Pattern Synthesis via Oblique Projection-Based Multipoint Array Response Control

Xuejing Zhang, *Student Member, IEEE*, Zishu He, *Member, IEEE*, Bin Liao, *Senior Member, IEEE*, Xuepan Zhang, and Yue Yang, *Student Member, IEEE*

***Abstract*— In this paper, we present two array response control algorithms for pattern synthesis with the aid of oblique projection. The proposed algorithms are developed on the basis of the weight vector orthogonal decomposition (WORD) approach, and both of them can control the array responses of multiple points starting from an arbitrarily given weight vector. They provide closed-form expressions and thus are computationally attractive and convenient to implement. Furthermore, for the preassigned angles to be controlled, the array responses can be adjusted flexibly and separately, and hence, the process of response control can be readily accomplished without complete recalculation, if some of the desired levels require adjustments. In addition, the second proposed algorithm modifies the first one and realizes the array response control without a beam axis shift. By successively performing the proposed algorithms to adjust the response to meet certain requirements, the array pattern can be synthesized. Extensive examples are provided to demonstrate the performances of the proposed algorithms in array response control and the effectiveness in pattern synthesis.**

***Index Terms*— Array pattern synthesis, array response control, array signal processing, oblique projection.**

I. Introduction

The array antenna has drawn considerable attention in the past decades because of its importance for radar, wireless communications, remote sensing, and many other applications. Accordingly, array pattern synthesis plays a critical role in, for example, suppressing undesirable interferences, enhancing the signal power at specific regions, or realizing multiuser reception. In pattern synthesis, it is expected to design a weight vector such that the corresponding array response satisfies specific requirements.

During the past several decades, quite a number of pattern synthesis algorithms have been presented, in either adaptive (data-dependent) or nonadaptive (data-independent) way. In the adaptive manner, the pattern is adaptively synthesized with measurements to force beams and nulls at the directions of targets and interferences, respectively. For instance, the linearly constrained minimum variance (LCMV) method [1] adjusts the array responses by imposing multiple directional constraints on the array responses. In [2], the sidelobe of the LCMV beamformer is optimized with the semidefinite relaxation (SDR) technique [3]. A diagonal loading method is used in [4] to adjust the responses at a range of arrival angles to exceed unity. In order to maintain a fairly stable gain in the region of interest, a relatively flat main lobe is synthesized by computing the outer product matrix and autocorrelation sequence of the array weight vector in [5] and [6] or using the iterative second-order cone programming (SOCP) [7].

To synthesize array responses in data-independent scenarios (which is the focus of our discussion later), several global optimization approaches, such as genetic algorithm (GA) [12], particle swarm optimization (PSO) [13], and simulated annealing (SA) [14], have been adopted, with the drawback of heavy computation burden. The method in [15] was developed to adjust array responses, by minimizing the mean-square error between the array response and the desired one over a main lobe region subject to a mean-square sidelobe constraint. In [16], a simple iterative algorithm is presented to design desired patterns by solving a sequence of linearly constrained least-squares problems. Several pattern synthesis algorithms [17]–[20] have been developed by taking advantage of the adaptive array theory [21]. The convex optimization theory [22] has also been fully exploited in [23]–[27], to control array response according to the given specification.

Recently, we devised a class of pattern synthesis approaches via array response control. In general, these algorithms can be grouped into two categories, i.e., single-point array response control and multipoint array response control. For the first category, including the accurate array response control ($\mathrm{A}^2\mathrm{RC}$) algorithm [28] and weight vector orthogonal decomposition (WORD) algorithm [29], the pattern is synthesized by iteratively controlling the response of a single angle. For the second one, such as the multipoint accurate array response control ($\mathrm{MA}^2\mathrm{RC}$) algorithm [30] and flexible array response control algorithm via oblique projection (FARCOP) [31], responses at multiple angles can be simultaneously adjusted in each iterative so as to achieve a faster

Manuscript received May 23, 2018; revised October 25, 2018; accepted April 7, 2019. Date of publication April 15, 2019; date of current version July 3, 2019. This work was supported in part by the National Nature Science Foundation of China under Grant 61671139, Grant 61671137, Grant 61771316, Grant 61701499, and Grant 61871085, in part by the Foundation of Shenzhen under Grant JCYJ20170302150044331, and in part by the China Scholarship Council (CSC). *(Corresponding author: Xuejing Zhang.)*

X. Zhang, Z. He, and Y. Yang are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: xjzhang7@163.com; zshe@uestc.edu.cn; yueyang@std.uestc.edu.cn).

B. Liao is with the Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, China (e-mail: binliao@szu.edu.cn).

X. Zhang is with the Qian Xuesen Laboratory of Space Technology, Beijing 100094, China (e-mail: zhangxuepan@qxslab.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAP.2019.2911317

synthesis. However, these algorithms may cause a beam axis shift of the resulting beampattern. In other words, the beam axis of the obtained pattern may be different from the desired one. Although the modified multipoint accurate array response control (M^{2}A^{2}RC) algorithm [30] provides a solution to adjust array response levels without leading to beam axis shift, the flexibility needs to be improved and a high computational complexity is suffered.

The above-mentioned shortcomings of the existing methods motivate us to develop new array response control approaches for pattern synthesis. Toward this end, in this paper, we present two array response control algorithms in sequence, by taking advantage of the WORD method [29] and oblique projection [32]. More specifically, given an initial weight vector, a series of weight vectors is first computed by using the WORD approach. Note that each weight vector is able to control the array response level at a preassigned angle as desired. Then, to realize the multipoint array response control, in the first algorithm, a linear constraint is imposed on the ultimate weight to restrict its responses at the predescribed angles, and an analytical expression of the final weight is obtained by using the oblique projection operators. In addition, to avoid the possible beam axis shift, we modify the first algorithm by imposing an additional derivative constraint on the beampattern, and this leads to the second algorithm that also has a closed-form solution. Note that our algorithms are able to control the array responses of multiple angles separately. For this reason, the weight calculation needs not be completely reconducted in the case when some desired levels are changed. The application of the proposed algorithms to array pattern synthesis is finally discussed.

The rest of this paper is organized as follows. In Section II, the WORD algorithm is briefly introduced. The two proposed algorithms are presented in Section III and their applications to array pattern synthesis are provided in Section IV. Representative simulations are presented in Section V, and conclusions are drawn in Section VI.

Notations: We use bold upper case and lower case letters to represent matrices and vectors, respectively. In particular, we use $\mathbf{I}$ to stand for the identity matrix. $j \triangleq \sqrt{-1}$. $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ denote the transpose and Hermitian transpose, respectively. $|\cdot|$ is the absolute value. $\Re(\cdot)$ represents the real part of a complex number. $\mathbb{R}$ and $\mathbb{C}$ denote the sets of real and complex numbers, respectively. $\mathcal{R}(\cdot)$ returns the column space of the input matrix, and $\mathcal{R}^{\perp}(\cdot)$ is the orthogonal complementary space of $\mathcal{R}(\cdot)$. $\mathbf{P_Z}$ and $\mathbf{P_Z^{\perp}}$ represent the projection matrices onto $\mathcal{R}(\mathbf{Z})$ and $\mathcal{R}^{\perp}(\mathbf{Z})$, respectively.

II. WORD Algorithm

To control the array response, we design a weight vector $\mathbf{w}$ such that

$$L(\theta,\theta_0) \triangleq |\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta)|^2/|\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2 \tag{1}$$

meets some specific requirements. In (1), θ_0 denotes the angle of the beam axis, and $\mathbf{a}(\theta)$ is the steering vector at θ and is given by

$$\mathbf{a}(\theta) = [g_1(\theta)e^{-j\omega\tau_1(\theta)}, \ldots, g_N(\theta)e^{-j\omega\tau_N(\theta)}]^{\mathrm{T}} \tag{2}$$

where N is the number of array elements, $g_n(\theta)$ denotes the pattern of the nth element, $\tau_n(\theta)$ represents the time delay between the nth element and the reference point, $n = 1, \ldots, N$, and ω denotes the operating frequency.

For a given weight vector $\mathbf{w}_{\mathrm{pre}}$, the WORD algorithm [29] is able to precisely control array response level at one preassigned angle θ_c as desired, with a closed-form expression. More specifically, the new weight vector satisfying the single-point response requirement is analytically obtained as

$$\breve{\mathbf{w}}_{\mathrm{new}} = [\mathbf{w}_{\perp}\ \mathbf{w}_{\parallel}][1\ \beta]^{\mathrm{T}} \tag{3}$$

where $\mathbf{w}_{\perp}$ and $\mathbf{w}_{\parallel}$ are orthogonally decomposed from the previous weight vector $\mathbf{w}_{\mathrm{pre}}$ as

$$\mathbf{w}_{\perp} \triangleq \mathbf{P}^{\perp}_{\mathbf{a}(\theta_c)}\mathbf{w}_{\mathrm{pre}}, \quad \mathbf{w}_{\parallel} \triangleq \mathbf{P}_{\mathbf{a}(\theta_c)}\mathbf{w}_{\mathrm{pre}}. \tag{4}$$

In (3), the real-valued number β can be selected to be either β_a or β_b, both of which can be determined by the desired level ρ_c at θ_c. In [29], it has been derived that

$$\beta_a = \frac{-\Re(\mathbf{B}(1,2)) + d}{\mathbf{B}(2,2)}, \quad \beta_b = \frac{-\Re(\mathbf{B}(1,2)) - d}{\mathbf{B}(2,2)} \tag{5}$$

where $\mathbf{B}$ and d satisfy

$$\mathbf{B} = \begin{bmatrix}\mathbf{w}_{\perp}^{\mathrm{H}}\mathbf{a}(\theta_c)\\ \mathbf{w}_{\parallel}^{\mathrm{H}}\mathbf{a}(\theta_c)\end{bmatrix}\begin{bmatrix}\mathbf{w}_{\perp}^{\mathrm{H}}\mathbf{a}(\theta_c)\\ \mathbf{w}_{\parallel}^{\mathrm{H}}\mathbf{a}(\theta_c)\end{bmatrix}^{\mathrm{H}} - \rho_c\begin{bmatrix}\mathbf{w}_{\perp}^{\mathrm{H}}\mathbf{a}(\theta_0)\\ \mathbf{w}_{\parallel}^{\mathrm{H}}\mathbf{a}(\theta_0)\end{bmatrix}\begin{bmatrix}\mathbf{w}_{\perp}^{\mathrm{H}}\mathbf{a}(\theta_0)\\ \mathbf{w}_{\parallel}^{\mathrm{H}}\mathbf{a}(\theta_0)\end{bmatrix}^{\mathrm{H}} \tag{6}$$

$$d = \sqrt{\Re^2(\mathbf{B}(1,2)) - \mathbf{B}(1,1)\mathbf{B}(2,2)}. \tag{7}$$

To obtain the ultimate expression of $\breve{\mathbf{w}}_{\mathrm{new}}$ that adjusts the response level of θ_c to ρ_c, the one (either β_a or β_b) that minimizes $F(\beta) = \|\mathbf{P}^{\perp}_{\mathbf{w}_{\mathrm{pre}}}\breve{\mathbf{w}}_{\mathrm{new}}/\|\breve{\mathbf{w}}_{\mathrm{new}}\|_2\|_2^2$ is selected. Again, it should be emphasized that the WORD algorithm can only control array response of a single point. Moreover, it may lead to a possible beam axis shift on the resulting beampattern. To overcome these deficiencies, we present two array response control algorithms in the following, with the aid of oblique projection.

III. Proposed Array Response Control Algorithms

In this section, two multipoint response control algorithms are presented. For clarity, we denote by $\theta_1, \ldots, \theta_Q$ the angles to be controlled, and let $\rho_1, \ldots, \rho_Q$ be the corresponding desired levels. Following the notation in Section II, the previous weight is denoted as $\mathbf{w}_{\mathrm{pre}}$, and we assume for convenience that $\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_0) = 1$, which is always satisfied via scaling if only $\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_0) \neq 0$. For the notation clarity, we define $\mathbf{A}(\theta_i, \ldots, \theta_j) \triangleq [\mathbf{a}(\theta_i), \ldots, \mathbf{a}(\theta_j)]$.

A. First Algorithm

In this section, given the previous weight vector $\mathbf{w}_{\mathrm{pre}}$, we consider how to find a new weight $\mathbf{w}_{\mathrm{new}}$ with closed-form expression satisfying

$$L_{\mathrm{new}}(\theta_q,\theta_0) = \rho_q, \quad q = 1, \ldots, Q \tag{8}$$

where $L_{\mathrm{new}}(\theta,\theta_0)$ denotes the corresponding pattern of $\mathbf{w}_{\mathrm{new}}$. In addition, small pattern variations on the uncontrolled points are desired.

