme called optimal and precise array response control (OPARC) which assigns a virtual interference to a direction to control the response level precisely has been investigated in [28]. It can only control the response at one angle in one step. In addition, a beampattern synthesis method based on weight vector orthogonal decomposition (WORD) was proposed in [29], which can control the array response accurately at a given direction. However, there are two weight vectors obtained by the WORD algorithm, which need a criterion to choose one of them as the ultimate weight vector. What's more, both the WORD method and the OPARC method work as point-by-point manner when beampattern is synthesized, which adjust the response at only one given direction in each iteration. Although a multi-point accurate array response control (MA^{2}RC) method based on the accurate array response control (A^{2}RC) method has been presented in [30] and a multi-point method based on oblique projection (OBPJ) has been presented in [31], the computational complexity of them is high when the number of elements and the number of the directions needed to be controlled are large.

Considering the drawbacks afore-mentioned, this paper is dedicated to two low computational complexity and accurate array response control algorithms. Firstly, the two weight vectors obtained by the WORD algorithm will be analyzed and the nonpositive one of the solutions would be discarded. Thus, the ultimate weight vector will be in an exact form and we present an Improved WORD (I-WORD) algorithm which can omit the selection procedure compared with the WORD algorithm. Then, combining the MA^{2}RC method, a multi-point response control at one step method based on the I-WORD algorithm is under consideration. To achieve low computational complexity for large-scale antenna array, a new matrix constructing manner leading to low computational complexity is applied, which is different from the MA^{2}RC method in [30]. Therefore, we obtain a low complexity Multi-point accurate array response control based on I-WORD (MI-WORD) algorithm. Finally, the proposed MI-WORD algorithm is applied to synthesize beampattern for antenna arrays.

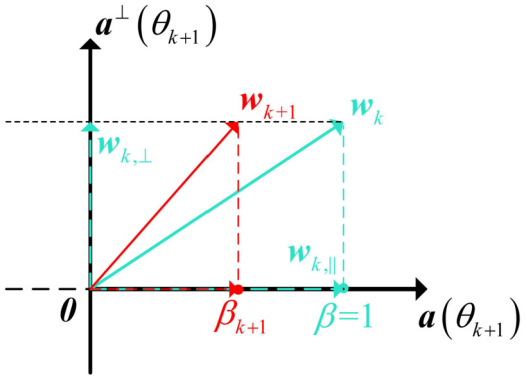

Fig. 1. Illustration of the weight vector orthogonal decomposition.

The rest of this paper is organized as follows. In Section 2, the beampattern synthesis problem is formulated and the WORD algorithm is introduced. The improved WORD algorithm is proposed in Section 3. Then, the proposed MI-WORD method is developed in Section 4. In Section 5, the application of the MI-WORD method for beampattern synthesis is described. Numerical examples are presented in Section 6. Finally, conclusions are drawn in Section 7.

2. Preliminaries

2.1. Formulation of beampattern synthesis problem

Considering an arbitrary geometry array with N elements, the steering vector in direction θ can be expressed as

$$\mathbf{a}(\theta) = \left[f_1(\theta)e^{-j\phi_1(\theta)}, \cdots, f_N(\theta)e^{-j\phi_N(\theta)} \right]^{\mathrm{T}} \tag{1}$$

where $f_n(\theta)$ denotes the element pattern, $j = \sqrt{-1}$ is the imaginary unit, $(\cdot)^{\mathrm{T}}$ is the transpose operation and $\phi_n(\theta)$ represents the phase delay between the nth element and the reference element. Generally, we define the complex weight vector for the array as $\mathbf{w} = [w_1, w_2, \cdots, w_N]^{\mathrm{T}}$. Thus, the far-field array response can be given as

$$P(\theta) = \left| \sum_{n=1}^{N} w_n^* f_n(\theta) e^{-j\phi_n(\theta)} \right| = |\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta)| \tag{2}$$

where $(\cdot)^*$ and $(\cdot)^{\mathrm{H}}$ represent the conjugate and the conjugate transpose operation, respectively.

Let us consider $P_d(\theta)$ as the desired array response. Then, the general beampattern synthesis problem can be stated as finding a weight vector $\mathbf{w}$ to make the synthesized array response meet the desired array response, i.e., $P(\theta) \approx P_d(\theta)$, for all $\theta \in [-90°, 90°]$. Therefore, the major problem is how to design the weights for the antenna array.

2.2. WORD algorithm

For self-contained, in this subsection, we briefly review the WORD scheme presented in [29]. The main idea of the WORD algorithm is that the weight vector is updated based on the vector orthogonal decomposition, shown in Fig. 1. In Fig. 1, $\mathbf{a}(\theta_{k+1})$ is the steering vector corresponding to the direction that will be controlled in the $k+1$ step and $\mathbf{a}^{\perp}(\theta_{k+1})$ is the orthogonal vector of $\mathbf{a}(\theta_{k+1})$. It can be seen that the weight vector $\mathbf{w}_{k+1}$ is a linear combination of two orthogonal vectors $\mathbf{w}_{k,\perp}$ and $\mathbf{w}_{k,\parallel}$ with the coefficients 1 and β_{k+1}, respectively. Note that k means the previous step in which the response at θ_k has been controlled and $k+1$ denotes the current step in which the response at θ_{k+1} will be

adjusted. In addition, the value of k herein is fixed when the algorithm is presented. No iteration is needed to find $\mathbf{w}_{k+1}$ for a given k. In [29], $\mathbf{w}_{k,\perp}$ and $\mathbf{w}_{k,\parallel}$ are defined as

$$\mathbf{w}_{k,\perp} = \mathbf{P}^{\perp}_{[\mathbf{a}(\theta_{k+1})]}\mathbf{w}_k, \quad \mathbf{w}_{k,\parallel} = \mathbf{P}_{[\mathbf{a}(\theta_{k+1})]}\mathbf{w}_k \tag{3}$$

where $\mathbf{P}_{[\mathbf{a}(\theta_{k+1})]}$ and $\mathbf{P}^{\perp}_{[\mathbf{a}(\theta_{k+1})]}$ are the orthogonal projection and the orthogonal complementary projection matrix onto the column space of $\mathbf{a}(\theta_{k+1})$, respectively.

For a given $\mathbf{w}_k$, the weight vector can be updated as

$$\mathbf{w}_{k+1} = [\mathbf{w}_{k,\perp} \quad \mathbf{w}_{k,\parallel}][1 \quad \beta_{k+1}]^{\mathrm{T}}, \tag{4}$$

where β_{k+1} has two solutions β_a and β_b according to the WORD method.

Then, one of the solutions β_a or β_b which one can minimize the variation function $F(\beta) = \|\mathbf{P}^{\perp}_{[\mathbf{w}_{(k)}]}\mathbf{w}_{k+1}/\|\mathbf{w}_{k+1}\|_2\|_2^2$ will be selected. It is noticed that the appropriate β is determined by a selection in the WORD approach. This selection procedure motivates us to develop an improved WORD algorithm, which has a definite solution and does not need to do the selection anymore.

3. The proposed improved WORD algorithm

In this section, we dedicate to present an improved WORD algorithm which does not need to select ultimate β from β_a and β_b but calculates the weight vector with an exact form.

For clarity, two solutions of the β_{k+1} are written below as

$$\beta_{k+1,a} = \frac{-\mathrm{Re}(\mathbf{B}(1,2)) + d}{\mathbf{B}(2,2)}, \beta_{k+1,b} = \frac{-\mathrm{Re}(\mathbf{B}(1,2)) - d}{\mathbf{B}(2,2)} \tag{5}$$

where $d = \sqrt{[\mathrm{Re}(\mathbf{B}(1,2))]^2 - \mathbf{B}(1,1)\mathbf{B}(2,2)}$, $\mathrm{Re}(\cdot)$ returns the real part of a complex number, and $\mathbf{B}$ is given by

$$\mathbf{B} = \begin{bmatrix} -\rho_{k+1}|\mathbf{w}_{k,\perp}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2 & -\rho_{k+1}\mathbf{w}_{k,\perp}^{\mathrm{H}}\mathbf{a}(\theta_0)\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}_{k,\parallel} \\ -\rho_{k+1}\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_0)\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}_{k,\perp} & |\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_{k+1})|^2 - \rho_{k+1}|\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2 \end{bmatrix}, \tag{6}$$

where $\rho_{k+1} = |P(\theta_{k+1})|^2/|P(\theta_0)|^2$ denotes the desired (normalized) response level at θ_{k+1} with main beam axis θ_0.

Then, these two solutions, i.e., β_a and β_b will be analyzed. To begin with, $\mathbf{B}(1,1)$ and $\mathbf{B}(2,2)$ in (5) will be analyzed. It is assumed that $0 \le \rho_{k+1} \le 1$. Thus, we have

$$\mathbf{B}(1,1) = -\rho_{k+1}|\mathbf{w}_{k,\perp}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2 \le 0. \tag{7}$$

Considering $\mathbf{w}_{k,\parallel}$ in (3), we can deduce that

$$\frac{|\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_{k+1})|^2}{\rho_{k+1}|\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2} = \frac{|\mathbf{a}^{\mathrm{H}}(\theta_{k+1})\mathbf{a}(\theta_{k+1})|^2}{\rho_{k+1}|\mathbf{a}^{\mathrm{H}}(\theta_{k+1})\mathbf{a}(\theta_0)|^2}. \tag{8}$$

Recalling that $0 \le \rho_{k+1} \le 1$ and $\mathbf{a}^{\mathrm{H}}(\theta_{k+1})\mathbf{a}(\theta_{k+1}) > \mathbf{a}^{\mathrm{H}}(\theta_{k+1})\mathbf{a}(\theta_0)$, (8) can be further derived that $\frac{|\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_{k+1})|^2}{\rho_{k+1}|\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2} \ge 1$. Thus, we can get

$$\mathbf{B}(2,2) = |\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_{k+1})|^2 - \rho_{k+1}|\mathbf{w}_{k,\parallel}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2 \ge 0. \tag{9}$$

Then, combining (7) and (9), we can obtain

$$\mathbf{B}(1,1)\mathbf{B}(2,2) \le 0. \tag{10}$$

So, it is obvious that

$$d = \sqrt{[\mathrm{Re}(\mathbf{B}(1,2))]^2 - \mathbf{B}(1,1)\mathbf{B}(2,2)} \ge |\mathrm{Re}(\mathbf{B}(1,2))|. \tag{11}$$

Therefore, we can obtain $-\mathrm{Re}(\mathbf{B}(1,2)) + d \ge 0$ and $-\mathrm{Re}(\mathbf{B}(1,2)) - d \le 0$. As we have $\mathbf{B}(2,2) \ge 0$, it can be deduced that

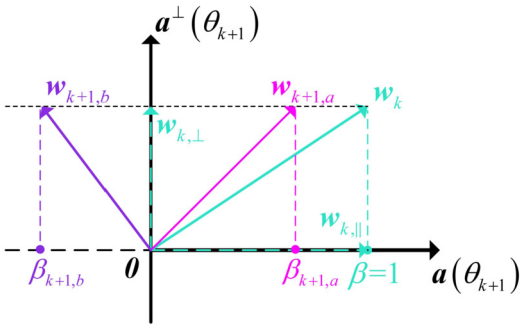

Fig. 2. Illustration of weight vectors with $\beta_{k+1,a} \ge 0$ and $\beta_{k+1,b} \le 0$.