As aforementioned, the WORD algorithm is able to adjust the array response level at one prescribed point. Thus, for any $q \in \{1, \ldots, Q\}$, the resulting weight vector $\breve{\mathbf{w}}_{\text{new},q}$, which adjusts the response level of θ_q to its desired level ρ_q, i.e., $|\breve{\mathbf{w}}_{\text{new},q}^{\text{H}}\mathbf{a}(\theta_q)|^2/|\breve{\mathbf{w}}_{\text{new},q}^{\text{H}}\mathbf{a}(\theta_0)|^2 = \rho_q$, is expressed as

$$\breve{\mathbf{w}}_{\text{new},q} = [\mathbf{w}_{q,\perp}\ \mathbf{w}_{q,\parallel}][1\ \beta_q]^{\text{T}} \tag{9}$$

where $\mathbf{w}_{q,\perp} = \mathbf{P}_{\mathbf{a}(\theta_q)}^{\perp}\mathbf{w}_{\text{pre}}$ and $\mathbf{w}_{q,\parallel} = \mathbf{P}_{\mathbf{a}(\theta_q)}\mathbf{w}_{\text{pre}}$. Once $\breve{\mathbf{w}}_{\text{new},q}$ in (9) is obtained, we scale it for later use as

$$\mathbf{w}_{\text{new},q} = \breve{\mathbf{w}}_{\text{new},q}/(\mathbf{a}^{\text{H}}(\theta_0)\breve{\mathbf{w}}_{\text{new},q}),\ \ q = 1, \ldots, Q. \tag{10}$$

For $\forall q \in \{1, \ldots, Q\}$, it can be easily verified that the resulting $\mathbf{w}_{\text{new},q}$ in (10) satisfies

$$\mathbf{w}_{\text{new},q}^{\text{H}}\mathbf{a}(\theta_0) = 1 \tag{11}$$

$$\left|\mathbf{w}_{\text{new},q}^{\text{H}}\mathbf{a}(\theta_q)\right|^2 = \rho_q. \tag{12}$$

To realize the array response control task in (8), we propose to construct the qualified $\mathbf{w}_{\text{new}}$ based on the Q weight vectors $\mathbf{w}_{\text{new},1}, \ldots, \mathbf{w}_{\text{new},Q}$. More specifically, we aim to devise a unified weight vector $\mathbf{w}_{\text{new}}$ satisfying

$$\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_0) = 1 \tag{13a}$$

$$\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_q) = e^{j\varphi_q}\mathbf{w}_{\text{new},q}^{\text{H}}\mathbf{a}(\theta_q),\ \ \ q = 1, \ldots, Q \tag{13b}$$

or compactly

$$\breve{\mathbf{A}}^{\text{H}}\mathbf{w}_{\text{new}} = \mathbf{f} \tag{14}$$

where

$$\breve{\mathbf{A}} \triangleq \mathbf{A}(\theta_0, \theta_1, \ldots, \theta_Q) \in \mathbb{C}^{N\times(Q+1)}$$
$$\mathbf{f} \triangleq \left[1, e^{j\varphi_1}\mathbf{w}_{\text{new},1}^{\text{H}}\mathbf{a}(\theta_1), \ldots, e^{j\varphi_Q}\mathbf{w}_{\text{new},Q}^{\text{H}}\mathbf{a}(\theta_Q)\right]^{\text{H}} \in \mathbb{C}^{Q+1}$$

and the real-valued φ_qs, $q = 1, \ldots, Q$, can be arbitrarily specified. It can be verified that the weight vector satisfying $\mathbf{w}_{\text{new}}$ also fulfills the response control requirements in (8). Clearly, for the specific case of $Q = N - 1$, we have $\mathbf{w}_{\text{new}} = \breve{\mathbf{A}}^{-1}\mathbf{f}$. However, if $Q < N - 1$, the linear equation (14) with respect to the weight vector $\mathbf{w}_{\text{new}}$ is underdetermined and, hence, has infinitely many solutions. To this end, we devise a closed-form solution to the problem (14) via oblique projection [32]. As we show later, the potential benefit may be small pattern variations at the uncontrolled points.

To proceed, let us define a matrix $\breve{\mathbf{A}}_{i-}$ obtained by removing $\mathbf{a}(\theta_i)$ from $\breve{\mathbf{A}}$ as

$$\breve{\mathbf{A}}_{i-} \triangleq \mathbf{A}(\theta_0, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_Q) \in \mathbb{C}^{N\times Q} \tag{15}$$

and accordingly define

$$\mathbf{T}_0 \triangleq \mathbf{I} - \mathbf{E}_{\breve{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)}^{\text{H}} \tag{16a}$$

$$\mathbf{T}_q \triangleq \mathbf{E}_{\mathbf{a}(\theta_q)|\breve{\mathbf{A}}_{q-}}^{\text{H}},\ \ q = 1, \ldots, Q \tag{16b}$$

where $\mathbf{E}_{\breve{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)}$ and $\mathbf{E}_{\mathbf{a}(\theta_q)|\breve{\mathbf{A}}_{q-}}$ are the oblique projectors[1] such as

$$\mathbf{E}_{\breve{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)} = \breve{\mathbf{A}}_{0-}\left(\breve{\mathbf{A}}_{0-}^{\text{H}}\mathbf{P}_{\mathbf{a}(\theta_0)}^{\perp}\breve{\mathbf{A}}_{0-}\right)^{-1}\breve{\mathbf{A}}_{0-}^{\text{H}}\mathbf{P}_{\mathbf{a}(\theta_0)}^{\perp}$$
$$\mathbf{E}_{\mathbf{a}(\theta_q)|\breve{\mathbf{A}}_{q-}} = \mathbf{a}(\theta_q)\left(\mathbf{a}^{\text{H}}(\theta_q)\mathbf{P}_{\breve{\mathbf{A}}_{q-}}^{\perp}\mathbf{a}(\theta_q)\right)^{-1}\mathbf{a}^{\text{H}}(\theta_q)\mathbf{P}_{\breve{\mathbf{A}}_{q-}}^{\perp}.$$

Algorithm 1 First Algorithm

```
give the previous weight vector $\mathbf{w}_{\text{pre}}$, $\theta_q$ and its desired level $\rho_q$, calculate $\mathbf{T}_0$ and $\mathbf{T}_q$ by (16), $q = 1, \ldots, Q$
for $q = 1, \ldots, Q$ do
    calculate $\beta_q$ using the WORD algorithm such that $L_{\text{new}}(\theta_q, \theta_0) = \rho_q$
    obtain $\breve{\mathbf{w}}_{\text{new},q} = [\mathbf{w}_{q,\perp},\ \mathbf{w}_{q,\parallel}][1,\ \beta_q]^{\text{T}}$
    obtain $\mathbf{w}_{\text{new},q} = \breve{\mathbf{w}}_{\text{new},q}/(\mathbf{a}^{\text{H}}(\theta_0)\breve{\mathbf{w}}_{\text{new},q})$
end for
output the weight vector $\mathbf{w}_{\text{new}}$ by (20), where $\varphi_q$, $q = 1, \ldots, Q$, can be arbitrarily specified
```

Then, it can be readily shown that

$$\mathbf{T}_i^{\text{H}}\mathbf{a}(\theta_j) = \begin{cases} \mathbf{a}(\theta_i), & \text{if } j = i \\ \mathbf{0}, & \text{if } j \neq i \end{cases} \tag{17}$$

where $i, j = 0, 1, \ldots, Q$. In other words, the linear transformation $\mathbf{T}_i^{\text{H}}$ passes $\mathbf{a}(\theta_i)$ without any change while blocking the components of $\mathbf{a}(\theta_j)$ as long as $j \neq i$.

By exploiting the above-mentioned properties, it will be shown below that $\mathbf{w}_{\text{new}}$ can be obtained as a linear combination of the transformed weight vectors $\mathbf{w}_{\text{pre}}$ and $\mathbf{w}_{\text{new},q}$s. More specifically, we recall the assumption that $\mathbf{w}_{\text{pre}}^{\text{H}}\mathbf{a}(\theta_0) = 1$. Then, according to (17), one gets

$$\mathbf{w}_{\text{pre}}^{\text{H}}\mathbf{T}_0^{\text{H}}\mathbf{a}(\theta_j) = \begin{cases} \mathbf{w}_{\text{pre}}^{\text{H}}\mathbf{a}(\theta_0) = 1, & \text{if } j = 0 \\ \mathbf{0}, & \text{if } j \neq 0 \end{cases} \tag{18}$$

and

$$\mathbf{w}_{\text{new,q}}^{\text{H}}\mathbf{T}_q^{\text{H}}\mathbf{a}(\theta_j) = \begin{cases} \mathbf{w}_{\text{new,q}}^{\text{H}}\mathbf{a}(\theta_q), & \text{if } j = q \\ \mathbf{0}, & \text{if } j \neq q. \end{cases} \tag{19}$$

According to (18) and (19), we can thus express a qualified $\mathbf{w}_{\text{new}}$ as

$$\mathbf{w}_{\text{new}} = \mathbf{T}_0\mathbf{w}_{\text{pre}} + \sum_{q=1}^{Q} e^{-j\varphi_q}\mathbf{T}_q\mathbf{w}_{\text{new},q}. \tag{20}$$

Obviously, it can be verified that

$$\begin{aligned} \mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_0) &= \mathbf{w}_{\text{pre}}^{\text{H}}\mathbf{T}_0^{\text{H}}\mathbf{a}(\theta_0) + \sum_{q=1}^{Q} e^{j\varphi_q}\mathbf{w}_{\text{new},q}^{\text{H}}\mathbf{T}_q^{\text{H}}\mathbf{a}(\theta_0) \\ &= \mathbf{w}_{\text{pre}}^{\text{H}}\mathbf{a}(\theta_0) \\ &= 1 \end{aligned} \tag{21a}$$

$$\begin{aligned} \mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_q) &= \mathbf{w}_{\text{pre}}^{\text{H}}\mathbf{T}_0^{\text{H}}\mathbf{a}(\theta_q) + \sum_{i=1}^{Q} e^{j\varphi_i}\mathbf{w}_{\text{new},i}^{\text{H}}\mathbf{T}_i^{\text{H}}\mathbf{a}(\theta_q) \\ &= e^{j\varphi_q}\mathbf{w}_{\text{new},q}^{\text{H}}\mathbf{a}(\theta_q),\ \ q = 1, \ldots, Q. \end{aligned} \tag{21b}$$

Therefore, the resulting $\mathbf{w}_{\text{new}}$ in (20) solves the linear equation (14). In other words, it realizes the multipoint array response control as described in (8).

It is easy to find from (21b) that the resulting $\mathbf{w}_{\text{new}}$ in (20) is able to adjust the response of θ_q separately, by simply renewing the corresponding weight $\mathbf{w}_{\text{new},q}$, $q = 1, \ldots, Q$. In this manner, the responses at other controlled points (i.e., θ_is,

[1] The orthogonal projector $\mathbf{P}_{\mathbf{Z}}$ with $\mathbf{Z} = [\mathbf{G}\ \ \mathbf{S}]$ can be expressed as $\mathbf{P}_{\mathbf{Z}} = \mathbf{E}_{\mathbf{G}|\mathbf{S}} + \mathbf{E}_{\mathbf{S}|\mathbf{G}}$, where $\mathbf{E}_{\mathbf{G}|\mathbf{S}}$ and $\mathbf{E}_{\mathbf{S}|\mathbf{G}}$ are oblique projectors satisfying $\mathbf{E}_{\mathbf{G}|\mathbf{S}} = \mathbf{G}(\mathbf{G}^{\text{H}}\mathbf{P}_{\mathbf{S}}^{\perp}\mathbf{G})^{-1}\mathbf{G}^{\text{H}}\mathbf{P}_{\mathbf{S}}^{\perp}$ and $\mathbf{E}_{\mathbf{S}|\mathbf{G}} = \mathbf{S}(\mathbf{S}^{\text{H}}\mathbf{P}_{\mathbf{G}}^{\perp}\mathbf{S})^{-1}\mathbf{S}^{\text{H}}\mathbf{P}_{\mathbf{G}}^{\perp}$. It is known that $\mathbf{E}_{\mathbf{G}|\mathbf{S}}\mathbf{G} = \mathbf{G}$, $\mathbf{E}_{\mathbf{G}|\mathbf{S}}\mathbf{S} = \mathbf{0}$, $\mathbf{E}_{\mathbf{S}|\mathbf{G}}\mathbf{S} = \mathbf{S}$, and $\mathbf{E}_{\mathbf{S}|\mathbf{G}}\mathbf{G} = \mathbf{0}$.

$i = 1, \ldots, Q$, $i \neq q$) remain unchanged, and the calculation of $\mathbf{w}_{\text{new}}$ need not be completely reconducted. Therefore, the above-mentioned algorithm is flexible in the adjustment of array responses, especially when only some desired levels of the preassigned angles vary. More benefits of the resulting $\mathbf{w}_{\text{new}}$ in (20) will be discussed later. Finally, to make the above-mentioned algorithm clear, we summarize the main steps in Algorithm 1.

B. Second Algorithm

In Section III-A, a closed-form array response control algorithm is presented to adjust the response levels of multiple angles, with the aid of oblique projection. Analysis shows that the proposed algorithm can adjust the responses independently. However, the first algorithm may lead to beam axis shift and thus make the maximum response level appear at an angle other than θ_0. A similar problem was considered in [30]. To tackle this imperfection, the second array response control algorithm is developed by modifying the first one with a derivative constraint.

Again, we give the previous weight vector $\mathbf{w}_{\text{pre}}$ (satisfying $\mathbf{w}_{\text{pre}}^{\text{H}}\mathbf{a}(\theta_0) = 1$) and the desired level ρ_q at θ_q, $q = 1, \ldots, Q$. Then, to realize array response control without beam axis shift, the ultimate weight vector $\mathbf{w}_{\text{new}}$ should satisfy (8), and additionally

$$\theta_0 = \arg\max_{\theta} \left|\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta)\right|. \tag{22}$$

Hence, to ensure the new constraint (22) satisfied, the following derivative constraint [30] can be imposed:

$$\left.\frac{\partial P(\theta)}{\partial \theta}\right|_{\theta=\theta_0} = 0 \tag{23}$$

where $P(\theta) = \mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta)\mathbf{a}^{\text{H}}(\theta)\mathbf{w}_{\text{new}}$ denotes the array power response of $\mathbf{w}_{\text{new}}$, and θ_0 is the desired beam axis. Substituting $P(\theta)$ into (23) yields

$$\begin{aligned}\frac{\partial P(\theta)}{\partial \theta} &= \mathbf{w}_{\text{new}}^{\text{H}}\frac{\partial \mathbf{a}(\theta)}{\partial \theta}\mathbf{a}^{\text{H}}(\theta)\mathbf{w}_{\text{new}} + \mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta)\frac{\partial \mathbf{a}^{\text{H}}(\theta)}{\partial \theta}\mathbf{w}_{\text{new}}\\ &= 2\Re\left(\mathbf{w}_{\text{new}}^{\text{H}}\frac{\partial \mathbf{a}(\theta)}{\partial \theta}\mathbf{a}^{\text{H}}(\theta)\mathbf{w}_{\text{new}}\right)\end{aligned} \tag{24}$$

and further

$$\left.\frac{\partial P(\theta)}{\partial \theta}\right|_{\theta=\theta_0} = 2\Re\left(\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{d}(\theta_0)\mathbf{a}^{\text{H}}(\theta_0)\mathbf{w}_{\text{new}}\right) \tag{25}$$

where $\mathbf{d}(\theta_0)$ is defined as

$$\mathbf{d}(\theta_0) \triangleq \left.\frac{\partial \mathbf{a}(\theta)}{\partial \theta}\right|_{\theta=\theta_0}. \tag{26}$$

Then, to control the array response control as required in (8) without leading to the beam axis shift, the weight vector $\mathbf{w}_{\text{new}}$ should satisfy

$$\Re\left(\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{d}(\theta_0)\mathbf{a}^{\text{H}}(\theta_0)\mathbf{w}_{\text{new}}\right) = 0 \tag{27a}$$

$$\left|\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_q)\right|^2 \Big/ \left|\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_0)\right|^2 = \rho_q,\ q = 1, \ldots, Q. \tag{27b}$$

Following the first algorithm, the identities in (27) can be guaranteed by setting

$$\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_0) = 1 \tag{28a}$$

$$\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_q) = e^{j\phi_q}\mathbf{w}_{\text{new},q}^{\text{H}}\mathbf{a}(\theta_q), \quad q = 1, \ldots, Q \tag{28b}$$

$$\Re\left(\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{d}(\theta_0)\right) = 0 \tag{28c}$$

where $\phi_q \in \mathbb{R}$ can be arbitrarily specified. Note that the weight vector $\mathbf{w}_{\text{new},q}$ in (28b) is obtained with the WORD algorithm (see (10)).