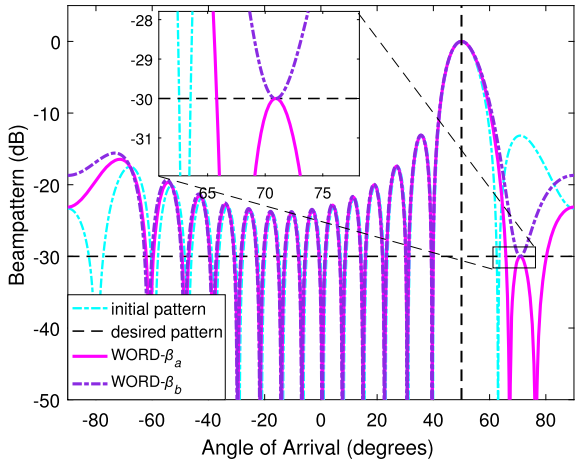

Fig. 3. The beampatterns synthesized with $\beta_{k+1,a}$ and $\beta_{k+1,b}$. A ULA with $N = 16$ and $\theta_0 = 50°$. Adjust the array response at $\theta_{k+1} = 70.9°$ to -30 dB as $\mathbf{w}_k = \mathbf{a}(\theta_0)$.

$$\beta_{k+1,a} \ge 0 \ge \beta_{k+1,b}. \tag{12}$$

In fact, the weight vector in (4) can be expressed as

$$\mathbf{w}_{k+1} = \mathbf{w}_{k,\perp} + \beta_{k+1}\mathbf{w}_{k,\parallel}. \tag{13}$$

It shows that β_{k+1} is a weighting factor on $\mathbf{w}_{k,\parallel}$ in $\mathbf{w}_{k+1}$. As is known, $\mathbf{w}_{k,\parallel}$ is the parallel component of $\mathbf{w}_k$. Therefore, β_{k+1} actually controls the similarity between $\mathbf{w}_{k+1}$ and $\mathbf{w}_k$, which provides similar functionality as shown by variation function $F(\beta)$. To minimize the variation of the beampatterns synthesized by $\mathbf{w}_{k+1}$ and $\mathbf{w}_k$, the similarity of $\mathbf{w}_{k+1}$ and $\mathbf{w}_k$ should be kept as high as possible.

As $\beta_{k+1,b}$ is nonpositive, it will make the parallel components of $\mathbf{w}_{k+1}$ and $\mathbf{w}_k$ be in the approximate opposite direction, resulting that the direction of the weight vector $\mathbf{w}_{k+1}$ will be reversed during the update process. Fortunately, as $\beta_{k+1,a} \ge 0$, the parallel components of weight vectors in adjacent iteration are always in the approximate same direction. For further explanation, the weight vectors obtained with $\beta_{k+1,a}$ and $\beta_{k+1,b}$, are depicted in Fig. 2. From Fig. 2, as $\beta_{k+1,a} \ge 0 \ge \beta_{k+1,b}$, it can be found that the angle between $\mathbf{w}_{k+1,a}$ and $\mathbf{w}_k$ will be always smaller than that between $\mathbf{w}_{k+1,b}$ and $\mathbf{w}_k$, which demonstrates that the similarity of $\mathbf{w}_{k+1,a}$ and $\mathbf{w}_k$ is much higher than that of $\mathbf{w}_{k+1,b}$ and $\mathbf{w}_k$.

The perspective of beampattern is also shown in Fig. 3. It can be seen that the array responses at $\theta_{k+1} = 70.9°$ both are accurately adjusted to -30 dB by the weight vector calculated with $\beta_{k+1,a}$ and $\beta_{k+1,b}$. However, the sidelobe region near the controlled direction synthesized with $\beta_{k+1,b}$ is quite different from the initial beampattern. On the contrast, the beampattern synthesized with $\beta_{k+1,a}$ has a good shape which is similar to the initial beampattern.

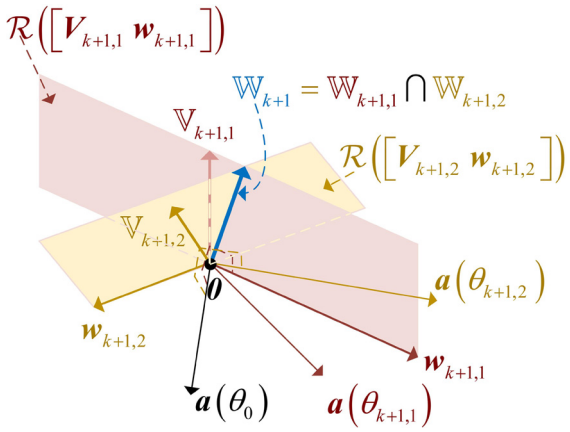

Fig. 4. Illustration of intersection $\mathbb{W}_{k+1} = \mathbb{W}_{k+1,1} \bigcap \mathbb{W}_{k+1,2}$.

According to the above analysis, we can discard $\beta_{k+1,b}$ and let $\beta_{k+1,a}$ be the ultimate solution. Thus, the ultimate weight vector can be computed as

$$\mathbf{w}_{k+1,a} = \mathbf{w}_{k,\perp} + \beta_{k+1,a}\mathbf{w}_{k,\parallel}. \tag{14}$$

Therefore, the ultimate weight vector is an exact form and can be directly calculated by (14) which do not need to make a judgement with variation function $F(\beta)$ anymore. Compared with the WORD scheme, the I-WORD approach can omit the selection procedure without performance loss. Note that the I-WORD algorithm is intended to find one new weight vector $\mathbf{w}_{k+1}$ for a given k with known $\mathbf{w}_k$, which controls one single point array response and is an important step in the later proposed method.

4. The proposed low complexity multi-point control method

In the preceding section, we have presented the I-WORD algorithm, which adjusts the array response at a single direction in each step. In this section, a low complexity Multi-point accurate array response control based on I-WORD (MI-WORD) method will be developed.

4.1. The proposed low complexity MI-WORD algorithm

In [30], a MA^{2}RC method which can achieve multi-point accurate array response control has been presented. It makes use of the intersection of several weight vector sets. The formation of two weight vector sets and their intersection are shown in Fig. 4.

According to [30], there are series of vector that can achieve the same response level at $\rho_{k+1,m}$ as $\mathbf{w}_{k+1,m}$. The set of the weight vectors for array response control at a single direction is established as

$$\mathbb{W}_{k+1,m} \equiv \mathcal{R}([\mathbf{V}_{k+1,m}\ \mathbf{w}_{k+1,m}]) \setminus \mathbb{V}_{k+1,m} \tag{15}$$

where $\mathbf{V}_{k+1}$ is a full column rank matrix whose column space is expressed as $\mathcal{R}^{\perp}(\mathbf{A}(\theta_0, \theta_{k+1,m}))$, $\mathbb{V}_{k+1,m}$ is defined as $\mathbb{V}_{k+1,m} \equiv \mathcal{R}(\mathbf{V}_{k+1,m})$, where $\mathcal{R}(\cdot)$ and $\mathcal{R}^{\perp}(\cdot)$ give the column space and the orthogonal complement space of a matrix, respectively. It defines $\mathbf{A}(\theta_0, \theta_{k+1,m}) \equiv [\mathbf{a}(\theta_0)\ \mathbf{a}(\theta_{k+1,m})]$. The weight vector $\mathbf{w}_{k+1,m}$ is obtained by using the I-WORD method for the $(k+1)$th step, i.e.,

$$\mathbf{w}_{k+1,m} = \mathbf{w}_{k,\perp} + \beta_{k+1,m}\mathbf{w}_{k,\parallel} \tag{16}$$

which can independently adjust the response at a single direction $\theta_{k+1,m}$ to the desired value. Note that, $\mathbf{w}_{k+1,m}$ herein will be calculated by the I-WORD method in this paper, which is different from the method in [30].

Therefore, the unique weight vector controlling M points at one step should lie in the intersection of the M sets. For the sake of easy understanding, $\mathbb{W}_{k+1,1} \bigcap \mathbb{W}_{k+1,2}$ has been shown in Fig. 4.

According to the linear algebra theory [32], the problem of finding a weight vector $\widetilde{\mathbf{w}}_{k+1}$ in $\bigcap_{m=1}^{M} \mathbb{W}_{k+1,m}$, can be converted to obtaining these appropriate vectors $[\mathbf{b}_m^{\mathrm{T}}\ \ c_m] \in \mathbb{C}^{(N-1)}$, for $m = 1, \cdots, M$, satisfying

$$\begin{aligned}\widetilde{\mathbf{w}}_{k+1} &= [\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} = [\mathbf{U}_{22}\ \mathbf{w}_{k+1,2}][\mathbf{b}_2^{\mathrm{T}}\ c_2]^{\mathrm{T}} = \cdots \\ &= [\mathbf{U}_{M2}\ \mathbf{w}_{k+1,M}][\mathbf{b}_M^{\mathrm{T}}\ c_M]^{\mathrm{T}},\end{aligned} \tag{17}$$

where $\mathbf{U}_{m2}$ is composed of the last $N-2$ columns of the unitary matrix which is obtained from the SVD operation on $\mathbf{A}(\theta_0, \theta_{k+1,m})$.

Different from the MA^{2}RC method in [30], we will present a new matrix construction manner to solve (17), resulting in a low computational complexity algorithm in this paper. The new scheme will be shown as following.