To find a weight $\mathbf{w}_{\text{new}}$ satisfying (28), we first note that $\Re(\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{d}(\theta_0)) = \Re(\mathbf{w}_{\text{new}}^{\text{T}})\Re(\mathbf{d}(\theta_0)) + \Im(\mathbf{w}_{\text{new}}^{\text{T}})\Im(\mathbf{d}(\theta_0))$. Then, one can convert (28c) to its real domain as

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\widetilde{\mathbf{d}}(\theta_0) = 0 \tag{29}$$

where the tilde notation is defined as $\widetilde{\mathbf{x}} \triangleq [\Re(\mathbf{x}^{\text{T}}), \Im(\mathbf{x}^{\text{T}})]^{\text{T}}$. On the other hand, one can readily find that

$$\begin{aligned}&\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta)\\ &\quad= \left[\Re\left(\mathbf{w}_{\text{new}}^{\text{T}}\right), \Im\left(\mathbf{w}_{\text{new}}^{\text{T}}\right)\right]\left[\begin{pmatrix}\Re[\mathbf{a}(\theta)]\\ \Im[\mathbf{a}(\theta)]\end{pmatrix} + j\begin{pmatrix}\Im[\mathbf{a}(\theta)]\\ -\Re[\mathbf{a}(\theta)]\end{pmatrix}\right]\\ &\quad= \widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\left[\widetilde{\mathbf{a}}(\theta) + j\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta)\right]\end{aligned} \tag{30}$$

where the matrix $\boldsymbol{\Upsilon}$ is defined as

$$\boldsymbol{\Upsilon} \triangleq \begin{bmatrix} & \mathbf{I}_N\\ -\mathbf{I}_N & \end{bmatrix} \in \mathbb{R}^{2N\times 2N}. \tag{31}$$

According to (30), constraint (28a) can be reformulated as

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\widetilde{\mathbf{a}}(\theta_0) = 1, \quad \widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_0) = 0. \tag{32}$$

Moreover, by defining

$$\mathbf{u}_q \triangleq e^{-j\phi_q}\mathbf{w}_{\text{new},q} \tag{33}$$

and then reshaping (28b) as

$$\mathbf{w}_{\text{new}}^{\text{H}}\mathbf{a}(\theta_q) = (e^{-j\phi_q}\mathbf{w}_{\text{new},q})^{\text{H}}\mathbf{a}(\theta_q) = \mathbf{u}_q^{\text{H}}\mathbf{a}(\theta_q) \tag{34}$$

one can rewrite the constraint (28b) as

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\widetilde{\mathbf{a}}(\theta_q) = \widetilde{\mathbf{u}}_q^{\text{T}}\widetilde{\mathbf{a}}(\theta_q), \quad \widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_q) = \widetilde{\mathbf{u}}_q^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_q) \tag{35}$$

where the index q can be taken from 1 to Q.

Combining (29), (32), and (35), the identities in (28) can be equivalently expressed as

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\widetilde{\mathbf{a}}(\theta_0) = 1 \tag{36a}$$

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_0) = 0 \tag{36b}$$

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\widetilde{\mathbf{a}}(\theta_q) = \widetilde{\mathbf{u}}_q^{\text{T}}\widetilde{\mathbf{a}}(\theta_q), \quad q = 1, \ldots, Q \tag{36c}$$

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_q) = \widetilde{\mathbf{u}}_q^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_q), \quad q = 1, \ldots, Q \tag{36d}$$

$$\widetilde{\mathbf{w}}_{\text{new}}^{\text{T}}\widetilde{\mathbf{d}}(\theta_0) = 0 \tag{36e}$$

or compactly

$$\check{\mathbf{C}}^{\text{T}}\widetilde{\mathbf{w}}_{\text{new}} = \mathbf{g} \tag{37}$$

where $\check{\mathbf{C}} \in \mathbb{R}^{2N\times(2Q+3)}$ and $\mathbf{g} \in \mathbb{R}^{2Q+3}$ are, respectively, detailed as

$$\check{\mathbf{C}} \triangleq [\mathbf{Y}(\theta_0), \mathbf{Y}(\theta_1), \ldots, \mathbf{Y}(\theta_Q), \widetilde{\mathbf{d}}(\theta_0)]$$

$$\mathbf{g} \triangleq \left[1, 0, \widetilde{\mathbf{u}}_1^{\text{T}}\widetilde{\mathbf{a}}(\theta_1), \widetilde{\mathbf{u}}_1^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_1), \ldots, \widetilde{\mathbf{u}}_Q^{\text{T}}\widetilde{\mathbf{a}}(\theta_Q), \widetilde{\mathbf{u}}_Q^{\text{T}}\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_Q), 0\right]^{\text{T}}$$

with $\mathbf{Y}(\theta)$ being defined by

$$\mathbf{Y}(\theta) \triangleq [\widetilde{\mathbf{a}}(\theta),\ \boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta)] \in \mathbb{R}^{2N\times 2}. \tag{38}$$

Once a weight vector $\widetilde{\mathbf{w}}_{\text{new}}$ satisfying (37) is obtained, we can reformulate it to complex domain and express the final weight vector $\mathbf{w}_{\text{new}}$ as

$$\mathbf{w}_{\text{new}} = [\mathbf{I}_N,\ j\mathbf{I}_N]\,\widetilde{\mathbf{w}}_{\text{new}}. \tag{39}$$

To find a closed-form $\widetilde{\mathbf{w}}_{\text{new}}$ satisfying (37), similar to the first algorithm, we first define a matrix $\breve{\mathbf{C}}_{i-} \in \mathbb{R}^{2N\times(2Q+1)}$, by removing the columns of $\widetilde{\mathbf{a}}(\theta_i)$ and $\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_i)$ from $\breve{\mathbf{C}}$, $i = 0, 1, \ldots, Q$, as

$$\breve{\mathbf{C}}_{i-} \triangleq \left[\mathbf{Y}(\theta_0), \ldots, \mathbf{Y}(\theta_{i-1}), \mathbf{Y}(\theta_{i+1}), \ldots, \mathbf{Y}(\theta_Q), \widetilde{\mathbf{d}}(\theta_0)\right].$$

Then, we define

$$\mathbf{Z}_0 \triangleq \mathbf{I} - \mathbf{E}^{\mathrm{T}}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)} \tag{40a}$$

$$\mathbf{Z}_q \triangleq \mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_q)|\breve{\mathbf{C}}_{q-}},\quad q = 1, \ldots, Q \tag{40b}$$

where $\mathbf{E}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)}$ and $\mathbf{E}_{\mathbf{Y}(\theta_q)|\breve{\mathbf{C}}_{q-}}$ are the oblique projectors as

$$\begin{aligned}\mathbf{E}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)} &= \breve{\mathbf{C}}_{0-}\big(\breve{\mathbf{C}}^{\mathrm{H}}_{0-}\mathbf{P}^{\perp}_{\mathbf{Y}(\theta_0)}\breve{\mathbf{C}}_{0-}\big)^{-1}\breve{\mathbf{C}}^{\mathrm{H}}_{0-}\mathbf{P}^{\perp}_{\mathbf{Y}(\theta_0)}\\ \mathbf{E}_{\mathbf{Y}(\theta_q)|\breve{\mathbf{C}}_{q-}} &= \mathbf{Y}(\theta_q)\big(\mathbf{Y}^{\mathrm{H}}(\theta_q)\mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{q-}}\mathbf{Y}(\theta_q)\big)^{-1}\mathbf{Y}^{\mathrm{H}}(\theta_q)\mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{q-}}\end{aligned}$$

with $q = 1, \ldots, Q$. Recalling the property of oblique projector, one can readily learn that

$$\mathbf{Z}_i^{\mathrm{T}}\mathbf{Y}(\theta_j) = \begin{cases}\mathbf{Y}(\theta_j), & \text{if } j = i\\ \mathbf{0}, & \text{if } j \neq i\end{cases} \tag{41a}$$

$$\mathbf{Z}_i^{\mathrm{T}}\widetilde{\mathbf{d}}(\theta_0) = \mathbf{0} \tag{41b}$$

where i and j can be taken as $0, 1, \ldots, Q$. Consequently, following a similar concept to the first algorithm, an analytical expression of $\widetilde{\mathbf{w}}_{\text{new}}$ can be given by

$$\widetilde{\mathbf{w}}_{\text{new}} = \mathbf{Z}_0\widetilde{\mathbf{w}}_{\text{pre}} + \sum_{q=1}^{Q}\mathbf{Z}_q\widetilde{\mathbf{u}}_q. \tag{42}$$

It can be readily verified that

$$\begin{aligned}\widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{new}}(\widetilde{\mathbf{a}}(\theta_0) + j\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_0)) &= \widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{pre}}\mathbf{Z}_0^{\mathrm{T}}(\widetilde{\mathbf{a}}(\theta_0) + j\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_0))\\ &= \widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{pre}}\mathbf{Z}_0^{\mathrm{T}}\mathbf{Y}(\theta_0)[1,\ j]^{\mathrm{T}}\\ &= \widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{pre}}\mathbf{Y}(\theta_0)[1,\ j]^{\mathrm{T}}\\ &= \widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{pre}}\mathbf{a}(\theta_0) + j\widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{pre}}\boldsymbol{\Upsilon}\mathbf{a}(\theta_0)\\ &= \mathbf{w}^{\mathrm{H}}_{\text{pre}}\mathbf{a}(\theta_0)\\ &= 1\end{aligned} \tag{43a}$$

$$\begin{aligned}\widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{new}}(\widetilde{\mathbf{a}}(\theta_q) + j\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_q)) &= \sum_{i=1}^{Q}\widetilde{\mathbf{u}}_i^{\mathrm{T}}\mathbf{Z}_i^{\mathrm{T}}\mathbf{Y}(\theta_q)[1,\ j]^{\mathrm{T}}\\ &= \widetilde{\mathbf{u}}_q^{\mathrm{T}}\mathbf{Y}(\theta_q)[1,\ j]^{\mathrm{T}}\\ &= \widetilde{\mathbf{u}}_q^{\mathrm{T}}(\widetilde{\mathbf{a}}(\theta_q) + j\boldsymbol{\Upsilon}\widetilde{\mathbf{a}}(\theta_q))\end{aligned} \tag{43b}$$

$$\begin{aligned}\widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{new}}\widetilde{\mathbf{d}}(\theta_0) &= \widetilde{\mathbf{w}}^{\mathrm{T}}_{\text{pre}}\mathbf{Z}_0^{\mathrm{T}}\widetilde{\mathbf{d}}(\theta_0) + \sum_{q=1}^{Q}\widetilde{\mathbf{u}}_q^{\mathrm{T}}\mathbf{Z}_q^{\mathrm{T}}\widetilde{\mathbf{d}}(\theta_0)\\ &= 0.\end{aligned} \tag{43c}$$

Algorithm 2 Second Algorithm

```
1: give the previous weight vector $\mathbf{w}_{\text{pre}}$, $\theta_q$ and its desired
   level $\rho_q$, calculate $\mathbf{T}_0$ and $\mathbf{T}_q$ by (16), $q = 1, \ldots, Q$
2: **for** $q = 1, \ldots, Q$ **do**
3:     calculate $\beta_q$ via WORD algorithm such that
       $L_{\text{new}}(\theta_q, \theta_0) = \rho_q$
4:     obtain $\breve{\mathbf{w}}_{\text{new},q} = [\mathbf{w}_{q,\perp},\ \mathbf{w}_{q,\parallel}][1,\ \beta_q]^{\mathrm{T}}$
5:     obtain $\mathbf{w}_{\text{new},q} = \breve{\mathbf{w}}_{\text{new},q}/(\mathbf{a}^{\mathrm{H}}(\theta_0)\breve{\mathbf{w}}_{\text{new},q})$
6: **end for**
7: calculate the weight vector $\widetilde{\mathbf{w}}_{\text{new}}$ by (42), where $\phi_q$ in $\mathbf{u}_q$,
   $q = 1, \ldots, Q$, can be arbitrarily specified
8: obtain the weight vector $\mathbf{w}_{\text{new}} = [\mathbf{I}_N,\ j\mathbf{I}_N]\,\widetilde{\mathbf{w}}_{\text{new}}$
```

This shows that the resulting $\widetilde{\mathbf{w}}_{\text{new}}$ in (42) satisfies (37), or equivalently, the corresponding $\mathbf{w}_{\text{new}}$ in (39) satisfies the identities in (28). Thus, the new weight vector $\mathbf{w}_{\text{new}}$ in (39) realizes the preassigned multipoint array response control (8) without leading to the beam axis shift. Similar to the first algorithm, $\mathbf{w}_{\text{new}}$ in (39) is able to adjust the response of θ_q separately, by simply renewing the corresponding weight $\mathbf{w}_{\text{new},q}$, $q = 1, \ldots, Q$. In this manner, the responses at other controlled points (i.e., θ_is, $i = 1, \ldots, Q$, $i \neq q$) remain unchanged, and the calculation of $\mathbf{w}_{\text{new}}$ need not be completely reconducted. Since an additional constraint is imposed in the above-mentioned algorithm, we know that at most $N-2$ angles can be jointly adjusted in this approach. Finally, we summarize the main steps of the second algorithm in Algorithm 2.

Remark 1: In the above-presented algorithms, the parameters φ_q and ϕ_q, $q = 1, \ldots, Q$, can be arbitrarily specified. Basically, how to further determine φ_q or ϕ_q relies on specific considerations and applications. We found through extensive experiments that good performance (e.g., small beampattern variation at uncontrolled region) can be obtained by simply setting $\varphi_q = 0$ or $\phi_q = 0$ for $q = 1, \ldots, Q$. With this setting, the phase outputs of beamformer at the controlled angles θ_qs also remain unchanged, $q = 1, \ldots, Q$.

C. Properties of the Proposed Algorithms

In this section, we shall show some interesting properties of the above-mentioned two algorithms. To begin with, we note that the component in $\mathcal{R}^{\perp}(\breve{\mathbf{A}})$ has no contribution to the array response levels at θ_i, $i = 0, 1, \ldots, Q$, but may affect the responses outside the controlled points. Then, compared to the previous weight $\mathbf{w}_{\text{pre}}$, a weight vector $\mathbf{w}_{\text{new}}$ that brings no redundancy (i.e., component in $\mathcal{R}^{\perp}(\breve{\mathbf{A}})$) should satisfy $\mathbf{w}_{\text{new}} - \mathbf{w}_{\text{pre}} \in \mathcal{R}(\breve{\mathbf{A}})$, or equivalently

$$\mathbf{P}^{\perp}_{\breve{\mathbf{A}}}(\mathbf{w}_{\text{new}} - \mathbf{w}_{\text{pre}}) = \mathbf{0}. \tag{44}$$

Interestingly, both the resulting two weight vectors in (20) and (39) satisfy (44). To see this, we can rewrite (see Appendix A for details) $\mathbf{w}_{\text{new}}$ in (20) as

$$\mathbf{w}_{\text{new}} = \mathbf{P}^{\perp}_{\breve{\mathbf{A}}}\mathbf{w}_{\text{pre}} + \breve{\mathbf{A}}\mathbf{H}\mathbf{c} \tag{45}$$

where $\mathbf{H}$ and $\mathbf{c}$ are defined in (59) and (60) in Appendix A, respectively. From (45), it can be readily validated that (44) holds true.

On the other hand, to show that the resulting $\mathbf{w}_{\rm new}$ in (39) also satisfies (44), we first reexpress $\widetilde{\mathbf{w}}_{\rm new}$ in (42) as

$$\widetilde{\mathbf{w}}_{\rm new} = \mathbf{P}^{\perp}_{\check{\mathbf{C}}}\widetilde{\mathbf{w}}_{\rm pre} + \check{\mathbf{C}}\mathbf{F}\boldsymbol{\alpha}. \tag{46}$$

The derivation details of (46) can be found in Appendix B, where the definitions of $\mathbf{F}$ and $\boldsymbol{\alpha}$ have been specified in (67). From (46), one can readily verify that

$$\mathbf{P}^{\perp}_{\check{\mathbf{C}}}(\widetilde{\mathbf{w}}_{\rm new} - \widetilde{\mathbf{w}}_{\rm pre}) = \mathbf{0}. \tag{47}$$

On this basis, it is derived in Appendix C that the reconstructed weight vector $\mathbf{w}_{\rm new}$ from $\widetilde{\mathbf{w}}_{\rm new}$ as (39) satisfies

$$\mathbf{P}^{\perp}_{[\check{\mathbf{A}},\ \mathbf{d}(\theta_0)]}(\mathbf{w}_{\rm new} - \mathbf{w}_{\rm pre}) = \mathbf{0}. \tag{48}$$

Since $\mathcal{R}^{\perp}([\check{\mathbf{A}},\ \mathbf{d}(\theta_0)]) \in \mathcal{R}^{\perp}(\check{\mathbf{A}})$, it is not difficult to learn that (44) holds true.

In addition, it is interesting to note that when taking $\rho_q = L_{\rm pre}(\theta_q,\theta_0) \triangleq |\mathbf{w}_{\rm pre}^{\rm H}\mathbf{a}(\theta_q)|^2/|\mathbf{w}_{\rm pre}^{\rm H}\mathbf{a}(\theta_0)|^2$ in (8) and $\varphi_q = 0$ in (20), $q = 1,\ldots,Q$, the resulting $\mathbf{w}_{\rm new}$ in (20) satisfies

$$\mathbf{w}_{\rm new} = \mathbf{w}_{\rm pre}. \tag{49}$$

The derivation of (49) can be found in Appendix D. This is consistent with the fact the weight vector should be unchanged if the response at θ_q is not adjusted.

Similarly, in the second algorithm, if $\rho_q = L_{\rm pre}(\theta_q,\theta_0)$ in (8) and $\phi_q = 0$ in (28b), $q = 1,\ldots,Q$, then the vector $\widetilde{\mathbf{w}}_{\rm new}$ in (42) satisfies (see Appendix E for the derivation details)

$$\widetilde{\mathbf{w}}_{\rm new} = \big(\mathbf{I} - \mathbf{E}^{\rm T}_{\widetilde{\mathbf{d}}(\theta_0)|\check{\mathbf{C}}_{\widetilde{\mathbf{d}}-}}\big)\widetilde{\mathbf{w}}_{\rm pre} \tag{50}$$

where

$$\check{\mathbf{C}}_{\widetilde{\mathbf{d}}-} \triangleq [\mathbf{Y}(\theta_0), \mathbf{Y}(\theta_1), \ldots, \mathbf{Y}(\theta_Q)] \in \mathbb{R}^{2N\times(2Q+2)}. \tag{51}$$

In this case, the weight vector $\mathbf{w}_{\rm new}$ [obtained with (39)] steers the beam axis to θ_0, with the response levels at θ_qs ($q = 1,\ldots,Q$) remaining unchanged. In fact, (50) provides a flexible manner to rechange the beam axis or scan the beampattern, with some fixed constraints on array response. Note that if no specific constraints are required on the array response levels, we can simply set $\check{\mathbf{C}}_{\widetilde{\mathbf{d}}-} = \mathbf{Y}(\theta_0)$ and take the new weight as

$$\mathbf{w}_{\rm new} = [\mathbf{I}_N,\ j\mathbf{I}_N]\big(\mathbf{I} - \mathbf{E}^{\rm T}_{\widetilde{\mathbf{d}}(\theta_0)|\mathbf{Y}(\theta_0)}\big)\widetilde{\mathbf{w}}_{\rm pre}. \tag{52}$$

In this case, the resulting weight vector $\mathbf{w}_{\rm new}$ refocuses the beam axis to θ_0, with possible less variations compared to the beampattern of the previous weight.

D. Computational Complexity and Advantage Summary

In this section, we analyze the computational complexities of the proposed two algorithms. Both of our algorithms provide closed-form expressions for the ultimate weight vectors [see (20) and (39)], and their main computations lie in the calculations of oblique projectors. For the first algorithm, the computational complexity of each oblique projector (i.e., $\mathbf{E}_{\check{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)}$ or $\mathbf{E}_{\mathbf{a}(\theta_q)|\check{\mathbf{A}}_{q-}}$, $q = 1,\ldots,Q$) is $\mathcal{O}(Q^3)$. Thus, the overall computational complexity of the first algorithm is $\mathcal{O}(Q^4)$. Similarly, in the second algorithm, the calculation of the oblique projector (i.e., $\mathbf{E}_{\check{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)}$ or $\mathbf{E}_{\mathbf{Y}(\theta_q)|\check{\mathbf{C}}_{q-}}$, $q = 1,\ldots,Q$) is $\mathcal{O}(Q^3)$. Thus, the computational complexity of the second algorithm is also $\mathcal{O}(Q^4)$. Note that the second algorithm may cost more space complexity compared to the first one, and both of the algorithms can control the responses without complete recalculation, whenever some of the desired levels require adjustments.

Algorithm 3 Proposed Pattern Synthesis Algorithm

```
give $\theta_0$, the desired pattern $L_d(\theta)$, the initial weight vector $\mathbf{w}_0$ and its corresponding response pattern $L_0(\theta,\theta_0)$, set $k = 1$
while 1 do
    select $Q_k$ angles by comparing $L_{k-1}(\theta,\theta_0)$ with $L_d(\theta)$
    apply the proposed algorithm (see Algorithm 1 or Algorithm 2) to realize $L_k(\theta_q,\theta_0) = L_d(\theta_q)$, $q = 1,\ldots,Q_k$, obtain $\mathbf{w}_k$ and the corresponding $L_k(\theta,\theta_0)$
    if $L_k(\theta,\theta_0)$ is not satisfactory then
        set $k = k + 1$
    else
        break
    end if
end while
output $\mathbf{w}_k$ and $L_k(\theta,\theta_0)$
```

To summarize, the proposed two algorithms have the following advantages:

1) They can precisely control the array responses at multiple points starting from an arbitrarily given weight vector, and the second algorithm realizes this task without leading to beam axis shift.
2) They provide closed-form expressions and thus are computationally attractive and convenient to implement.
3) They are able to control the array responses of multiple angles separately. Hence, we do not need to completely reconduct the weight calculation whenever some desired levels are changed.
4) They can achieve array responses control without causing much beampattern variations at the uncontrolled regions.

IV. Pattern Synthesis Using the Proposed Algorithms

In this section, the application of the proposed algorithms to pattern synthesis is briefly introduced. Generally, the strategy herein shares a similar concept of pattern synthesis using MA^{2}RC [30] or FARCOP [31]. More specifically, set $k = 1$ and give an initial weight vector $\mathbf{w}_0$ that can be freely configured. Multiple angles are first determined according to $L_{k-1}(\theta,\theta_0)$ (standing for the array response pattern of $\mathbf{w}_{k-1}$), and the desired pattern, denoted as $L_d(\theta)$. Following the angle selection strategy in [20], for the sidelobe synthesis, we select Q_k ($Q_k \le N-1$) peak angles where the response differences (from the desired levels) are relatively large. For the main lobe synthesis, few discrete angles, where the responses deviate large from the desired ones, are chosen. Once those angles are picked out, the proposed algorithms can be utilized to find the weight $\mathbf{w}_k$ by adjusting the corresponding responses to their desired values. Then, we set $k = k + 1$ and repeat the above-mentioned procedure until the response pattern is satisfactorily synthesized.

Note that for a desired pattern with narrow beam (no specific shape constraints on the main lobe region), we apply the second algorithm to avoid beam axis shift. As for a desired pattern with lower-bound constraints on its main lobe, e.g., beampattern with a flat-top main lobe, we can use the first algorithm since the beam axis shift problem is less critical in this case. In addition, the number of the selected angles (denoted as Q_k) can be different in each step, and it should satisfy $Q_k \le N - 1$ for the first algorithm and $Q_k \le N - 2$ for the second one. Finally, we summarize the proposed pattern synthesis algorithm in Algorithm 3.

V. Numerical Results

In this section, the effectiveness and flexibility of the proposed algorithms for multipoint control are first shown. Then, representative numerical examples are conducted to demonstrate their superiorities in pattern synthesis under different situations. As mentioned earlier, we simply set $\varphi_q = \phi_q = 0$, $q = 1, \dots, Q$, in all simulations. For comparison, the results of convex programming (CP) method in [23], SDR method in [24], Philip's method in [18], WORD algorithm in [29], MA^{2}RC and M^{2}A^{2}RC in [30], and the FARCOP algorithm in [31] will also be examined if applicable.

A. Illustrations of the Proposed Algorithms

1) Response Control for ULA: In the first example, a linearly half-wavelength-spaced array with 16 isotropic elements is considered. The beam axis is taken as $\theta_0 = 20°$. We set the previous weight vector $\mathbf{w}_{\text{pre}}$ as the Chebyshev weight with a -25 dB of sidelobe attenuation, and prescribe three angles, i.e., $\theta_1 = -60°$, $\theta_2 = -36°$, and $\theta_3 = -12°$. In the first case, we take the corresponding desired levels as $\rho_1 = \rho_2 = \rho_3 = -40$ dB. Accordingly, one can calculate that $\beta_1 = 0.1878$, $\beta_2 = 0.1788$, and $\beta_3 = 0.1772$. Fig. 1(a) shows the resulting beampattern of various algorithms. One can see that all the methods have realized the preassigned array response control task, and the obtained beampatterns are similar. After zooming in the main lobe region, it can be found that MA^{2}RC, FARCOP, and the first proposed algorithm lead to the undesirable beam axis shifts on the beampatterns. For the second algorithm, the beam axis is unshifted at θ_0 and its effectiveness can be verified. In addition, we notice that the responses of the uncontrolled angles are almost unchanged for the proposed two algorithms. We have specified the resulting weights of our algorithms in Table I.

In the second case, we take the desired levels as $\rho_1 = -35$ dB, $\rho_2 = -40$ dB, and $\rho_3 = -20$ dB. In this scenario, we have $\beta_1 = 0.3341$, $\beta_2 = 0.1788$, and $\beta_3 = 1.7849$. Fig. 1(b) depicts the obtained beampatterns and Table II provides the resulting weights of our two algorithms. Again, the proposed two algorithms fulfill the given array response task and the second one does not bring a beam axis shift to its resulting pattern. Compared to the first case, it should be noted that we have altered ρ_1 and ρ_3 and have kept ρ_2 unvaried. For this reason, it is only required to recalculate the corresponding $\mathbf{w}_{\text{new},1}$ and $\mathbf{w}_{\text{new},3}$ in our algorithms and

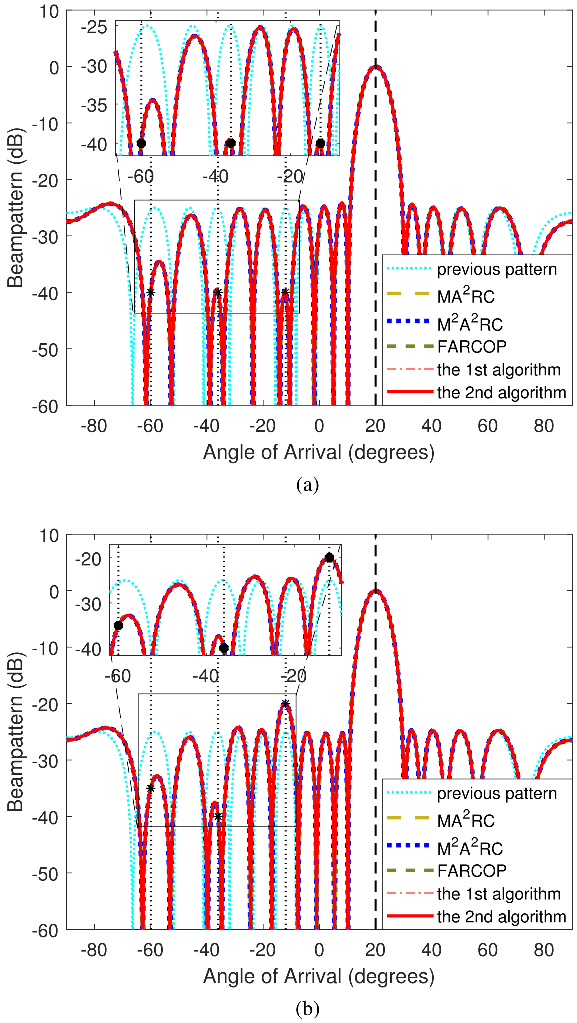


Fig. 1. Illustration of the proposed algorithms on array response control. (a) First case. (b) Second case.