Considering $c_m \neq 0$ and $m = 2, \cdots, M$, the equation (17) is equivalent to

$$[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} = [\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}][\mathbf{b}_m^{\mathrm{T}}\ c_m]^{\mathrm{T}}. \tag{18}$$

According to the linear algebra theory, we know

$$\mathcal{R}([\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}]) = \mathcal{N}^{\perp}([\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}]^{\mathrm{T}}) \tag{19}$$

where $\mathcal{N}(\cdot)$ returns the null space of matrix. Hence, we have

$$[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} \in \mathcal{N}^{\perp}([\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}]^{\mathrm{T}}). \tag{20}$$

Then, the problem in (17) can be stated as

$$\begin{aligned}&\text{find} \quad [\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} \in \mathbb{C}^{(N-1)} \\ &\text{s.t.} \quad [\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} \in \mathcal{N}^{\perp}([\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}]^{\mathrm{T}}) \\ &\qquad \text{for} \quad m = 2, \cdots, M \\ &\qquad c_1 \neq 0.\end{aligned} \tag{21}$$

Proceeding, the constraint in (21) should be analyzed. In order to express $\mathcal{N}^{\perp}([\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}]^{\mathrm{T}})$, the SVD operation on $[\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}]$ can be applied. Let $\tilde{\mathbf{U}}_m \in \mathbb{C}^{N\times N}$ be its unitary matrix composed of the left-singular vectors. Thus, $\tilde{\mathbf{U}}_m$ is a matrix of full column rank and it can be divided into two submatrices composed with the first $N-1$ columns and the last column, respectively. It can be described as

$$\tilde{\mathbf{U}}_m = [\underbrace{\tilde{\mathbf{u}}_1 \cdots \tilde{\mathbf{u}}_{N-1}}_{\mathbf{D}_m}\ \underbrace{\tilde{\mathbf{u}}_N}_{\mathbf{d}_m}]. \tag{22}$$

According to the SVD theory, it is known that $\mathbf{d}_m$ can be the basis vector of $\mathcal{N}([\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}]^{\mathrm{T}})$. Combining with (20), we find that $\mathbf{d}_m$ is orthogonal to $[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}}$, which can be expressed as

$$\mathbf{d}_m^{\mathrm{H}}[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} = 0. \tag{23}$$

Therefore, the problem of (21) can be formulated as

$$\begin{aligned}&\text{find} \quad [\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} \in \mathbb{C}^{(N-1)} \\ &\text{s.t.} \quad \mathbf{d}_m^{\mathrm{H}}[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} = 0 \\ &\qquad \text{for} \quad m = 2, \cdots, M \\ &\qquad c_1 \neq 0.\end{aligned} \tag{24}$$

For simplicity, let's define

$$\begin{aligned}&\mathbf{g}_m^{\mathrm{T}} \equiv \mathbf{d}_m^{\mathrm{H}}[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}] \in \mathbb{C}^{1\times(N-1)}\\&\mathbf{G} \equiv [\mathbf{g}_2\ \mathbf{g}_3\ \cdots\ \mathbf{g}_M]^{\mathrm{T}} \in \mathbb{C}^{(M-1)\times(N-1)}\end{aligned} \tag{25}$$

where $m = 2, \cdots, M$. Thus, (24) can be simplified as

$$\begin{aligned}&\text{find}\quad [\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} \in \mathbb{C}^{(N-1)}\\&\text{s.t.}\quad \mathbf{G}[\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} = \mathbf{0}\\&\qquad c_1 \neq 0.\end{aligned} \tag{26}$$

Then, to be more easier to solve (26), those constraints will be more simplified. Considering $c_1 \neq 0$, we have $\mathbf{G}[\tilde{\mathbf{b}}_1^{\mathrm{T}}\ 1]^{\mathrm{T}} = \mathbf{0}$ where $\tilde{\mathbf{b}}_1^{\mathrm{T}} = \mathbf{b}_1^{\mathrm{T}}/c_1$. Hence, the problem of (26) is equivalent to the optimization problem stated as

$$\begin{aligned}&\text{find}\quad \tilde{\mathbf{b}}_1^{\mathrm{T}} \in \mathbb{C}^{(N-2)}\\&\text{s.t.}\quad \mathbf{G}[\tilde{\mathbf{b}}_1^{\mathrm{T}}\ 1]^{\mathrm{T}} = \mathbf{0}.\end{aligned} \tag{27}$$

To obtain the analytical solution to (27), the matrix $\mathbf{G} \in \mathbb{C}^{(M-1)\times(N-1)}$ is partitioned into block matrices, expressed as

$$\mathbf{G} = [\mathbf{T}\ \mathbf{t}] \tag{28}$$

where $\mathbf{T} \in \mathbb{C}^{(M-1)\times(N-2)}$ is composed with the first $N-2$ columns and $\mathbf{t} \in \mathbb{C}^{(M-1)}$ represents the last column of $\mathbf{G}$. They can be given by

$$\mathbf{T} = \begin{bmatrix}\mathbf{d}_2^{\mathrm{H}}\mathbf{U}_{12}\\ \mathbf{d}_3^{\mathrm{H}}\mathbf{U}_{12}\\ \vdots\\ \mathbf{d}_M^{\mathrm{H}}\mathbf{U}_{12}\end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix}\mathbf{d}_2^{\mathrm{H}}\mathbf{w}_{k+1,1}\\ \mathbf{d}_3^{\mathrm{H}}\mathbf{w}_{k+1,1}\\ \vdots\\ \mathbf{d}_M^{\mathrm{H}}\mathbf{w}_{k+1,1}\end{bmatrix}. \tag{29}$$

Substituting (28) into the constraint condition in (27), we can obtain the general solution to the optimization problem as

$$\tilde{\mathbf{b}}_1^{\mathrm{T}} = -\mathbf{T}^{\dagger}\mathbf{t} + \mathbf{z},\ \forall \mathbf{z} \in \mathcal{N}(\mathbf{T}), \tag{30}$$

where † is the pseudo-inverse operation.

On account of the equivalence between (26) and (27), the solution to (26) can be obtained as

$$[\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}} = c_1\begin{bmatrix}-\mathbf{T}^{\dagger}\mathbf{t} + \mathbf{z}\\ 1\end{bmatrix}, c_1 \neq 0, \forall \mathbf{z} \in \mathcal{N}(\mathbf{T}). \tag{31}$$

Thus, the weight vector $\widetilde{\mathbf{w}}_{k+1}$ in (17) can be expressed as

$$\begin{aligned}\widetilde{\mathbf{w}}_{k+1} &= [\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\mathbf{b}_1^{\mathrm{T}}\ c_1]^{\mathrm{T}}\\ &= c_1[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}]\begin{bmatrix}-\mathbf{T}^{\dagger}\mathbf{t} + \mathbf{z}\\ 1\end{bmatrix}, c_1 \neq 0, \forall \mathbf{z} \in \mathcal{N}(\mathbf{T})\end{aligned} \tag{32}$$

which can adjust responses at all M directions to the desired level at one step. It's remarkable that the number of the picked directions cannot exceed $N-1$ and it must be guaranteed that $\mathbf{a}(\theta_0), \mathbf{a}(\theta_{k+1,1}), \cdots, \mathbf{a}(\theta_{k+1,M})$ are linearly independent to ensure equation (17) solvable.

Remark: The proposed MI-WORD algorithm can impose a derivative constraint on the direction of beam center to avoid the shifting of the main beam axis, which is similar to M^{2}A^{2}RC algorithm in [30]. As it's not the major work in this paper, and for brevity, the final weight vector which has been considered the beam shifting can be directly expressed as

$$\widehat{\mathbf{w}}_{k+1} = c_1[\boldsymbol{\Xi}\ \mathbf{w}_{k+1,1}][\mathbf{C}^{\dagger}\mathbf{k} + \mathbf{c}_n\ 1]^{\mathrm{T}},\ c_1 \neq 0,\ \forall \mathbf{c}_n \in \mathcal{N}(\mathbf{C}), \tag{33}$$

of which the details of derivation can be seen in the Appendix A.

4.2. Computational complexity analysis

From MA^{2}RC algorithm in [30] and the proposed MI-WORD algorithm in the preceding subsection, we find that solving the weight vector $\widetilde{\mathbf{w}}_{k+1}$ occupies the major complexity, especially calculating $\mathbf{T}^{\dagger}\mathbf{t}$. Thus, we only focus on the computational complexity of $\mathbf{T}^{\dagger}\mathbf{t}$.

As afore-mentioned, in MI-WORD algorithm, $\mathbf{T} \in \mathbb{C}^{(M-1)\times(N-2)}$ and $\mathbf{t} \in \mathbb{C}^{(M-1)}$, so the computational complexity of $\mathbf{T}^{\dagger}\mathbf{t}$ is $\mathcal{O}(2(M-1)(N-2)^2 + (N-2)^3 + (M-1)(N-2))$. However, compared with MA^{2}RC algorithm, an additional SVD operation must be taken on $[\mathbf{U}_{m2}\ \mathbf{w}_{k+1,m}] \in \mathbb{C}^{N\times(N-1)}$, which has $\mathcal{O}(2N(N-1)^2)$ computational complexity. Totally, the computational complexity in the proposed approach is $\mathcal{O}(2(M-1)(N-2)^2 + (N-2)^3 + (M-1)(N-2) + 2N(N-1)^2)$. In fact, the number of the directions needed to be controlled is always smaller than that of the elements, i.e., $M < N$ according to the *Remark 1* in [30]. Hence, the computational complexity can be approximately to $\mathcal{O}(2N(N-1)^2)$ when N is large. With the same observation, the complexity of MA^{2}RC algorithm can be approximately to $\mathcal{O}(2N(M-1)(N-2)^2)$. The complexity of the OBPJ algorithm in [31] is $\mathcal{O}(M^2N^2)$. It is noteworthy that M beam peak angles where the response differences (from the desired levels) are relatively large will be chosen for sidelobe synthesis and we will make the value of M get as large as possible to reduce iteration when the beampattern is synthesized. For a large N, M will become large too. According to the above analysis, we can find that the proposed MI-WORD approach has lower computational complexity than the MA^{2}RC method, especially for large-scale antenna array. When the array elements num N and the number of directions needed to be controlled M are large enough, the computational complexity of the proposed MI-WORD method will be smaller than that of the OBPJ method. Thanks to the new matrix construction manner, the MI-WORD algorithm has low complexity compared with the MA^{2}RC algorithm and the OBPJ method.

5. Beampattern synthesis using the proposed MI-WORD algorithm

In the above sections, the I-WORD and MI-WORD algorithms have been presented. Following, the application of the proposed MI-WORD algorithm to beampattern synthesis will be introduced. To achieve synthesis, the MI-WORD algorithm is iterative to control multi-point response, where the I-WORD algorithm obtains the weight vector for each point by a single step.

To be specific, we set $\mathbf{a}(\theta_0)$ as the initial weight vector $\overline{\mathbf{w}}_0$ and obtain the beampattern $L_0(\theta)$. Let $L_d(\theta)$ denote the desired beampattern. After setting $k = 0$, the iterative process will be started. In the $(k+1)$th step, M directions needed to be adjusted should be determined, according to the previous beampattern $L_k(\theta)$ and the desired beampattern $L_d(\theta)$. M beam peak angles where the response differences (from the desired levels) are relatively large will be chosen for sidelobe synthesis and we will make the value of M get as large as possible to reduce iteration. More details about the direction selection strategy can be found in [30] and [31].

Next, the desired normalized array response level $\rho_{k+1,m}$ at direction $\theta_{k+1,m}$ should be figured out. Therefore, the corresponding weight vector $\mathbf{w}_{k+1,m}$ is calculated by using the I-WORD algorithm with (16). Then, the weight vector $\widetilde{\mathbf{w}}_{k+1}$ which controls M angles response can be computed by (32). Thus, the corresponding normalized array response can be obtained. Then, check whether the beampattern is satisfactorily synthesized. If not, set $k = k+1$ and repeat the above steps until the synthesized beampattern meets the requirements of the desired beampattern. Otherwise, output the ultimate weight vector $\widetilde{\mathbf{w}}_{k+1}$ and synthesized beampattern. The above steps are summarized in Table 1.

Table 1
Summary of MI-WORD method for beampattern synthesis.