TABLE I

Obtained Weights of Array Response Control (the First Case)

n	w_n (the 1st algorithm)	w_n (the 2nd algorithm)
1	$0.0359e^{-j0.0355}$	$0.0359e^{-j0.0368}$
2	$0.0397e^{+j1.1253}$	$0.0397e^{+j1.1243}$
3	$0.0464e^{+j2.1787}$	$0.0464e^{+j2.1779}$
4	$0.0602e^{-j3.1195}$	$0.0602e^{-j3.1200}$
5	$0.0664e^{-j1.9809}$	$0.0664e^{-j1.9812}$
6	$0.0783e^{-j0.8513}$	$0.0783e^{-j0.8515}$
7	$0.0880e^{+j0.1043}$	$0.0880e^{+j0.1042}$
8	$0.0856e^{+j1.2699}$	$0.0856e^{+j1.2698}$
9	$0.0856e^{+j2.2811}$	$0.0856e^{+j2.2811}$
10	$0.0880e^{-j2.8366}$	$0.0880e^{-j2.8365}$
11	$0.0783e^{-j1.8809}$	$0.0783e^{-j1.8807}$
12	$0.0664e^{-j0.7514}$	$0.0664e^{-j0.7510}$
13	$0.0602e^{+j0.3873}$	$0.0602e^{+j0.3877}$
14	$0.0464e^{+j1.3722}$	$0.0464e^{+j1.3730}$
15	$0.0397e^{+j2.4257}$	$0.0397e^{+j2.4267}$
16	$0.0359e^{-j2.6967}$	$0.0359e^{-j2.6954}$

TABLE II

OBTAINED WEIGHTS OF ARRAY RESPONSE CONTROL (THE SECOND CASE)

n	w_n (the 1st algorithm)	w_n (the 2nd algorithm)
1	$0.0408e^{+j0.0227}$	$0.0408e^{+j0.0294}$
2	$0.0405e^{+j0.9910}$	$0.0405e^{+j0.9966}$
3	$0.0403e^{+j2.1670}$	$0.0403e^{+j2.1722}$
4	$0.0602e^{-j3.0218}$	$0.0602e^{-j3.0192}$
5	$0.0713e^{-j2.0024}$	$0.0713e^{-j2.0005}$
6	$0.0758e^{-j0.9173}$	$0.0758e^{-j0.9162}$
7	$0.0825e^{+j0.1368}$	$0.0825e^{+j0.1375}$
8	$0.0891e^{+j1.3154}$	$0.0891e^{+j1.3157}$
9	$0.0891e^{+j2.2355}$	$0.0891e^{+j2.2352}$
10	$0.0825e^{-j2.8690}$	$0.0825e^{-j2.8697}$
11	$0.0758e^{-j1.8150}$	$0.0758e^{-j1.8160}$
12	$0.0713e^{-j0.7298}$	$0.0713e^{-j0.7317}$
13	$0.0602e^{+j0.2896}$	$0.0602e^{+j0.2896}$
14	$0.0403e^{+j1.3840}$	$0.0403e^{+j1.3788}$
15	$0.0405e^{+j2.5599}$	$0.0405e^{+j2.5544}$
16	$0.0408e^{-j2.7549}$	$0.0408e^{-j2.7616}$

TABLE III

PARAMETERS OF THE NONISOTROPIC LINEAR RANDOM ARRAY

n	$x_n(\lambda)$	$l_n(\lambda)$	ζ_n(deg)	n	$x_n(\lambda)$	$l_n(\lambda)$	ζ_n(deg)
1	0.00	0.3	0.0	6	2.64	0.27	10
2	0.45	0.25	-4.0	7	3.09	0.23	1.0
3	0.93	0.24	5.0	8	3.55	0.24	-10
4	1.56	0.20	-32	9	4.09	0.25	0.5
5	2.04	0.26	-3.2	10	4.52	0.21	7.2

then readily renew the ultimate weight vectors. One can see that the array responses can be separately adjusted by the devised two algorithms, while MA^{2}RC and M^{2}A^{2}RC have no such ability, although they obtain similar beampatterns to those of our algorithms in the above-mentioned testings. Note again that the responses of the uncontrolled-angles are almost unchanged for our algorithms.

2) Response Control for Nonisotropic Random Array: In this example, we consider a ten-element nonisotropic linear random array (see [16] and [28]). The pattern of the nth element is given by

$$g_n(\theta) = (\cos(\pi l_n \sin(\theta + \zeta_n)) - \cos(\pi l_n))/\cos(\theta + \zeta_n) \quad (53)$$

where ζ_n and l_n represent the orientation and length of the element, respectively. More details of the array can be found in Table III, where the element positions (in wavelength) are also specified. In this example, we set $\theta_0 = 0°$ and take $\mathbf{w}_{\text{pre}} = \mathbf{a}(\theta_0)$. It is expected to adjust the response levels of $\theta_1 = -25°$ and $\theta_2 = -4°$ to be $\rho_1 = -20$ and $\rho_2 = 0\,\text{dB}$, respectively. Note that θ_1 is located in the sidelobe region and θ_2 is inside the main lobe region.

The results of various methods are shown in Fig. 2, from which we find that the obtained beampatterns of MA^{2}RC and M^{2}A^{2}RC have been seriously distorted in the main lobe region. When testing our first algorithm, one can see that it realizes the predescribed array response task and obtains a similar beampattern to that of the FARCOP algorithm. Nevertheless, both of these two algorithms bring beam axis shifts to the resulting beampatterns. As for our second algorithm, it overcomes this drawback and obtains a qualified pattern, as shown in Fig. 2. The resulting weights of our algorithms have been specified in Table IV.

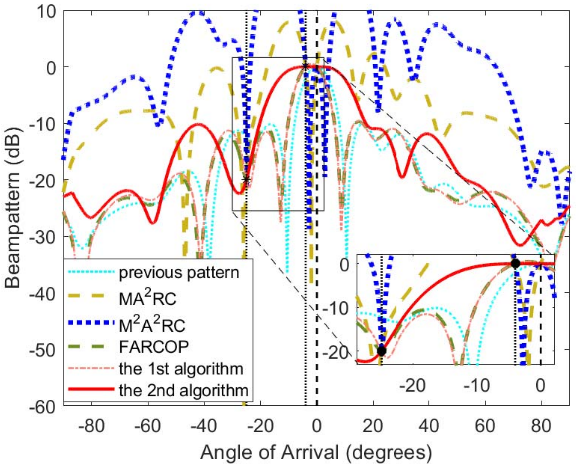


Fig. 2. Simulation results of multipoint responses control for a nonisotropic nonuniform random array.

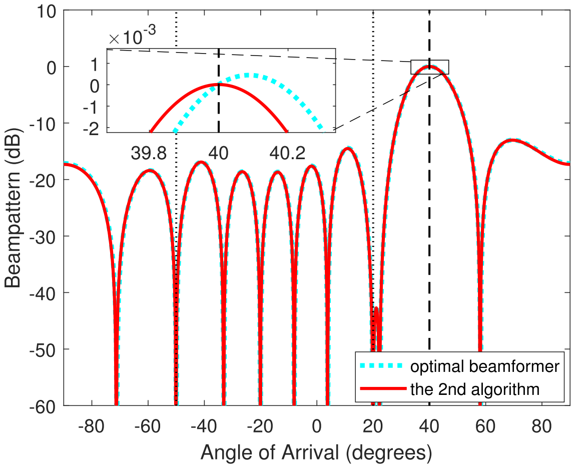
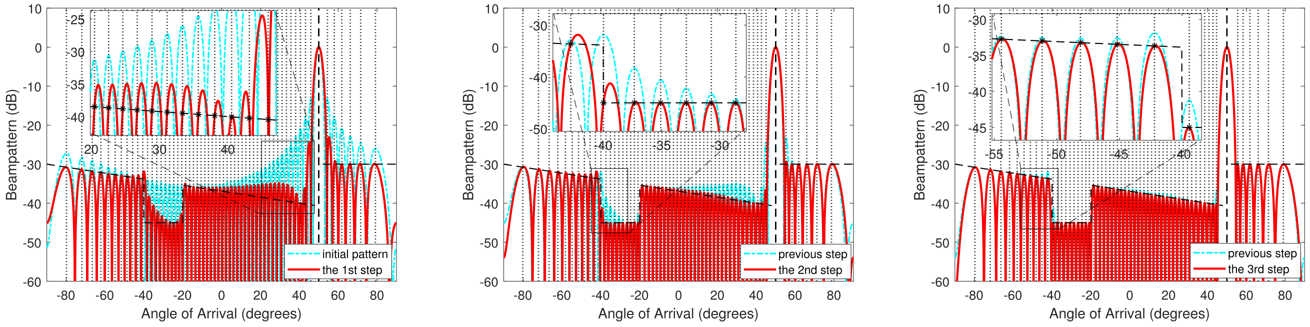


Fig. 3. Illustration of the second algorithm on beampattern refocus.

3) Beampattern Refocus for the Optimal Beamformer: In this section, we consider a ten-element uniform linear array (ULA) and present the application of the second algorithm on beampattern refocusing. More specifically, we take $\theta_0 = 40°$ and assume two interferences impinging from $-50°$ and $20°$, respectively. The interference-to-noise ratios (INRs) are set as 30 dB for both interferences, and the signal-to-noise ratio (SNR) is taken as 10 dB. In this case, we set the previous weight as the optimal beamformer, i.e., $\mathbf{w}_{\text{pre}} = \mathbf{w}_{\text{opt}} = \mathbf{R}_{n+i}^{-1}\mathbf{a}(\theta_0)$ (as detailed in Table V), where $\mathbf{R}_{n+i}$ is the noise-plus-interference covariance matrix and $\mathbf{w}_{\text{opt}}$ maximizes the signal-to-interference-plus-noise ratio (SINR).

The obtained beampattern of the optimal beamformer is depicted in Fig. 3, in which one can see that two nulls are formed at the directions of interferences. In addition, it can be observed that the resulting beam axis of the optimal beamformer has been shifted about 0.1° away the preassigned



(a) (b) (c)

Fig. 4. Intermediate results of pattern synthesis with nonuniform sidelobe. (a) Synthesized beampattern at the first step. (b) Synthesized beampattern at the second step. (c) Synthesized beampattern at the third step.

TABLE IV

OBTAINED WEIGHTS WHEN CONTROLLING ARRAY RESPONSES FOR A NONISOTROPIC NONUNIFORM RANDOM ARRAY

n	w_n (the 1st algorithm)	w_n (the 2nd algorithm)
1	$0.5031e^{+j0.3458}$	$0.1341e^{-j2.2553}$
2	$0.3542e^{+j0.4552}$	$0.2403e^{+j0.3682}$
3	$0.4043e^{+j0.4328}$	$0.6117e^{+j0.4319}$
4	$0.2542e^{+j0.1450}$	$0.5158e^{+j0.1478}$
5	$0.4813e^{-j0.1514}$	$1.1541e^{-j0.0600}$
6	$0.3782e^{-j0.2750}$	$0.8980e^{-j0.2463}$
7	$0.2713e^{-j0.1255}$	$0.4412e^{-j0.1242}$
8	$0.3365e^{-j0.1678}$	$0.2900e^{+j0.1860}$
9	$0.4065e^{-j0.4278}$	$0.0592e^{+j1.9237}$
10	$0.2549e^{-j0.7852}$	$0.4584e^{+j2.9935}$

TABLE V

SPECIFICATIONS OF THE OPTIMAL BEAMFORMER $\mathbf{w}_{\text{opt}}$ AND THE OBTAINED $\mathbf{w}_{\text{new}}$ AFTER BEAMPATTERN REFOCUS

n	the nth entry of $\mathbf{w}_{\text{opt}}$	the nth entry of $\mathbf{w}_{\text{new}}$
1	$0.9346e^{-j0.1781}$	$0.9322e^{-j0.1630}$
2	$0.7410e^{+j1.9812}$	$0.7407e^{+j1.9962}$
3	$0.8518e^{-j2.1060}$	$0.8529e^{-j2.0968}$
4	$1.0969e^{-j0.0092}$	$1.0980e^{-j0.0050}$
5	$1.1830e^{+j1.9076}$	$1.1831e^{+j1.9089}$
6	$1.1830e^{-j2.5827}$	$1.1831e^{-j2.5841}$
7	$1.0969e^{-j0.6660}$	$1.0980e^{-j0.6702}$
8	$0.8518e^{+j1.4308}$	$0.8529e^{+j1.4216}$
9	$0.7410e^{-j2.6564}$	$0.7407e^{-j2.6713}$
10	$0.9346e^{-j0.4971}$	$0.9322e^{-j0.5121}$

θ_0. Following the analysis in Section III-C, we refocus the optimal beampattern to θ_0 by calculating a new weight vector $\mathbf{w}_{\text{new}}$ using our second algorithm, more exactly, the formulation (52). The obtained weightings are specified in Table V and the corresponding result is shown with red line in Fig. 3, in which we find that the beam axis has been refocused to θ_0. Moreover, the two nulls at the directions of interferences are remained, and the obtained beampattern shape is almost the same as that of $\mathbf{w}_{\text{opt}}$.

B. Pattern Synthesis Using the Proposed Algorithms

In this section, we present the numerical results to validate the applicability and illustrate the performances of the proposed algorithms in pattern synthesis. For convenience, we set the initial weight vector as $\mathbf{w}_0 = \mathbf{a}(\theta_0)$ and only show one beampattern synthesis result of our proposed algorithms, according to the specification of desired pattern.

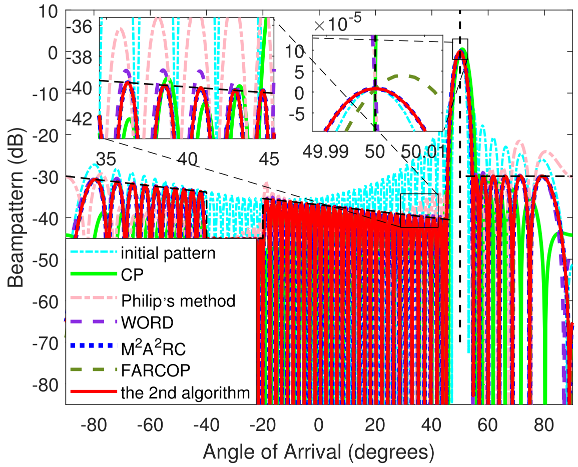


Fig. 5. Result comparison of nonuniform sidelobe synthesis.