Input	$k=0$, θ_0, $\widetilde{\mathbf{w}}_0=\mathbf{a}(\theta_0)$, $L_0(\theta)$, $L_d(\theta)$
Step 1.	Determine M_{k+1} angles $\theta_{k+1,m}$ by comparing $L_k(\theta)$ with $L_d(\theta)$ and the corresponding steering vector $\mathbf{a}(\theta_{k+1,m})$, for $m=1,\cdots,M_{k+1}$.
Step 2.	Figure out the desired response level $\rho_{k+1,m}$ at $\theta_{k+1,m}$. Determine $\mathbf{B}$ in (6) and calculate $\beta_{k+1,a}$ in (5). Then, calculate $\mathbf{w}_{k+1,m}$ with (16). Repeat this step until find all M_{k+1} weight vectors via the I-WORD method.
Step 3.	Compute $\widetilde{\mathbf{w}}_{k+1}$ by (32) and the normalized response can be obtained as $L_{k+1}(\theta)=\lvert\widetilde{\mathbf{w}}_{k+1}^{\mathrm{H}}\mathbf{a}(\theta)\rvert^2/\lvert\widetilde{\mathbf{w}}_{k+1}^{\mathrm{H}}\mathbf{a}(\theta_0)\rvert^2$.
Step 4.	Setting $k=k+1$, go to **Step 1** unless the normalized beampattern $L_{k+1}(\theta)$ satisfies $L_d(\theta)$.
Output	$\widetilde{\mathbf{w}}_{k+1}$ and the final normalized beampattern $L_{k+1}(\theta)$.

6. Simulations

In this section, the proposed I-WORD algorithm will be firstly simulated. Then, the effectiveness and the property of low computational complexity of the proposed MI-WORD algorithm will be illustrated with several examples. For comparison, the convex-optimization-based method (labelled as "convex method" following) in [17] and the MA^{2}RC method are simulated if available. Note that all the examples about the running time of the method are simulated on a computer with Intel Core CPU i7-10700 at 2.9 GHz.

6.1. Illustration of the proposed I-WORD algorithm

To illustrate the proposed I-WORD algorithm, a linear array with $N=60$ nonuniformly spaced elements is considered in this subsection. The element locations are given in Table 2. The desired sidelobe response is -30 dB with $\theta_0=-30^\circ$. For comparison, 100 iterations will be taken with the weight vector calculated by β_a and β_b, respectively.

The comparison results are shown in Fig. 5. In Fig. 5(a), the first sidelobe peak on initial beampattern is chosen to be controlled and their responses on both beampattern synthesized with β_a and β_b are adjusted to -30 dB. The beampattern synthesized with β_a still keep it as the first sidelobe while it is changed in the beampattern synthesized with β_b. The beampattern variation resulting from β_b is larger than that from β_a. From Fig. 5(b) and Fig. 5(c), we can see that both the weights obtained with β_a and β_b can accurately adjust response level to the desired level. However, the response levels in the region near -85° and 85°, marked by an ellipse in these two figures, rise up alternately on the beampattern obtained with β_b. Contrasted with that, the beampattern obtained with β_a shows a good downward iteration trend. Moreover, the beampattern synthesized by β_a almost satisfies the desired pattern at the 42th step. The resulting weights obtained with β_a (with 100 iteration steps) are specified in Table 2.

To give a perspective of iteration, we define the max difference sidelobe level between the resulting beampattern and the desired beampattern as $\mathrm{MDL}=\max\{L(\theta)-L_d(\theta)|\theta\in\Omega_s\}$, where Ω_s denotes the sidelobe region. As shown in Fig. 6, the MDL obtained with β_a can converge faster than that obtained with β_b, which indicates that β_a has higher iteration efficiency than β_b.

6.2. Uniform sidelobe beampattern synthesis

In this subsection, the effectiveness of the proposed MI-WORD algorithm will be firstly illustrated. An uniform linear array (ULA) with $N=60$ elements will be considered. To avoid grating lobes, the array element spacing is set as half wavelength. The beam center is $\theta_0=20^\circ$. The desired uniform sidelobe level is designed to be lower than -40 dB. The results of beampattern synthesis by using the proposed MI-WORD approach are shown in Fig. 7.

In Fig. 7(a), there are $M=58$ peak sidelobes, which are selected from the beampattern formed with the quiescent weight vector $\widetilde{\mathbf{w}}_0=\mathbf{a}(\theta_0)$. At the first step iteration, a weight vector is calculated by low complexity MI-WORD algorithm, which can control the response levels of those directions. As we can see, the response levels of those selected directions have been adjusted to -40 dB accurately. The proposed algorithm has ability to adjust the response levels of multi-directions to the desired levels at one step.

Next, at the second iteration, those sidelobe peaks around main-lobe, whose response levels still much higher than desired values, are selected as other sidelobe peaks. After this step iteration, the whole sidelobe peaks values in current beampattern are adjusted to nearby -40 dB, shown in Fig. 7(b).

Then, in Fig. 7(c), we can see that the sidelobe peaks values have been further adjusted to the desired values in the third step. At this step, the current beampattern almost satisfies the desired beampattern. At last, the output beampattern is shown in Fig. 7(d). We can find that the sidelobe level of the convex method is higher than that of the proposed method, though they are all closed to the desired level. Besides, the proposed method performs as at least good as the state-of-the-art MA^{2}RC approach.

The process of beampattern synthesis illustrates that the proposed algorithm can control multi-point response levels accurately. It also demonstrates that to make the synthesized beampattern satisfy the desired beampattern, only few steps should be taken by the proposed method.

To show the property of low complexity of the proposed algorithm, an example contrasting the running time with different array elements has been simulated among the different methods. In this simulation, to have enough sidelobe directions to be controlled on the beampattern of large scale array, the desired uniform sidelobe level is set to -50 dB, specially. The comparisons are shown in Fig. 8. It is obvious that the running time of convex method is much larger than those of the others method. Otherwise, when the array scale is large, the running time of the proposed method is much smaller than that of the MA^{2}RC method. When the array elements num is larger than 200, the running time of the proposed MI-WORD method becomes smaller than that of the OBPJ method. These observations are in accordance with our complexity analysis.

To illustrate that the running time varies with the value of M, we have simulated these mentioned methods with different M and the results are shown in Table 3. We can find that the running time will get large as M becomes large. When the array elements num N is large enough, for example $N\geq 300$, the running time of the proposed MI-WORD method is smallest as the M gets large enough value. When both M and N are large enough, the running time of the proposed MI-WORD method will be the smallest.

6.3. Multibeam beampattern synthesis for nonuniformly spaced linear array

Considering the multi-target tracking, multibeam beampattern synthesis for a nonuniformly spaced array whose locations are same as in Table 2 is considered. The two beams point at -40° and 20°. We herein design $\mathbf{a}(20^\circ)$ as the initial weight. Then, the array response at -40° is adjusted to 0 dB by the I-WORD algorithm. Next, the sidelobe response level will be controlled to be lower than -30 dB via the MI-WORD algorithm. Finally, we obtain a satisfied beampattern with the weights shown in Table 4. The beampatterns achieved by the proposed method and other methods are plotted in Fig. 9. It can be seen that the shapes of these three beampattern satisfy the desired pattern, while the corresponding response level for convex method at 24.5° is about 0.01 dB higher than the desired value. Under such simulation param-

Table 2
Element locations of nonuniformly spaced linear array and the weights obtained by I-WORD method with β_a.

n	$x_n(\lambda)$	w_n	n	$x_n(\lambda)$	w_n	n	$x_n(\lambda)$	w_n
1	0.00	$0.53e^{+j0.008}$	21	9.95	$1.10e^{+j0.089}$	41	20.02	$1.06e^{-j0.060}$
2	0.48	$0.51e^{-j1.509}$	22	10.50	$1.10e^{-j1.538}$	42	20.49	$1.06e^{-j1.509}$
3	0.98	$0.37e^{+j3.127}$	23	10.97	$1.22e^{-j3.048}$	43	20.95	$1.00e^{-j3.014}$
4	1.53	$0.34e^{+j1.558}$	24	11.46	$1.18e^{+j1.645}$	44	21.51	$0.92e^{+j1.535}$
5	1.94	$0.36e^{+j0.014}$	25	11.99	$1.19e^{+j0.052}$	45	22.03	$0.94e^{-j0.026}$
6	2.50	$0.37e^{-j1.433}$	26	12.52	$1.30e^{-j1.597}$	46	22.52	$0.84e^{-j1.687}$
7	2.97	$0.51e^{-j3.088}$	27	12.97	$1.26e^{-j3.106}$	47	22.97	$0.82e^{-j3.012}$
8	3.49	$0.53e^{+j1.542}$	28	13.48	$1.24e^{+j1.652}$	48	23.48	$0.75e^{+j1.618}$
9	4.03	$0.55e^{-j0.089}$	29	14.00	$1.32e^{+j0.023}$	49	23.97	$0.67e^{+j0.120}$
10	4.52	$0.62e^{-j1.624}$	30	14.46	$1.27e^{-j1.486}$	50	24.50	$0.72e^{-j1.508}$
11	4.98	$0.65e^{-j3.099}$	31	15.01	$1.30e^{-j3.123}$	51	24.96	$0.55e^{-j3.121}$
12	5.46	$0.71e^{+j1.697}$	32	15.47	$1.30e^{+j1.615}$	52	25.52	$0.55e^{+j1.636}$
13	5.97	$0.74e^{+j0.067}$	33	15.99	$1.22e^{+j0.039}$	53	25.94	$0.58e^{+j0.148}$
14	6.50	$0.80e^{-j1.688}$	34	16.52	$1.31e^{-j1.568}$	54	26.53	$0.44e^{-j1.705}$
15	7.02	$0.89e^{+j3.133}$	35	16.69	$1.24e^{-j3.123}$	55	27.02	$0.40e^{+j3.125}$
16	7.47	$0.92e^{+j1.563}$	36	17.53	$1.18e^{+j1.515}$	56	27.50	$0.36e^{+j1.572}$
17	8.05	$0.89e^{-j0.135}$	37	17.79	$1.22e^{+j0.092}$	57	27.97	$0.32e^{+j0.124}$
18	8.53	$1.05e^{-j1.632}$	38	18.45	$1.18e^{-j1.412}$	58	28.49	$0.41e^{-j1.482}$
19	8.98	$0.99e^{+j3.131}$	39	18.98	$1.55e^{-j3.078}$	59	28.96	$0.46e^{-j3.106}$
20	9.54	$1.03e^{+j1.512}$	40	19.50	$1.10e^{+j1.574}$	60	29.48	$0.54e^{+j1.710}$

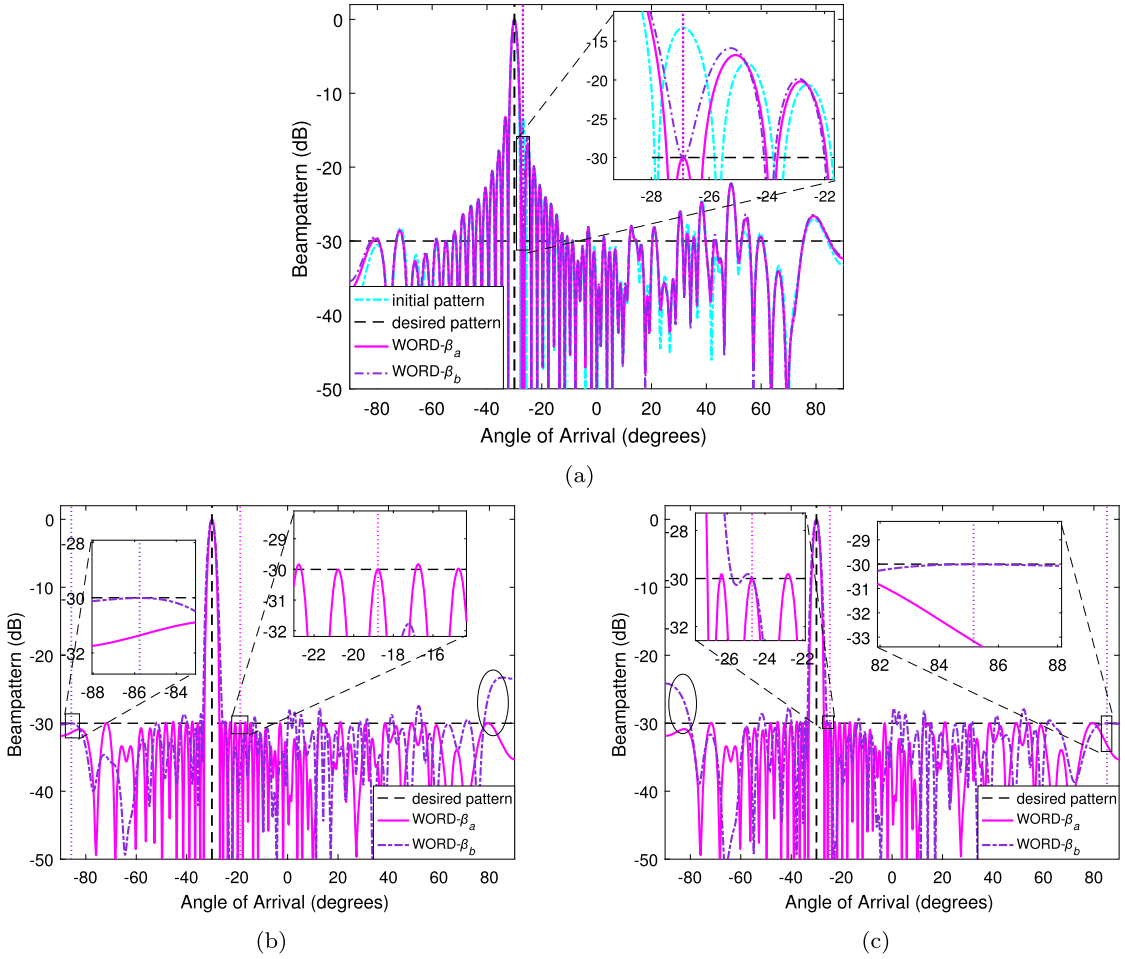

Fig. 5. Beampattern comparison with weights vector calculated by β_a and β_b. Beampatterns at (a) the 1th step, (b) the 41th step, (c) the 42th step.

eters setting, the proposed algorithm takes the shortest time to obtain the satisfied beampattern.

6.4. Beampattern synthesis with nonisotropic elements

A large-scale antenna array with $N = 140$ nonisotropic microstrip patch antennas used for the elevation-scanning will be simulated in this subsection. The E-plane beampattern of the nth microstrip patch antenna is described as

$$f_n(\theta) = \frac{\sin(\pi h_n \sin\theta)}{\pi h_n \sin\theta} \cos(\pi l_n \cos\theta) \tag{34}$$

where h_n is the relative thickness of the dielectric to the wavelength and l_n represents the relative length of the patch of the

Table 3
The running time with the number of beam position.

Time(:s) \ M / Method	$N=80$			$N=100$			$N=300$			$N=600$		
	10	40	78	10	50	98	10	150	298	10	300	598
MA^{2}RC	2	7.1	13.5	2.3	10.3	21.7	8.3	122.8	245.3	17.5	571.4	1145.3
OBPJ	0.2	0.3	0.5	0.3	0.4	0.8	1.4	11.2	29.6	3.6	104.2	327.3
MI-WORD	0.1	0.4	0.7	0.2	0.7	1.3	0.8	11.6	24.9	3.1	121.3	242.4
convex	260.5			279.3			834.1			2192.9		

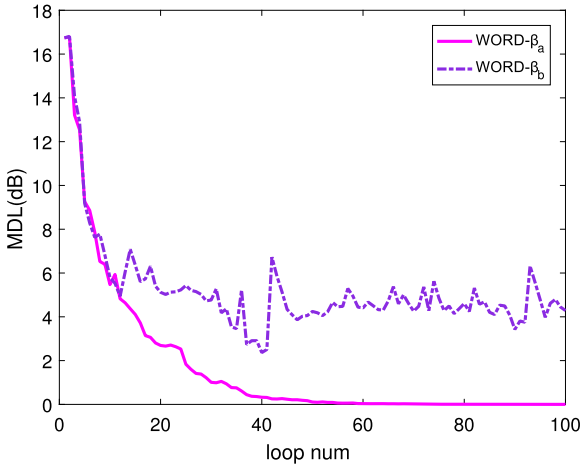

Fig. 6. Max difference sidelobe level with iteration steps.

Table 4
The weights obtained by MI-WORD method for nonuniform array.

n	w_n	n	w_n	n	w_n
1	$0.70e^{+j1.028}$	21	$0.22e^{+j2.788}$	41	$1.29e^{-j1.915}$
2	$1.08e^{+j0.684}$	22	$2.33e^{-j2.499}$	42	$1.90e^{+j0.602}$
3	$0.19e^{-j2.859}$	23	$0.55e^{-j2.566}$	43	$1.22e^{+j0.285}$
4	$0.69e^{+j2.888}$	24	$2.54e^{-j0.289}$	44	$1.67e^{+j2.270}$
5	$0.25e^{-j0.865}$	25	$0.64e^{-j0.269}$	45	$1.06e^{+j2.547}$
6	$0.92e^{-j1.210}$	26	$2.71e^{+j1.626}$	46	$1.54e^{-j1.264}$
7	$0.30e^{+j1.246}$	27	$0.56e^{+j1.895}$	47	$1.16e^{-j1.698}$
8	$0.93e^{+j1.009}$	28	$2.56e^{-j2.219}$	48	$1.20e^{+j0.816}$
9	$0.31e^{-j3.051}$	29	$1.02e^{-j2.332}$	49	$1.08e^{+j0.607}$
10	$1.23e^{-j3.058}$	30	$2.70e^{+j0.050}$	50	$0.91e^{+j3.056}$
11	$0.16e^{-j0.675}$	31	$1.18e^{-j0.289}$	51	$0.98e^{+j2.784}$
12	$1.60e^{-j0.894}$	32	$2.85e^{+j2.211}$	52	$0.83e^{-j1.039}$
13	$0.21e^{+j0.764}$	33	$0.97e^{+j1.962}$	53	$0.84e^{-j1.346}$
14	$1.62e^{+j1.324}$	34	$2.57e^{-j1.960}$	54	$0.71e^{+j1.093}$
15	$0.13e^{+j2.586}$	35	$0.95e^{-j1.983}$	55	$0.74e^{+j0.875}$
16	$2.03e^{-j2.756}$	36	$2.31e^{+j0.281}$	56	$0.41e^{-j2.849}$
17	$0.22e^{-j1.144}$	37	$1.24e^{-j0.025}$	57	$0.68e^{+j2.960}$
18	$2.16e^{-j0.619}$	38	$2.15e^{+j2.480}$	58	$0.51e^{-j0.667}$
19	$0.24e^{+j1.206}$	39	$1.46e^{+j2.221}$	59	$0.98e^{-j1.070}$
20	$2.39e^{+j1.549}$	40	$2.13e^{-j1.612}$	60	$0.69e^{+j1.453}$

antenna to the wavelength. The number of iteration step is set to 5. The results are shown in Fig. 10. The beampatterns of three methods all can satisfy the desired beampattern. But there exist some directions whose response levels on the beampattern obtained by convex method are higher than the desired levels. Moreover, the running time of the proposed low complexity MI-WORD algorithm is 49.2 seconds, which is smaller than 56.1 seconds by MA^{2}RC method and 110.8 seconds by convex method. This demonstrates that the proposed method can be faster than the MA^{2}RC method and the convex method to calculate the ultimate weight vector, though the final beampattern can be very similar.

Table 5
The coefficients between nine adjacent elements.

h_1	1	h_4	0.4039+j0.1563	h_7	0.0903-j0.0825
h_2	0.6278-j0.3974	h_5	0.3045-j0.0963	h_8	0.0473-j0.0412
h_3	0.4943+j0.2659	h_6	0.1278+j0.1470	h_9	0.0086+j0.0037

In order to evaluate the performance of the proposed approach with the finite-resolution of the phase shifters, the phase weightings are quantized in discrete phase with different resolutions. The resulting beampatterns are depicted in Fig. 11. It can be seen that the beampattern with 8-bit resolution is closed to the resulting beampattern with full-resolution. As observed, there exist some sidelobe peaks at some regions on the beampattern with 5-bit resolution which are higher than those on others beampattern. Fortunately, the shape of beampattern with 5-bit resolution still satisfying that of the desired beampattern.

6.5. Flat-top pattern synthesis for a linear array

In this subsection, the proposed MI-WORD algorithm is utilized to synthesize a flat-top pattern to show the effectiveness for synthesizing shaped beam pattern. A 30-element 0.45λ-spaced linear array will be considered. We set the mainlobe region as $[-9°, 9°]$ and the sidelobe region as $[-90°, -15°]$ and $[15°, 90°]$. The response levels in mainlobe are expected to be 0 dB and the desired sidelobe level is below -25 dB. A SDR method [14], which is based on convex method but overcoming the drawback of the lower bound constraint causing non-convex, is tested in this example. The initial beampattern and the synthesized beampatterns are depicted in Fig. 12. It can be seen that all these methods can obtain a satisfactory beampattern. The ripple level of the proposed MI-WORD algorithm is minimal as it is less than 0.15 dB. Moreover, the proposed MI-WORD algorithm needs less iteration steps than the WORD method to complete the synthesis procedure. However, the SDR method needs dozens of seconds, which is much longer than that of the WORD method and the MI-WORD algorithm as a few seconds.

6.6. Beampattern synthesis with mutual coupling

In this subsection, the synthesis of a linear array beampattern in the presence of mutual coupling will be presented. The number of antenna elements in the array is set as $N = 100$. In this example, only nine adjacent elements (containing the current element) on each side of the current element will be considered the mutual coupling effects. It is assuming that the mutual coupling beyond 4λ is negligible according to [33]. The coefficients vector of nine adjacent elements can be set as $\mathbf{h} = [h_1\ h_2\ \cdots\ h_9]$, while the other coefficients are set as 0. The value of $\mathbf{h}$ is given in Table 5. The number of iteration step is set to 8.