1) Nonuniform Sidelobe Synthesis for ULA: In this example, the pattern synthesis for a linearly half-wavelength-spaced array with $N = 60$ isotropic elements is considered. The desired pattern steers at $\theta_0 = 50°$ with a nonuniform sidelobe level (see the black dashed lines in Fig. 4 for details).

Fig. 4 presents the intermediate synthesis results when using our second array response control algorithm. One can clearly see that, at each step of synthesis process, all sidelobe peaks of the previous pattern are selected, and then adjusted to their desired levels. Moreover, it only requires three steps to obtain a desirable beampattern, without leading to the beam axis shift. The resulting weights are specified in Table VI. The resulting comparison with other existing approaches is displayed in Fig. 5. It can be observed that the pattern envelopes of CP method, Philip's method, and the WORD method (after carrying out 100 iteration steps) are not aligned with the desired one. For WORD, it may require more steps to obtain a satisfactory pattern. Moreover, CP method, Philip's method, and the WORD method result unexpected beam axis shifts to their resulting beampatterns, either does the FARCOP algorithm as clearly shown in Fig. 5. The M^{2}A^{2}RC algorithm (also with three iteration steps) obtains a qualified sidelobe

TABLE VI
OBTAINED WEIGHTS IN NONUNIFORM SIDELOBE SYNTHESIS

n	w_n	n	w_n	n	w_n
1	$0.084e^{-j0.566}$	21	$0.226e^{-j2.096}$	41	$0.218e^{+j1.996}$
2	$0.058e^{+j2.002}$	22	$0.232e^{+j0.292}$	42	$0.206e^{-j1.888}$
3	$0.074e^{-j1.293}$	23	$0.242e^{+j2.700}$	43	$0.197e^{+j0.513}$
4	$0.050e^{+j1.104}$	24	$0.248e^{-j1.176}$	44	$0.187e^{+j2.913}$
5	$0.066e^{-j2.942}$	25	$0.253e^{+j1.230}$	45	$0.178e^{-j0.948}$
6	$0.073e^{-j0.503}$	26	$0.257e^{-j2.658}$	46	$0.163e^{+j1.448}$
7	$0.085e^{+j1.940}$	27	$0.264e^{-j0.253}$	47	$0.157e^{-j2.449}$
8	$0.085e^{-j1.948}$	28	$0.264e^{+j2.161}$	48	$0.145e^{-j0.008}$
9	$0.103e^{+j0.380}$	29	$0.267e^{-j1.731}$	49	$0.133e^{+j2.387}$
10	$0.114e^{+j2.864}$	30	$0.269e^{+j0.677}$	50	$0.121e^{-j1.489}$
11	$0.121e^{-j1.035}$	31	$0.269e^{+j3.082}$	51	$0.114e^{+j0.895}$
12	$0.133e^{+j1.373}$	32	$0.267e^{-j0.793}$	52	$0.103e^{-j2.904}$
13	$0.145e^{-j2.516}$	33	$0.264e^{+j1.598}$	53	$0.085e^{-j0.576}$
14	$0.157e^{-j0.075}$	34	$0.264e^{-j2.271}$	54	$0.085e^{+j1.820}$
15	$0.163e^{+j2.311}$	35	$0.257e^{+j0.134}$	55	$0.073e^{-j2.021}$
16	$0.178e^{-j1.576}$	36	$0.253e^{+j2.529}$	56	$0.066e^{+j0.418}$
17	$0.187e^{+j0.846}$	37	$0.248e^{-j1.348}$	57	$0.050e^{+j2.655}$
18	$0.197e^{-j3.037}$	38	$0.242e^{+j1.060}$	58	$0.074e^{-j1.231}$
19	$0.206e^{-j0.636}$	39	$0.232e^{-j2.816}$	59	$0.058e^{+j1.757}$
20	$0.218e^{+j1.763}$	40	$0.226e^{-j0.428}$	60	$0.084e^{-j1.958}$

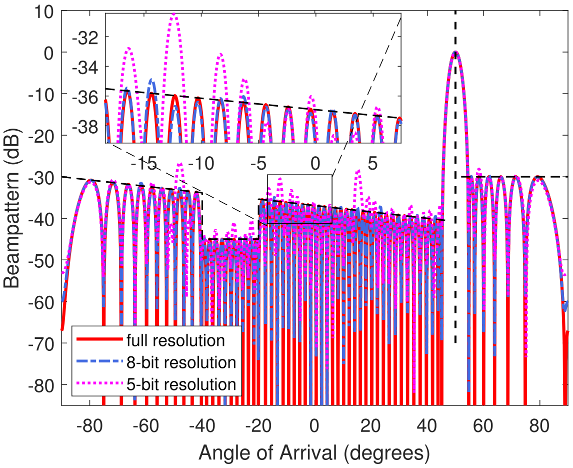


Fig. 6. Beampattern comparison with different phase shifter resolutions.

without shifting the beam axis. However, its runtime is longer than that of our second algorithm.

In the above-mentioned testing, the phase shifters are assumed to have infinite resolutions. In practice, however, the full-precision phase shifter is less practical, due to the hardware limitations. For this reason, we next consider low-resolution phase shifters with only finite phase states available and investigate the performance of our algorithm under this situation. More precisely, the resulting continuous phase values are quantized to the feasible set directly, and the runtime is almost unchanged compared to the infinite-resolution case. Fig. 6 shows the comparison of the full-resolution beampattern with those of 8 bit and 5 bit resolutions. It is seen that the 8 bit resolution phase shifters can provide comparable accuracy as the continuous ones. However, it is observed in Fig. 6 that, if 5 bit resolution phase shifters are used, the obtained envelope can still remain generally, but slight fluctuations at several angle sectors appear. Hence, an 8 bit resolution of phase weighting is a good candidate to achieve satisfactory accuracy and fulfill physical restrictions in practice.

TABLE VII
ELEMENT LOCATIONS OF NONUNIFORMLY SPACED LINEAR ARRAY AND WEIGHTS OBTAINED BY THE SECOND ALGORITHM

n	$x_n(\lambda)$	w_n	n	$x_n(\lambda)$	w_n
1	0.00	$0.0570e^{+j0.6370}$	9	3.99	$0.0094e^{-j2.2645}$
2	0.47	$0.0802e^{+j1.0907}$	10	4.48	$0.1297e^{+j2.0305}$
3	1.01	$0.0115e^{+j1.0213}$	11	4.96	$0.1275e^{+j2.5504}$
4	1.47	$0.1043e^{-j1.0534}$	12	5.43	$0.0135e^{-j2.8770}$
5	1.97	$0.1469e^{-j0.5261}$	13	5.94	$0.0853e^{+j0.4168}$
6	2.54	$0.0092e^{-j1.6467}$	14	6.49	$0.0699e^{+j1.0249}$
7	3.06	$0.1637e^{-j2.5912}$	15	6.98	$0.0103e^{-j2.7826}$
8	3.53	$0.1405e^{-j2.0673}$	16	7.46	$0.0440e^{-j1.2403}$

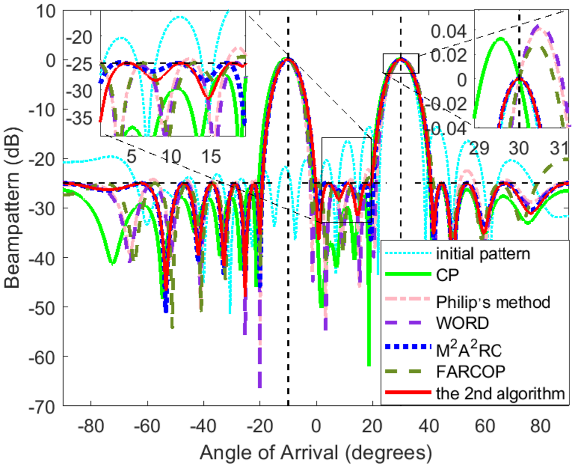


Fig. 7. Synthesized multibeam patterns for a nonuniformly spaced linear array.

2) Multibeam Pattern Synthesis for Nonuniformly Spaced Linear Array: In this example, multibeam pattern synthesis for the 16-element nonuniformly spaced linear array is considered. The element locations are given in Table VII, which follow the setting in [29]. The two beams steer at 30° and −10°, respectively. Here, we take $\mathbf{a}(30^\circ)$ as the initial weight and then apply the second algorithm to control the response at −10° to 0 dB, by setting $L_1(-10^\circ, 30^\circ) = 0$ dB. On this basis, the second algorithm is iteratively applied to adjust sidelobe response to be lower than −25 dB. After implementing four steps, a satisfactory pattern is obtained, and the corresponding weightings are listed in Table VII.

The synthesized patterns of CP, Philip's method, WORD (with 50 iteration steps), M^{2}A^{2}RC, FARCOP, and our second algorithm are shown in Fig. 7. To make a fair comparison, we have conducted M^{2}A^{2}RC and FARCOP the same iteration numbers as our algorithm. One can clearly see that CP, Philip's method, WORD, and FARCOP have brought serious shifts to their beam axes. Moreover, the pattern envelope of Philip's method is higher than the desired level at some regions, mainly due to its empirical parameter selection scheme. For our algorithm, the beam axis is unaltered and its execution time is shorter than that of M^{2}A^{2}RC.

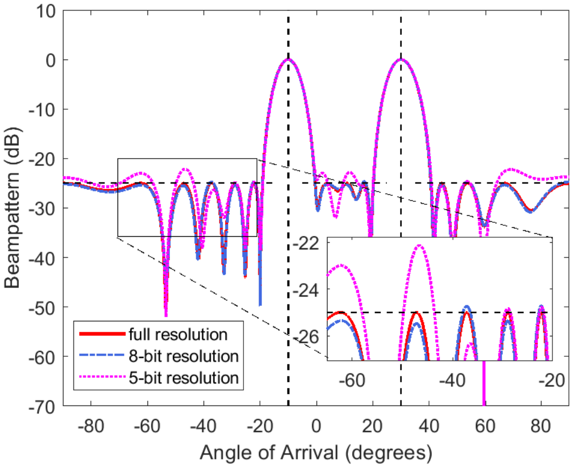


Fig. 8. Beampattern comparison with different phase shifter resolutions.

TABLE VIII
PARAMETERS OF THE NONISOTROPIC RANDOM ARRAY AND THE OBTAINED WEIGHTINGS BY THE FIRST ALGORITHM

n	$x_n(\lambda)$	$l_n(\lambda)$	ζ_n(deg)	w_n
1	0.00	0.30	0.0	$0.0605e^{+j2.9601}$
2	0.45	0.25	-4.0	$0.0380e^{+j2.3996}$
3	0.95	0.24	5.0	$0.0638e^{+j0.7497}$
4	1.50	0.20	-32	$0.1548e^{+j0.1279}$
5	2.04	0.26	-3.2	$0.0464e^{-j0.3445}$
6	2.64	0.27	10	$0.1397e^{+j2.4827}$
7	3.09	0.23	1.0	$0.1509e^{+j2.5651}$
8	3.55	0.24	-10	$0.1809e^{+j1.7285}$
9	4.05	0.25	0.0	$0.3635e^{+j0.2664}$
10	4.55	0.21	7.0	$0.7718e^{+j0.1308}$
11	5.06	0.20	5.0	$1.0000e^{+j0.1003}$
12	5.50	0.20	5.0	$0.5894e^{+j0.2047}$
13	6.01	0.29	4.0	$0.1912e^{+j0.6017}$
14	6.53	0.20	5.0	$0.2654e^{+j2.4105}$
15	7.07	0.26	-9.0	$0.1715e^{+j2.5461}$
16	7.52	0.21	7.0	$0.1617e^{+j3.1080}$
17	8.00	0.25	10	$0.0177e^{-j2.1015}$
18	8.47	0.21	6.0	$0.0214e^{+j0.5102}$
19	8.98	0.20	-8.0	$0.0249e^{+j0.1101}$
20	9.53	0.26	0.0	$0.0318e^{+j2.5806}$
21	10.01	0.25	5.0	$0.0262e^{+j0.6182}$

To further investigate the performance of our algorithm under the circumstance of finite phase resolution, we quantize the phase values and depict the resulting beampatterns with different resolutions shown in Fig. 8. It can be seen that the pattern with 8 bit resolution is almost the same as that of the full-resolution one. Although the result with 5 bit resolution brings some fluctuations at specific regions, the general envelope of the beampattern can still remain.

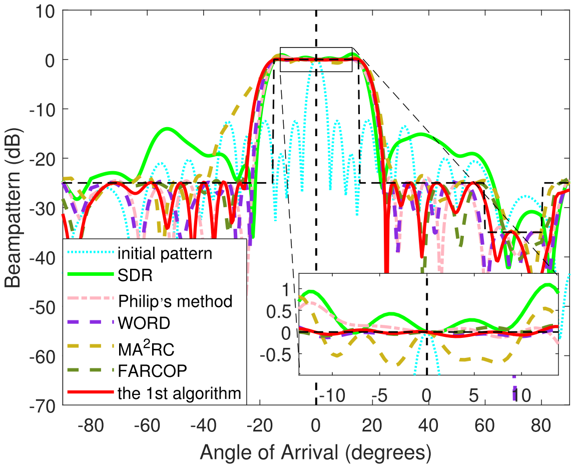


Fig. 9. Synthesized pattern with flat-top main lobe and broad-notch sidelobe for a nonisotropic random array.

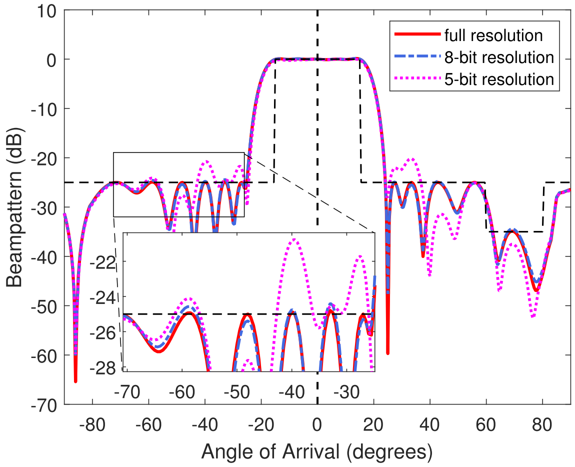


Fig. 10. Beampattern comparison with different phase shifter resolutions.