The resulting weightings obtained by the proposed approach are listed in Table 6. In Fig. 13, we can find that the proposed algorithm can achieve the desired beampattern perfectly and the

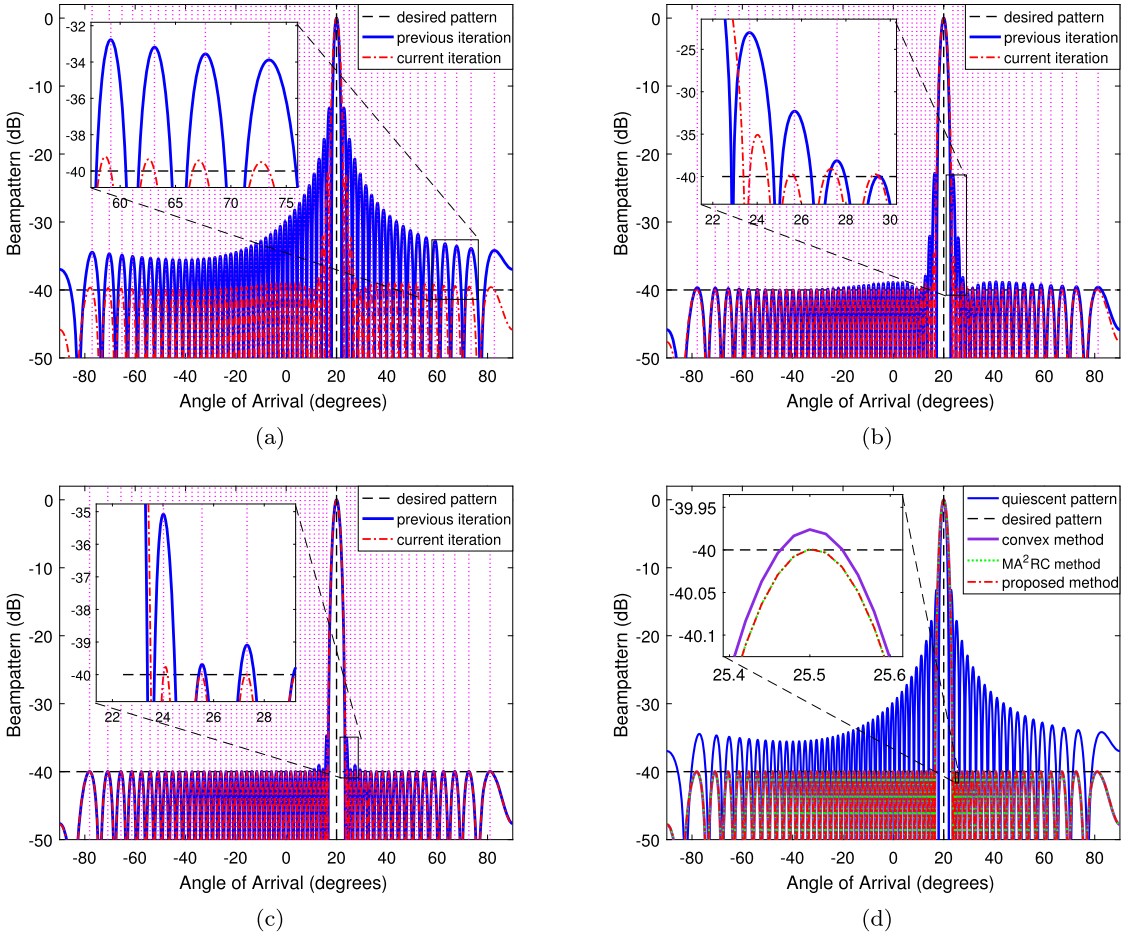

Fig. 7. Simulation results for beampattern synthesis via MI-WORD. Beampatterns at (a) the first step, (b) the second step, (c) the third step. (d) The output beampattern.

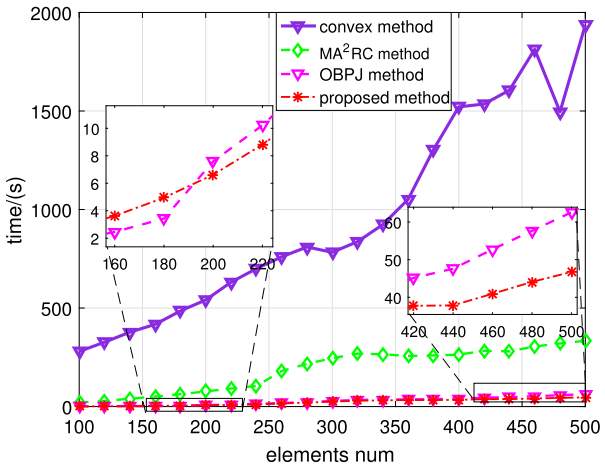

Fig. 8. The curve of running time with the number of array elements.

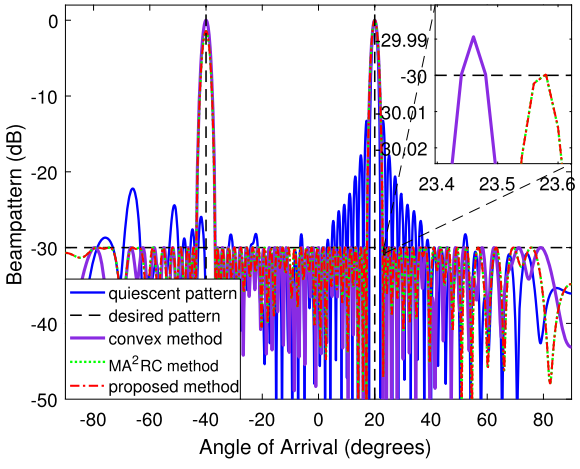

Fig. 9. The resulting beampatterns of nonuniform array.

beampattern of the proposed algorithm has the lowest sidelobe response level among these three methods. About the running time of the program, the proposed algorithm takes 39.7 seconds. On the contrast, it takes 44.9 seconds by the MA^{2}RC method and 95.7 seconds by the convex method. So, the running time of the proposed method is less than those of the MA^{2}RC method and the convex method. Consequently, the effectiveness and the low complexity of the proposed MI-WORD approach contrasted with the MA^{2}RC method can be demonstrated in this example.

6.7. Beampattern synthesis for two-dimensional array

To show the extensive applicability of the proposed algorithm, it is applied to synthesize beampattern for two-dimensional array, which is also a kind of large-scale antenna array in a sense. An uniform planar array composed of 10×10 antennas is considered in this example. We define $u = \sin(\theta_e)\cos(\theta_a)$, $v = \sin(\theta_e)\sin(\theta_a)$ as new variables, where θ_e and θ_a represent elevation and azimuth angles, respectively. Thus, the desired beampattern can be expressed as $L_d(u, v)$ shown in Fig. 14(a). The beam points to

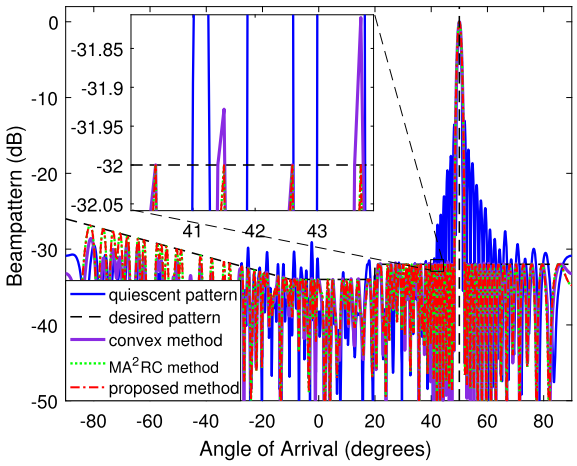

Fig. 10. Nonuniform sidelobe beampatterns with nonisotropic elements.

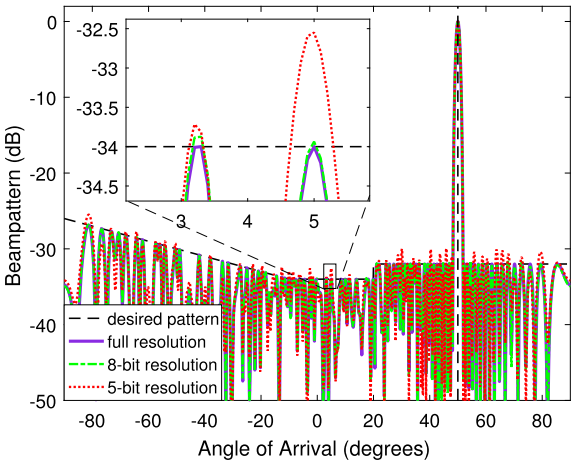

Fig. 11. The beampatterns with different phase shifter resolutions.

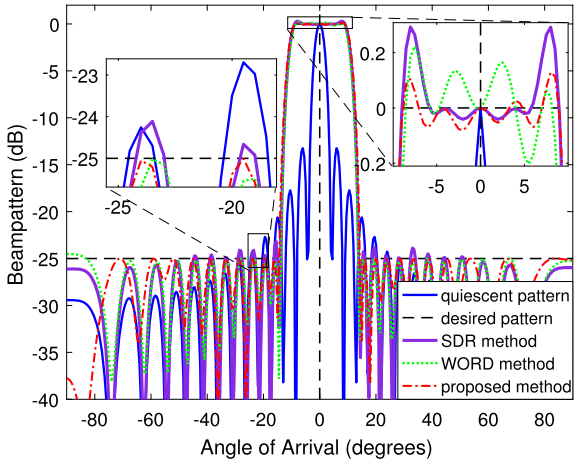

Fig. 12. The beampatterns with flat-top mainlobe.

$(u_0, v_0) = (-0.5, 0.5)$. A special region is divided in $\Theta = \{(u, v) \mid 0.2 \le u \le 0.9, -0.9 \le v \le -0.1\}$ and the desired level inside the region Θ is set to be lower than -30 dB. The desired level in the rest of region is required to be below -15 dB.

The resulting synthesized beampattern is depicted in Fig. 14(b) and Fig. 14(c) shows the top view of synthesized beampattern. From Fig. 14, it can be known that the sidelobes levels in the region Θ can be adjusted to the desired values as well as outside of the special region. It indicates that this proposed algorithm is also effective when the array is 2-D antenna array.

Moreover, the comparison of running time among different method is shown in Table 7. The unit of the time is second, and

Table 6
The obtained weights with mutual coupling effect.

n	w_n	n	w_n	n	w_n
1	$0.68e^{+j0.430}$	35	$1.67e^{-j0.375}$	69	$1.60e^{-j2.147}$
2	$0.33e^{-j2.144}$	36	$1.72e^{+j2.349}$	70	$1.56e^{+j0.567}$
3	$0.26e^{+j1.438}$	37	$1.75e^{-j1.218}$	71	$1.50e^{-j2.986}$
4	$0.28e^{+j3.028}$	38	$1.81e^{+j1.500}$	72	$1.44e^{-j0.269}$
5	$0.25e^{-j0.714}$	39	$1.85e^{-j2.057}$	73	$1.38e^{+j2.460}$
6	$0.33e^{+j2.514}$	40	$1.88e^{+j0.657}$	74	$1.32e^{-j1.092}$
7	$0.43e^{-j1.061}$	41	$1.91e^{-j2.910}$	75	$1.28e^{+j1.622}$
8	$0.27e^{+j1.564}$	42	$1.94e^{-j0.185}$	76	$1.25e^{-j1.932}$
9	$0.31e^{-j2.056}$	43	$1.96e^{+j2.532}$	77	$1.19e^{+j0.798}$
10	$0.48e^{+j0.744}$	44	$2.00e^{-j1.029}$	78	$1.14e^{-j2.788}$
11	$0.46e^{-j2.814}$	45	$2.01e^{+j1.695}$	79	$1.10e^{-j0.073}$
12	$0.51e^{-j0.103}$	46	$2.02e^{-j1.869}$	80	$1.03e^{+j2.627}$
13	$0.56e^{+j2.602}$	47	$2.06e^{+j0.849}$	81	$0.96e^{-j0.940}$
14	$0.57e^{-j1.011}$	48	$2.07e^{-j2.712}$	82	$0.89e^{+j1.806}$
15	$0.65e^{+j1.722}$	49	$2.07e^{-j0.002}$	83	$0.81e^{-j1.741}$
16	$0.70e^{-j1.785}$	50	$2.08e^{+j2.724}$	84	$0.77e^{+j0.974}$
17	$0.71e^{+j0.919}$	51	$2.07e^{-j0.840}$	85	$0.76e^{-j2.536}$
18	$0.78e^{-j2.641}$	52	$2.07e^{+j1.875}$	86	$0.71e^{+j0.222}$
19	$0.84e^{+j0.094}$	53	$2.07e^{-j1.682}$	87	$0.71e^{+j2.911}$
20	$0.90e^{+j2.789}$	54	$2.05e^{+j1.044}$	88	$0.66e^{-j0.658}$
21	$0.96e^{-j0.761}$	55	$2.05e^{-j2.523}$	89	$0.62e^{+j1.927}$
22	$0.99e^{+j1.970}$	56	$2.04e^{+j0.197}$	90	$0.60e^{-j1.638}$
23	$1.04e^{-j1.626}$	57	$2.02e^{+j2.917}$	91	$0.48e^{+j1.067}$
24	$1.11e^{+j1.103}$	58	$2.01e^{-j0.647}$	92	$0.32e^{-j2.455}$
25	$1.15e^{-j2.442}$	59	$1.99e^{+j2.073}$	93	$0.25e^{+j0.134}$
26	$1.20e^{+j0.274}$	60	$1.95e^{-j1.459}$	94	$0.30e^{-j2.869}$
27	$1.27e^{+j3.001}$	61	$1.92e^{+j1.219}$	95	$0.28e^{+j0.095}$
28	$1.32e^{-j0.564}$	62	$1.89e^{-j2.334}$	96	$0.44e^{+j2.658}$
29	$1.38e^{+j2.145}$	63	$1.84e^{+j0.389}$	97	$0.30e^{-j0.948}$
30	$1.44e^{-j1.408}$	64	$1.81e^{+j3.106}$	98	$0.59e^{+j1.032}$
31	$1.48e^{+j1.311}$	65	$1.77e^{-j0.450}$	99	$0.53e^{-j2.411}$
32	$1.53e^{-j2.264}$	66	$1.74e^{+j2.268}$	100	$0.36e^{-j0.648}$
33	$1.58e^{+j0.462}$	67	$1.71e^{-j1.296}$		
34	$1.62e^{-j3.096}$	68	$1.66e^{+j1.426}$		

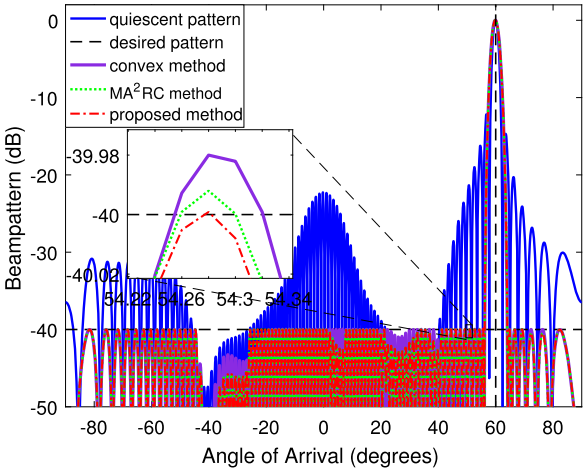

Fig. 13. Beampattern synthesis with mutual coupling effect.

Table 7
The running time with different array (TIME:s).

array size	convex method	MA^{2}RC method	proposed method
10×10	105.2	1.2	0.7
15×15	206.7	11.4	5.9
20×20	427.9	143.2	56.7

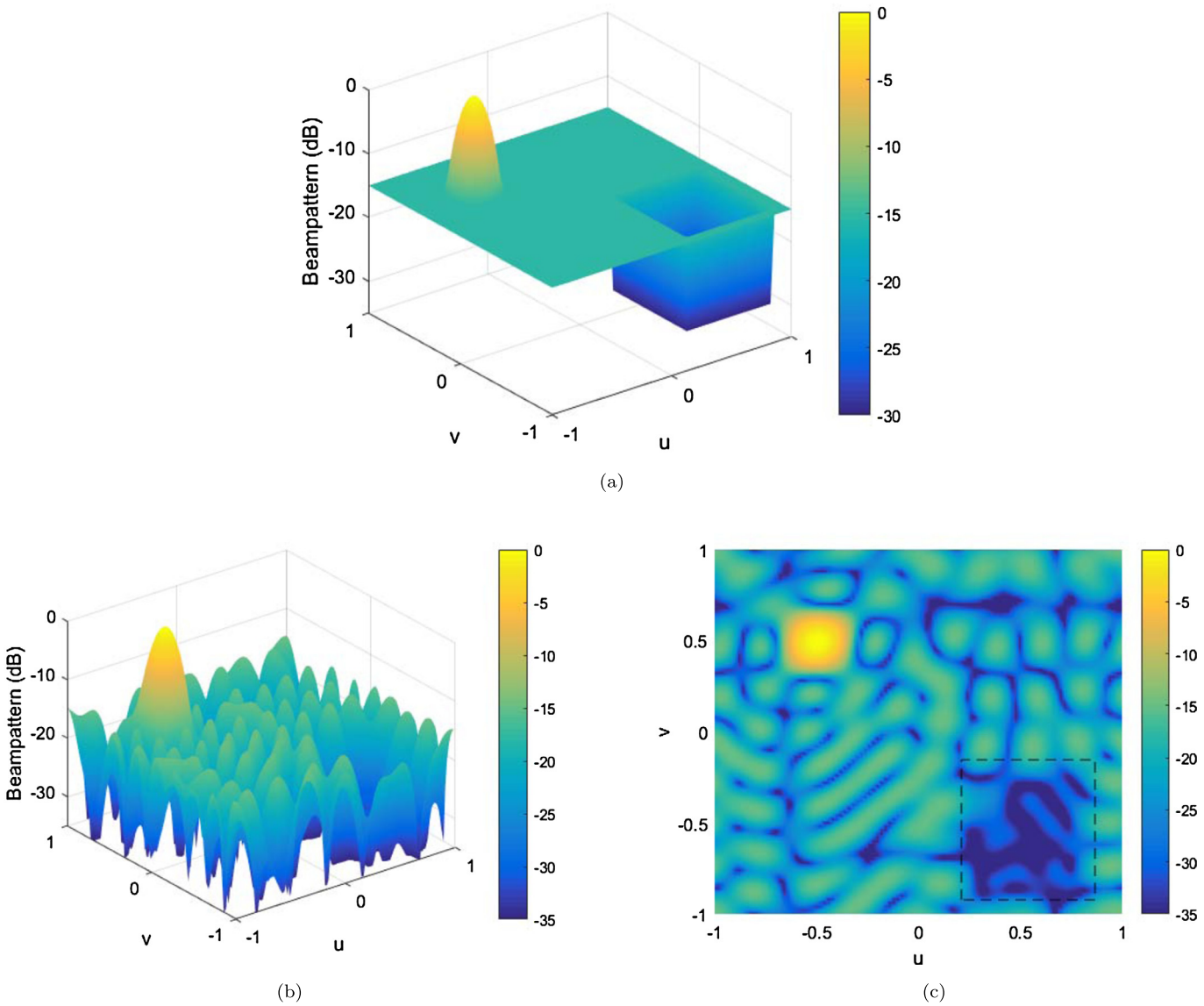

Fig. 14. Beampattern synthesis with a 2-D planar array. (a) The desired 2-D beampattern. (b) The 2-D synthesized beampattern. (c) Top view of the 2-D synthesized beampattern.

the sizes of array are set as 10×10, 15×15 and 20×20 respectively. As seen from Table 7, the proposed method takes the least time to converge to the desired beampattern.

7. Conclusions

In this paper, we have proposed a low computational complexity multi-point accurate array response control algorithm for beampattern synthesis. After the analysis of the two solutions in WORD algorithm, we can discard the weight vector calculated with the nonpositive coefficient and obtain the I-WORD algorithm. The proposed I-WORD algorithm can omit the selection procedure in the WORD algorithm. Then, based on the I-WORD algorithm, the proposed MI-WORD method is developed by finding the intersection of weight vector sets. A low dimension matrix is constructed with SVD of matrix formed by weight vector set, which is different from the MA^{2}RC method and leads to the low computational complexity of the MI-WORD algorithm. The simulations results show the effectiveness and the low complexity of the proposed method.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61771095 and 62031007.

Appendix A

From the proposed low complexity MI-WORD algorithm, we know that any weight vector which can control multi-point response can be expressed as

$$\begin{aligned} &\mathbf{w} = c_1[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\tilde{\mathbf{b}}_1^{\mathrm{T}}\ 1]^{\mathrm{T}} \\ &\text{s.t. } \mathbf{T}\tilde{\mathbf{b}}_1^{\mathrm{T}} = -\mathbf{t}, c_1 \neq 0. \end{aligned} \tag{A.1}$$

To make sure that $\mathbf{w}$ has the ability to avoid the possible beam center shift, the following derivative constraint has to be added in (A.1)

$$\mathrm{Re}[\mathbf{w}^{\mathrm{H}}\mathbf{d}(\theta_0)\mathbf{a}(\theta_0)\mathbf{w}] = 0 \tag{A.2}$$

where $\mathbf{d}(\theta_0) \equiv \frac{\partial \mathbf{a}(\theta)}{\partial(\theta)}|_{\theta=\theta_0}$. Considering $\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{U}_{12} = \mathbf{0}$, the derivative constraint can be rewritten as

$$\mathrm{Re}[\tilde{\mathbf{b}}_1^{\mathrm{H}}\mathbf{p}] = \gamma \tag{A.3}$$

where

$$\begin{aligned} \mathbf{p} &= \mathbf{U}_{12}^{\mathrm{H}}\mathbf{d}(\theta_0)\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}_{k+1,1} \\ \gamma &= -\mathrm{Re}[\mathbf{w}_{k+1,1}^{\mathrm{H}}\mathbf{d}(\theta_0)\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}_{k+1,1}]. \end{aligned} \tag{A.4}$$

Hence, the problem of multi-point response control with non-shifting constraint can be stated as

$$\begin{aligned} &\widehat{\mathbf{w}}_{k+1} = c_1[\mathbf{U}_{12}\ \mathbf{w}_{k+1,1}][\tilde{\mathbf{b}}_1^{\mathrm{T}}\ 1]^{\mathrm{T}} \\ &\text{s.t. } \mathbf{T}\tilde{\mathbf{b}}_1^{\mathrm{T}} = -\mathbf{t}, c_1 \neq 0, \mathrm{Re}[\tilde{\mathbf{b}}_1^{\mathrm{H}}\mathbf{p}] = \gamma. \end{aligned} \tag{A.5}$$