3) Pattern Synthesis With Constraints on Both Main Lobe and Sidelobe: To further examine the performance of the proposed method for pattern synthesis, a 21-element nonisotropic linear random array is considered. The pattern of the nth element is specified by $g_n(\theta)$ in (53). More details of the array can be found in Table VIII, where the element positions (in wavelength) are also specified. In this case, we steer the beam axis to $\theta_0 = 0°$. The desired pattern has a flat-top main lobe and nonuniform sidelobes. More specifically, all response levels in the main lobe $[\theta_0 - 15°, \theta_0 + 15°]$ are expected to be 0 dB. The upper bound level is -35 dB in the sidelobe region $[60°, 80°]$ and -25 dB in the rest of the sidelobe region. In this example, the first algorithm is applied since it is unnecessary to keep the beam axis unchanged in the flat-top scenario.

With the above-mentioned setting, the CP method is not applicable owing to the nonconvex lower bound constraint on beampattern. For this reason, the SDR method [24], which relaxes the nonconvex constraint into the convex one, is conducted and tested. The simulation results of SDR, Philip's method, WORD (with 450 iteration steps), MA^{2}RC, FARCOP, and the proposed algorithm (with all the later three approaches carrying out 200 iteration steps) are compared, as shown in Fig. 9. It is shown that the obtained sidelobe response of SDR does not satisfy the prescribed requirement due to the fact that the relaxation operator can only lead to an approximate solution. For Philip's method, the general shape of its beampattern looks good, with some sidelobes being higher than the desired values. On the other hand, we can see that the resulting beampattern of MA^{2}RC has been distorted. Since MA^{2}RC is a modified version of A^{2}RC in [28], the unsatisfactory result may be caused by the empirical parameter selection of A^{2}RC as discussed in [29]. For our proposed algorithm (the resulting weightings are listed in Table VIII), it obtains a main lobe with ripple about 0.1dB and qualified sidelobe levels, although the initial pattern is considerably different from the desired one.

To measure the performance of our algorithm with finite-resolution phase shifters, we depict the resulting beampatterns by quantizing the phase weightings to different finite sets. The comparison is presented in Fig. 10, from which we can see that the beampattern with 8 bit phase resolution is almost identical with the full-precision one. When testing the performance of our algorithm with 5 bit phase resolution, slight differences are observed between the obtained pattern and the full-resolution one. Nevertheless, in general, the shape of the former can be maintained as presented in Fig. 10.

VI. Conclusion

In this paper, we have presented two closed-from algorithms for multipoint array response control, with the aid of oblique projection operator. Both the proposed two algorithms work on the foundation of the WORD approach, and they can adjust array responses separately in very simple manners, with small pattern variations in the uncontrolled region. Our algorithms are able to separately control the array responses. In this manner, the weight calculation need not be completely reconducted in the case when some desired levels are changed. Moreover, the second algorithm, which is an extension of the first one, has realized array response adjustment without leading to beam axis shift. The devised algorithms are computationally efficient. The application to pattern synthesis has been discussed and their effectiveness and superiority have been validated by examples under various situations.

Appendix A
Derivation of (45)

To begin with, we expand $\mathbf{E}_{\mathbf{a}(\theta_i)|\breve{\mathbf{A}}_{i-}}$ as

$$\begin{aligned}\mathbf{E}_{\mathbf{a}(\theta_i)|\breve{\mathbf{A}}_{i-}} &= \mathbf{a}(\theta_i)(\mathbf{a}^{\mathrm{H}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i))^{-1}\mathbf{a}^{\mathrm{H}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\\ &= \xi_i\mathbf{a}(\theta_i)\mathbf{a}^{\mathrm{H}}(\theta_i)(\mathbf{I}-\mathbf{P}_{\breve{\mathbf{A}}_{i-}}),\quad i=0,1,\ldots,Q \end{aligned}\tag{54}$$

where $\xi_i = (\mathbf{a}^{\mathrm{H}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i))^{-1}$, $i = 0, 1, \ldots, Q$, are real values. In addition, after some calculation, it is not hard to derive that

$$(\mathbf{I}-\mathbf{P}_{\breve{\mathbf{A}}_{i-}})\mathbf{a}(\theta_i) = \breve{\mathbf{A}}\mathbf{h}_i,\quad i=0,1,\ldots,Q \tag{55}$$

where $\mathbf{h}_i \in \mathbb{C}^{Q+1}$ is obtained by inserting an element one at the $(i+1)$th entry of $\mathbf{b}_i = -(\breve{\mathbf{A}}^{\mathrm{H}}_{i-}\breve{\mathbf{A}}_{i-})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}_{i-}\mathbf{a}(\theta_i)$, $i = 0, 1, \ldots, Q$. More exactly, we have

$$\mathbf{h}_i = [\mathbf{b}_i(1),\ldots,\mathbf{b}_i(i),1,\mathbf{b}_i(i+1),\ldots,\mathbf{b}_i(Q)]^{\mathrm{T}}.$$

Then, recalling (16), one can obtain that

$$\begin{aligned}\mathbf{T}_0\mathbf{w}_{\mathrm{pre}} &= \big(\mathbf{I}-\mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)}\big)\mathbf{w}_{\mathrm{pre}}\\ &= \big(\mathbf{I}-\mathbf{P}^{\mathrm{H}}_{\breve{\mathbf{A}}}+\mathbf{E}^{\mathrm{H}}_{\mathbf{a}(\theta_0)|\breve{\mathbf{A}}_{0-}}\big)\mathbf{w}_{\mathrm{pre}}\\ &= \mathbf{P}^{\perp}_{\breve{\mathbf{A}}}\mathbf{w}_{\mathrm{pre}}+\xi_0(\mathbf{I}-\mathbf{P}_{\breve{\mathbf{A}}_{0-}})\mathbf{a}(\theta_0)\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}_{\mathrm{pre}}\\ &= \mathbf{P}^{\perp}_{\breve{\mathbf{A}}}\mathbf{w}_{\mathrm{pre}}+c_0\breve{\mathbf{A}}\mathbf{h}_0 \end{aligned}\tag{56}$$

and

$$\begin{aligned}e^{-j\varphi_q}\mathbf{T}_q\mathbf{w}_{\mathrm{new},q} &= e^{-j\varphi_q}\mathbf{E}^{\mathrm{H}}_{\mathbf{a}(\theta_q)|\breve{\mathbf{A}}_{q-}}\mathbf{w}_{\mathrm{new},q}\\ &= e^{-j\varphi_q}\xi_q(\mathbf{I}-\mathbf{P}_{\breve{\mathbf{A}}_{q-}})\mathbf{a}(\theta_q)\mathbf{a}^{\mathrm{H}}(\theta_q)\mathbf{w}_{\mathrm{new},q}\\ &= c_q e^{-j\varphi_q}\breve{\mathbf{A}}\mathbf{h}_q \end{aligned}\tag{57}$$

where $c_0 \triangleq \xi_0\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}_{\mathrm{pre}}$ and $c_q \triangleq \xi_q\mathbf{a}^{\mathrm{H}}(\theta_q)\mathbf{w}_{\mathrm{new},q}$, $q = 1, \ldots, Q$. Thus, the $\mathbf{w}_{\mathrm{new}}$ in (20) can be reexpressed as

$$\mathbf{w}_{\mathrm{new}} = \mathbf{P}^{\perp}_{\breve{\mathbf{A}}}\mathbf{w}_{\mathrm{pre}}+\breve{\mathbf{A}}\mathbf{H}\mathbf{c} \tag{58}$$

where

$$\mathbf{H} \triangleq [\mathbf{h}_0,\mathbf{h}_1,\ldots,\mathbf{h}_Q] \in \mathbb{C}^{(Q+1)\times(Q+1)} \tag{59}$$

$$\mathbf{c} \triangleq [c_0, c_1e^{-j\varphi_1},\ldots,c_Qe^{-j\varphi_Q}]^{\mathrm{T}} \in \mathbb{C}^{Q+1}. \tag{60}$$

This completes the derivation of (45).

Appendix B
Derivation of (46)

For a given index $i = 0, 1, \ldots, Q$, we first define $\mathbf{J}_{1i} \in \mathbb{R}^{(2Q+3)\times 2}$ and $\mathbf{J}_{2i} \in \mathbb{R}^{(2Q+3)\times(2Q+1)}$ as

$$\mathbf{J}_{1i} = \begin{bmatrix}\mathbf{O}_{2i\times 2}\\ \mathbf{I}_2\\ \mathbf{O}_{[2(Q-i)+1]\times 2}\end{bmatrix} \tag{61a}$$

$$\mathbf{J}_{2i} = \begin{bmatrix}\mathbf{I}_{2i} & \mathbf{O}_{2i\times[2(Q-i)+1]}\\ \mathbf{O}_{2\times 2i} & \mathbf{O}_{2\times[2(Q-i)+1]}\\ \mathbf{O}_{[2(Q-i)+1]\times 2i} & \mathbf{I}_{2(Q-i)+1}\end{bmatrix}. \tag{61b}$$

Then, it is not hard to find that

$$\mathbf{Y}(\theta_i) = \breve{\mathbf{C}}\mathbf{J}_{1i},\quad \breve{\mathbf{C}}_{i-} = \breve{\mathbf{C}}\mathbf{J}_{2i},\ i=0,1,\ldots,Q. \tag{62}$$

On this basis, we can expand $\mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_i)|\breve{\mathbf{C}}_{i-}}$ as (63), shown on the top of next page, where the matrix $\mathbf{V}_i$, $i = 0, 1, \ldots, Q$, has also been specified.

From (63), it can be easily obtained that

$$\mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_q)|\breve{\mathbf{C}}_{q-}}\widetilde{\mathbf{u}}_q = \breve{\mathbf{C}}\mathbf{V}_q\widetilde{\mathbf{u}}_q,\ q=1,\ldots,Q. \tag{64}$$

In addition, note that $\mathbf{E}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)} + \mathbf{E}_{\mathbf{Y}(\theta_0)|\breve{\mathbf{C}}_{0-}} = \mathbf{P}_{\breve{\mathbf{C}}}$, or equivalently $\mathbf{I} - \mathbf{E}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)} = \mathbf{I} - \mathbf{P}_{\breve{\mathbf{C}}} + \mathbf{E}_{\mathbf{Y}(\theta_0)|\breve{\mathbf{C}}_{0-}}$. Thus, we have

$$\big(\mathbf{I}-\mathbf{E}^{\mathrm{T}}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)}\big)\widetilde{\mathbf{w}}_{\mathrm{pre}} = \big(\mathbf{P}^{\perp}_{\breve{\mathbf{C}}}+\breve{\mathbf{C}}\mathbf{V}_0\big)\widetilde{\mathbf{w}}_{\mathrm{pre}}. \tag{65}$$

$$\begin{aligned}\mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_i)|\breve{\mathbf{C}}_{i-}} &= \mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{i-}}\mathbf{Y}(\theta_i)\big(\mathbf{Y}^{\mathrm{T}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{i-}}\mathbf{Y}(\theta_i)\big)^{-1}\mathbf{Y}^{\mathrm{T}}(\theta_i)\\ &= \big(\mathbf{I}-\breve{\mathbf{C}}_{i-}\big(\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\breve{\mathbf{C}}_{i-}\big)^{-1}\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\big)\mathbf{Y}(\theta_i)\big(\mathbf{Y}^{\mathrm{T}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{i-}}\mathbf{Y}(\theta_i)\big)^{-1}\mathbf{Y}^{\mathrm{T}}(\theta_i)\\ &= \big(\mathbf{Y}(\theta_i)-\breve{\mathbf{C}}_{i-}\big(\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\breve{\mathbf{C}}_{i-}\big)^{-1}\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\mathbf{Y}(\theta_i)\big)\big(\mathbf{Y}^{\mathrm{T}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{i-}}\mathbf{Y}(\theta_i)\big)^{-1}\mathbf{Y}^{\mathrm{T}}(\theta_i)\\ &= \big(\breve{\mathbf{C}}\mathbf{J}_{1i}-\breve{\mathbf{C}}\mathbf{J}_{2i}\big(\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\breve{\mathbf{C}}_{i-}\big)^{-1}\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\mathbf{Y}(\theta_i)\big)\big(\mathbf{Y}^{\mathrm{T}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{i-}}\mathbf{Y}(\theta_i)\big)^{-1}\mathbf{Y}^{\mathrm{T}}(\theta_i)\\ &= \breve{\mathbf{C}}\underbrace{\big(\mathbf{J}_{1i}-\mathbf{J}_{2i}(\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\breve{\mathbf{C}}_{i-})^{-1}\breve{\mathbf{C}}^{\mathrm{T}}_{i-}\mathbf{Y}(\theta_i)\big)\big(\mathbf{Y}^{\mathrm{T}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{C}}_{i-}}\mathbf{Y}(\theta_i)\big)^{-1}\mathbf{Y}^{\mathrm{T}}(\theta_i)}_{\triangleq\mathbf{V}_i} \end{aligned} \tag{63}$$

Combining (64) and (65), one can reshape the weight vector $\widetilde{\mathbf{w}}_{\mathrm{new}}$ in (42) as

$$\begin{aligned}\widetilde{\mathbf{w}}_{\mathrm{new}} &= \mathbf{Z}_0\widetilde{\mathbf{w}}_{\mathrm{pre}}+\sum_{q=1}^{Q}\mathbf{Z}_q\widetilde{\mathbf{u}}_q\\ &= \big(\mathbf{I}-\mathbf{E}^{\mathrm{T}}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)}\big)\widetilde{\mathbf{w}}_{\mathrm{pre}}+\sum_{q=1}^{Q}\mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_q)|\breve{\mathbf{C}}_{q-}}\widetilde{\mathbf{u}}_q\\ &= \big(\mathbf{P}^{\perp}_{\breve{\mathbf{C}}}+\breve{\mathbf{C}}\mathbf{V}_0\big)\widetilde{\mathbf{w}}_{\mathrm{pre}}+\sum_{q=1}^{Q}\breve{\mathbf{C}}\mathbf{V}_q\widetilde{\mathbf{u}}_q\\ &= \mathbf{P}^{\perp}_{\breve{\mathbf{C}}}\widetilde{\mathbf{w}}_{\mathrm{pre}}+\breve{\mathbf{C}}\mathbf{F}\boldsymbol{\alpha}\end{aligned} \tag{66}$$

where

$$\mathbf{F} \triangleq [\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_Q] \in \mathbb{R}^{(2Q+3)\times 2N(Q+1)} \tag{67a}$$

$$\boldsymbol{\alpha} \triangleq \big[\widetilde{\mathbf{w}}^{\mathrm{T}}_{\mathrm{pre}}, \widetilde{\mathbf{u}}^{\mathrm{T}}_1, \ldots, \widetilde{\mathbf{u}}^{\mathrm{T}}_Q\big]^{\mathrm{T}} \in \mathbb{R}^{2N(Q+1)}. \tag{67b}$$

This completes the derivation of (46).