Let's define

$$\begin{aligned} \mathbf{T}_{\mathrm{c}} &\equiv \begin{bmatrix} \mathrm{Re}(\mathbf{T}) & -\mathrm{Im}(\mathbf{T}) \\ \mathrm{Im}(\mathbf{T}) & \mathrm{Re}(\mathbf{T}) \end{bmatrix} \\ \mathbf{b}_{\mathrm{c}} &\equiv \begin{bmatrix} \mathrm{Re}(\tilde{\mathbf{b}}_1^{\mathrm{T}}) & \mathrm{Im}(\tilde{\mathbf{b}}_1^{\mathrm{T}}) \end{bmatrix}^{\mathrm{T}} \\ \mathbf{t}_{\mathrm{c}} &\equiv \begin{bmatrix} -\mathrm{Re}(\mathbf{t}^{\mathrm{T}}) & -\mathrm{Im}(\mathbf{t}^{\mathrm{T}}) \end{bmatrix}^{\mathrm{T}} \\ \mathbf{p}_{\mathrm{c}} &\equiv \begin{bmatrix} \mathrm{Re}(\mathbf{p}^{\mathrm{T}}) & \mathrm{Im}(\mathbf{p}^{\mathrm{T}}) \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{A.6}$$

where $\mathrm{Im}(\cdot)$ returns the imaginary part of a complex number. Therefore, the constraint on $\tilde{\mathbf{b}}_1^{\mathrm{T}}$ can be simplified to

$$\mathbf{C}\mathbf{b}_{\mathrm{c}} = \mathbf{k} \tag{A.7}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{T}_{\mathrm{c}}^{\mathrm{T}}\ \mathbf{p}_{\mathrm{c}} \end{bmatrix}^{\mathrm{T}}, \mathbf{k} = \begin{bmatrix} \mathbf{t}_{\mathrm{c}}^{\mathrm{T}}\ \gamma \end{bmatrix}^{\mathrm{T}} \tag{A.8}$$

Consequently, the weight vector which can avoid main beam shifting can be obtained as

$$\widehat{\mathbf{w}}_{k+1} = c_1[\Xi\ \mathbf{w}_{k+1,1}][\mathbf{C}^{\dagger}\mathbf{k} + \mathbf{c}_n\ 1]^{\mathrm{T}},\ c_1 \neq 0,\ \forall \mathbf{c}_n \in \mathcal{N}(\mathbf{C}), \tag{A.9}$$

where $\Xi \equiv [\mathbf{U}_{12}\ j\mathbf{U}_{12}]$.

More details can be seen in [30].

References

[1] A. Basit, W. Wang, S.Y. Nusenu, Adaptive transmit array sidelobe control using FDA-MIMO for tracking in joint radar-communications, Digit. Signal Process. 97 (2020) 102619.

[2] Y. Liu, J. Bai, K.D. Xu, Z. Xu, F. Han, Q. Liu, Y. Jay Guo, Linearly polarized shaped power pattern synthesis with sidelobe and cross-polarization control by using semidefinite relaxation, IEEE Trans. Antennas Propag. 66 (6) (2018) 3207–3212.

[3] Y. Liu, M. Li, R.L. Haupt, Y.J. Guo, Synthesizing shaped power patterns for linear and planar antenna arrays including mutual coupling by refined joint rotation/phase optimization, IEEE Trans. Antennas Propag. 68 (6) (2020) 4648–4657.

[4] G.C. Alexandropoulos, Position aided beam alignment for millimeter wave backhaul systems with large phased arrays, in: 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, 2017, pp. 1–5.

[5] H.G. Hoang, H.D. Tuan, B. Vo, Low-dimensional sdp formulation for large antenna array synthesis, IEEE Trans. Antennas Propag. 55 (6) (2007) 1716–1725.

[6] C.L. Dolph, A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level, Proc. IRE 34 (6) (1946) 335–348.

[7] C. Tseng, L. Griffiths, A simple algorithm to achieve desired patterns for arbitrary arrays, IEEE Trans. Signal Process. 40 (11) (1992) 2737–2746.

[8] B. Ng, M. Er, C. Kot, A flexible array synthesis method using quadratic programming, IEEE Trans. Antennas Propag. 41 (11) (1993) 1541–1550.

[9] Y. Liu, X. Huang, K. Xu, Z. Song, S. Yang, Pattern synthesis of unequally spaced linear arrays including mutual coupling using iterative fft via virtual active element pattern expansion, IEEE Trans. Antennas Propag. 65 (8) (2017) 3950–3958.

[10] H. Shen, B. Wang, Two-dimensional unitary matrix pencil method for synthesizing sparse planar arrays, Digit. Signal Process. 73 (2018) 40–46.

[11] B.V. Ha, M. Mussetta, P. Pirinoli, R.E. Zich, Modified compact genetic algorithm for thinned array synthesis, IEEE Antennas Wirel. Propag. Lett. 15 (2016) 1105–1108.

[12] D.W. Boeringer, D.H. Werner, Particle swarm optimization versus genetic algorithms for phased array synthesis, IEEE Trans. Antennas Propag. 52 (3) (2004) 771–779.

[13] V. Murino, A. Trucco, C. Regazzoni, Synthesis of unequally spaced arrays by simulated annealing, IEEE Trans. Signal Process. 44 (1) (1996) 119–122.

[14] B. Fuchs, Application of convex relaxation to array synthesis problems, IEEE Trans. Antennas Propag. 62 (2) (2014) 634–640.

[15] H. L, V. Trees, Optimum Array Processing, John Wiley and Sons, Inc., New York, NY, USA, 2002.

[16] B. Fuchs, S. Rondineau, Array pattern synthesis with excitation control via norm minimization, IEEE Trans. Antennas Propag. 64 (10) (2016) 4228–4234.

[17] H. Lebret, S. Boyd, Antenna array pattern synthesis via convex optimization, IEEE Trans. Signal Process. 45 (3) (1997) 526–532.

[18] K.M. Tsui, S.C. Chan, Pattern synthesis of narrowband conformal arrays using iterative second-order cone programming, IEEE Trans. Antennas Propag. 58 (6) (2010) 1959–1970.

[19] F. Wang, V. Balakrishnan, P.Y. Zhou, J. Chen, C. Frank, Optimal array pattern synthesis using semidefinite programming, IEEE Trans. Signal Process. 51 (5) (2003) 1172–1183.

[20] Y. Xu, X. Shi, J. Xu, L. Huang, W. Li, Range–angle-decoupled beampattern synthesis with subarray-based frequency diverse array, Digit. Signal Process. 64 (2017) 49–59.

[21] W. Keizer, Large planar array thinning using iterative FFT techniques, IEEE Trans. Antennas Propag. 57 (10) (2009) 3359–3362.

[22] Y. Liu, L. Chen, C. Zhu, Y. Ban, Y.J. Guo, Efficient and accurate frequency-invariant beam pattern synthesis utilizing iterative spatiotemporal Fourier transform, IEEE Trans. Antennas Propag. 68 (8) (2020) 6069–6079.

[23] Y. Liu, J. Zheng, M. Li, Q. Luo, Y. Rui, Y.J. Guo, Synthesizing beam-scannable thinned massive antenna array utilizing modified iterative FFT for millimeter-wave communication, IEEE Antennas Wirel. Propag. Lett. 19 (11) (2020) 1983–1987.

[24] S. Caorsi, A. Massa, M. Pastorino, A. Randazzo, Optimization of the difference patterns for monopulse antennas by a hybrid real/integer-coded differential evolution method, IEEE Trans. Antennas Propag. 53 (1) (2005) 372–376.

[25] C. Cui, Y. Jiao, L. Zhang, Synthesis of some low sidelobe linear arrays using hybrid differential evolution algorithm integrated with convex programming, IEEE Antennas Wirel. Propag. Lett. 16 (2017) 2444–2448.

[26] M. Li, Y. Liu, Y.J. Guo, Shaped power pattern synthesis of a linear dipole array by element rotation and phase optimization using dynamic differential evolution, IEEE Antennas Wirel. Propag. Lett. 17 (4) (2018) 697–701.

[27] F. Liu, Y. Liu, F. Han, Y. Ban, Y. Jay Guo, Synthesis of large unequally spaced planar arrays utilizing differential evolution with new encoding mechanism and Cauchy mutation, IEEE Trans. Antennas Propag. 68 (6) (2020) 4406–4416.

[28] X. Zhang, Z. He, X. Xia, B. Liao, X. Zhang, Y. Yang, Oparc: optimal and precise array response control algorithm part I: fundamentals, IEEE Trans. Signal Process. 67 (3) (2019) 652–667.

[29] X. Zhang, Z. He, B. Liao, X. Zhang, W. Peng, Pattern synthesis for arbitrary arrays via weight vector orthogonal decomposition, IEEE Trans. Signal Process. 66 (5) (2018) 1286–1299.

[30] X. Zhang, Z. He, B. Liao, X. Zhang, Pattern synthesis with multipoint accurate array response control, IEEE Trans. Antennas Propag. 65 (8) (2017) 4075–4088.

[31] X. Zhang, Z. He, B. Liao, X. Zhang, Y. Yang, Pattern synthesis via oblique projection-based multipoint array response control, IEEE Trans. Antennas Propag. 67 (7) (2019) 4602–4616.

[32] G. Strang, Linear Algebra and Its Applications, 4th ed., Wellesley-Cambridge, New York, NY, USA, 2005.

[33] E. Friel, K. Pasala, Effects of mutual coupling on the performance of stap antenna arrays, IEEE Trans. Aerosp. Electron. Syst. 36 (2) (2000) 518–527.

Weilai Peng was born in Jiangxi, China. He received the B.Eng. degree in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2015, where he is currently pursuing the Ph.D. degree in electronic engineering. His current research interests include array signal processing , digital beamforming, MIMO radar, and optimization theory.

Xuejing Zhang was born in Hebei, China. He received the B.S. degree in electrical engineering from Huaqiao University, Xiamen, China, and the M.S. degree in signal and information processing from Xidian University,

Xi'an, China, and the Ph.D. degree in signal and information processing from University of Electronic Science and Technology of China, Chengdu, China, in 2011, 2014, and 2019, respectively. From 2017 to 2019, he was a visiting student with the University of Delaware, Newark, DE, USA. His research interests include array signal processing and wireless communications.

Zishu He was born in Sichuan, China, in 1962. He received the B.S., M.S., and Ph.D. degrees in signal and information processing from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1984, 1988, and 2000, respectively. He is currently a Professor with the School of Information and Communication Engineering, UESTC. His current research interests include array signal processing, digital beam forming, the theory on multiple-input multiple-output (MIMO) communication, and MIMO radar, adaptive signal processing and interference cancellation.

Julan Xie was born in Leiyang, China. She received the B.S. and Ph.D. degrees in signal and information processing from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2005 and 2012, respectively. She is currently an Associate Professor of signal and information processing with the School of Information and Communication Engineering, UESTC. Her current research interests include array signal processing, digital beamforming, interference cancellation, and array optimization.

Chunlin Han was born in Hebei, China, in 1962. He received the B.S., M.S., and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1984, 1989, and 2004, respectively. He is currently a Professor with the School of Information and Communication Engineering, UESTC. His current research interests include array signal processing and digital beamforming.