Appendix C
Derivation of (48)

If (47) is true, we have $\widetilde{\mathbf{w}}_{\mathrm{new}}-\widetilde{\mathbf{w}}_{\mathrm{pre}} \in \mathcal{R}(\breve{\mathbf{C}})$. Then, one can express $\widetilde{\mathbf{w}}_{\mathrm{new}}-\widetilde{\mathbf{w}}_{\mathrm{pre}}$ as

$$\widetilde{\mathbf{w}}_{\mathrm{new}}-\widetilde{\mathbf{w}}_{\mathrm{pre}} = \breve{\mathbf{C}}\boldsymbol{\mu} \tag{68}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{2Q+3}$ stands for the coefficient vector.

Recalling $\mathbf{w}_{\mathrm{new}} = [\mathbf{I}_N,\ j\mathbf{I}_N]\widetilde{\mathbf{w}}_{\mathrm{new}}$, $\mathbf{w}_{\mathrm{pre}} = [\mathbf{I}_N,\ j\mathbf{I}_N]\widetilde{\mathbf{w}}_{\mathrm{pre}}$ and the expression of $\breve{\mathbf{C}}$, it is not hard to obtain the following equation:

$$\begin{aligned}\mathbf{w}_{\mathrm{new}}-\mathbf{w}_{\mathrm{pre}} &= [\mathbf{I}_N,\ j\mathbf{I}_N]\breve{\mathbf{C}}\boldsymbol{\mu}\\ &= [(\mu_1-j\mu_2)\mathbf{a}(\theta_0), \ldots,\\ &\qquad (\mu_{2Q+1}-j\mu_{2Q+2})\mathbf{a}(\theta_Q), \mu_{2Q+3}\mathbf{d}(\theta_0)]\\ &= [\breve{\mathbf{A}},\ \mathbf{d}(\theta_0)]\boldsymbol{\Gamma}\boldsymbol{\mu}\end{aligned} \tag{69}$$

where μ_i stands for the ith element of $\boldsymbol{\mu}$, $i = 1, \ldots, 2Q+3$, $\boldsymbol{\Gamma} \in \mathbb{C}^{(Q+2)\times(2Q+3)}$ is defined as

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} 1 & -j & & & & & & \\ & & 1 & -j & & & & \\ & & & & \ddots & \ddots & & \\ & & & & & 1 & -j & \\ & & & & & & & 1 \end{bmatrix}. \tag{70}$$

According to (69), one can readily find that

$$\mathbf{P}^{\perp}_{[\breve{\mathbf{A}},\ \mathbf{d}(\theta_0)]}(\mathbf{w}_{\mathrm{new}}-\mathbf{w}_{\mathrm{pre}}) = \mathbf{0}. \tag{71}$$

This completes the derivation of (48).

Appendix D
Derivation of (49)

Taking $\rho_q = L_{\mathrm{pre}}(\theta_q, \theta_0)$ in (8), one can learn from [29] that $\beta_q = 1$, $q = 1, \ldots, Q$. Then, it can be readily found from (9) and (10) that

$$\mathbf{w}_{\mathrm{new},q} = \mathbf{w}_{\mathrm{pre}}, \quad q = 1, \ldots, Q. \tag{72}$$

Based on (72), if $\varphi_q = 0$, $q = 1, \ldots, Q$, is taken in (20), the resulting $\mathbf{w}_{\mathrm{new}}$ can be expressed as

$$\mathbf{w}_{\mathrm{new}} = \mathbf{T}_0\mathbf{w}_{\mathrm{pre}}+\sum_{q=1}^{Q}\mathbf{T}_q\mathbf{w}_{\mathrm{pre}} = \left(\sum_{i=0}^{Q}\mathbf{T}_i\right)\mathbf{w}_{\mathrm{pre}}. \tag{73}$$

To further prove that $\mathbf{w}_{\mathrm{new}} = \mathbf{w}_{\mathrm{pre}}$, we first define permutation matrices $\mathbf{J}_i$, $i = 0, 1, \ldots, Q$, as

$$\mathbf{J}_i \triangleq \left[\begin{array}{c:c:c} & \mathbf{I}_i & \\ \hdashline 1 & & \\ \hdashline & & \mathbf{I}_{Q-i} \end{array}\right] \in \mathbb{R}^{(Q+1)\times(Q+1)}. \tag{74}$$

Then, it is not hard to find that $[\mathbf{a}(\theta_i),\ \breve{\mathbf{A}}_{i-}] = \breve{\mathbf{A}}\mathbf{J}_i$ and $\big[\mathbf{a}(\theta_i),\ \mathbf{0}_{N\times Q}\big] = \big[\mathbf{0}_{N\times i},\ \mathbf{a}(\theta_i),\ \mathbf{0}_{N\times(Q-i)}\big]\mathbf{J}_i$, $i = 0, 1, \ldots, Q$. Thus, we have

$$\begin{aligned}\sum_{i=0}^{Q}\mathbf{E}_{\mathbf{a}(\theta_i)|\breve{\mathbf{A}}_{i-}} &= \sum_{i=0}^{Q}[\mathbf{a}(\theta_i),\ \mathbf{0}]((\breve{\mathbf{A}}\mathbf{J}_i)^{\mathrm{H}}(\breve{\mathbf{A}}\mathbf{J}_i))^{-1}(\breve{\mathbf{A}}\mathbf{J}_i)^{\mathrm{H}}\\ &= \sum_{i=0}^{Q}[\mathbf{a}(\theta_i),\ \mathbf{0}](\mathbf{J}_i^{\mathrm{H}}\breve{\mathbf{A}}^{\mathrm{H}}\breve{\mathbf{A}}\mathbf{J}_i)^{-1}\mathbf{J}_i^{\mathrm{H}}\breve{\mathbf{A}}^{\mathrm{H}}\\ &= \sum_{i=0}^{Q}[\mathbf{a}(\theta_i),\ \mathbf{0}]\mathbf{J}_i^{-1}(\breve{\mathbf{A}}^{\mathrm{H}}\breve{\mathbf{A}})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}\\ &= \sum_{i=0}^{Q}[\mathbf{0}_{N\times i},\ \mathbf{a}(\theta_i),\ \mathbf{0}_{N\times(Q-i)}](\breve{\mathbf{A}}^{\mathrm{H}}\breve{\mathbf{A}})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}\\ &= \breve{\mathbf{A}}(\breve{\mathbf{A}}^{\mathrm{H}}\breve{\mathbf{A}})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}\\ &= \mathbf{P}_{\breve{\mathbf{A}}}.\end{aligned} \tag{75}$$

Recalling the definitions of $\mathbf{T}_i$ $(i = 0, 1, \ldots, Q)$ in (16), one can obtain that

$$\begin{aligned}\sum_{i=0}^{Q}\mathbf{T}_i &= \mathbf{I} - \mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)} + \sum_{q=1}^{Q}\mathbf{E}^{\mathrm{H}}_{\mathbf{a}(\theta_q)|\breve{\mathbf{A}}_{q-}} \\ &= \mathbf{I} - \mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)} - \mathbf{E}^{\mathrm{H}}_{\mathbf{a}(\theta_0)|\breve{\mathbf{A}}_{0-}} + \sum_{i=0}^{Q}\mathbf{E}^{\mathrm{H}}_{\mathbf{a}(\theta_i)|\breve{\mathbf{A}}_{i-}} \\ &= \mathbf{I} - \mathbf{P}^{\mathrm{H}}_{\breve{\mathbf{A}}} + \mathbf{P}^{\mathrm{H}}_{\breve{\mathbf{A}}} \\ &= \mathbf{I}. \end{aligned} \tag{76}$$

According to (73) and (76), one can see that $\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{pre}}$. This completes the derivation of (49).

Appendix E
Derivation of (50)

The same as the proof in Appendix D, one can see that $\mathbf{w}_{\text{new},q} = \mathbf{w}_{\text{pre}}$ and then $\widetilde{\mathbf{w}}_{\text{new},q} = \widetilde{\mathbf{w}}_{\text{pre}}$, if taking $\rho_q = L_{\text{pre}}(\theta_q, \theta_0)$, $q = 1, \ldots, Q$. On this basis, setting $\phi_q = 0$ $(q = 1, \ldots, Q)$ in (33) yields

$$\widetilde{\mathbf{u}}_q = \widetilde{\mathbf{w}}_{\text{new},q} = \widetilde{\mathbf{w}}_{\text{pre}}, \quad q = 1, \ldots, Q. \tag{77}$$

Then, the resulting $\widetilde{\mathbf{w}}_{\text{new}}$ in (42) can be expressed as

$$\widetilde{\mathbf{w}}_{\text{new}} = \left(\sum_{i=0}^{Q}\mathbf{Z}_i\right)\widetilde{\mathbf{w}}_{\text{pre}}. \tag{78}$$

Similar to the derivation in (75), it is not difficult to obtain that

$$\mathbf{E}_{\widetilde{\mathbf{d}}(\theta_0)|\breve{\mathbf{C}}_{\widetilde{\mathbf{d}}-}} + \sum_{i=0}^{Q}\mathbf{E}_{\mathbf{Y}(\theta_i)|\breve{\mathbf{C}}_{i-}} = \mathbf{P}_{\breve{\mathbf{C}}} \tag{79}$$

or equivalently

$$\sum_{q=1}^{Q}\mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_q)|\breve{\mathbf{C}}_{q-}} = \mathbf{P}_{\breve{\mathbf{C}}} - \mathbf{E}^{\mathrm{T}}_{\widetilde{\mathbf{d}}(\theta_0)|\breve{\mathbf{C}}_{\widetilde{\mathbf{d}}-}} - \mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_0)|\breve{\mathbf{C}}_{0-}} \tag{80}$$

where

$$\breve{\mathbf{C}}_{\widetilde{\mathbf{d}}-} \triangleq [\mathbf{Y}(\theta_0), \mathbf{Y}(\theta_1), \ldots, \mathbf{Y}(\theta_Q)] \in \mathbb{R}^{2N\times(2Q+2)}. \tag{81}$$

Thus, we have

$$\begin{aligned}\left(\sum_{i=0}^{Q}\mathbf{Z}_i\right)\widetilde{\mathbf{w}}_{\text{pre}} &= \left(\mathbf{I} - \mathbf{E}^{\mathrm{T}}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)} + \sum_{q=1}^{Q}\mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_q)|\breve{\mathbf{C}}_{q-}}\right)\widetilde{\mathbf{w}}_{\text{pre}} \\ &= \left(\mathbf{I} - \mathbf{E}^{\mathrm{T}}_{\widetilde{\mathbf{d}}(\theta_0)|\breve{\mathbf{C}}_{\widetilde{\mathbf{d}}-}}\right)\widetilde{\mathbf{w}}_{\text{pre}} \end{aligned} \tag{82}$$

where we have used the fact that

$$\mathbf{P}_{\breve{\mathbf{C}}} = \mathbf{E}^{\mathrm{T}}_{\breve{\mathbf{C}}_{0-}|\mathbf{Y}(\theta_0)} + \mathbf{E}^{\mathrm{T}}_{\mathbf{Y}(\theta_0)|\breve{\mathbf{C}}_{0-}}. \tag{83}$$

Recalling (78), the derivation of (50) can be completed.

Acknowledgment

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions.

References

[1] O. L. Frost, III, “An algorithm for linearly constrained adaptive array processing,” *Proc. IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.

[2] J. Xu, G. Liao, S. Zhu, and L. Huang, “Response vector constrained robust LCMV beamforming based on semidefinite programming,” *IEEE Trans. Signal Process.*, vol. 63, no. 21, pp. 5720–5732, Nov. 2015.

[3] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[4] C. Y. Chen and P. P. Vaidyanathan, “Quadratically constrained beamforming robust against direction-of-arrival mismatch,” *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4139–4150, Aug. 2007.

[5] Z. L. Yu, W. Ser, M. H. Er, Z. Gu, and Y. Li, “Robust adaptive beamformers based on worst-case optimization and constraints on magnitude response,” *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2615–2628, Jul. 2009.

[6] Z. L. Yu, M. H. Er, and W. Ser, “A novel adaptive beamformer based on semidefinite programming (SDP) with magnitude response constraints,” *IEEE Trans. Antennas Propag.*, vol. 56, no. 5, pp. 1297–1307, May 2008.

[7] B. Liao, K. M. Tsui, and S. C. Chan, “Robust beamforming with magnitude response constraints using iterative second-order cone programming,” *IEEE Trans. Antennas Propag.*, vol. 59, no. 9, pp. 3477–3482, Sep. 2011.

[8] S. E. Nai, W. Ser, Z. L. Yu, and S. Rahardja, “A robust adaptive beamforming framework with beampattern shaping constraints,” *IEEE Trans. Antennas Propag.*, vol. 57, no. 7, pp. 2198–2203, Jul. 2009.

[9] S. A. Vorobyov, A. B. Gershman, and Z.-Q. Luo, “Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem,” *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 313–324, Feb. 2003.

[10] C. C. Gaudes, I. Santamaria, J. Via, E. M. Gomez, and T. S. Paules, “Robust array beamforming with sidelobe control using support vector machines,” *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 574–584, Feb. 2007.

[11] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[12] K. Chen, X. Yun, Z. He, and C. Han, “Synthesis of sparse planar arrays using modified real genetic algorithm,” *IEEE Trans. Antennas Propag.*, vol. 55, no. 4, pp. 1067–1073, Apr. 2007.

[13] D. W. Boeringer and D. H. Werner, “Particle swarm optimization versus genetic algorithms for phased array synthesis,” *IEEE Trans. Antennas Propag.*, vol. 52, no. 3, pp. 771–779, Mar. 2004.

[14] V. Murino, A. Trucco, and C. S. Regazzoni, “Synthesis of unequally spaced arrays by simulated annealing,” *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 119–122, Jan. 1996.

[15] M. H. Er, “Array pattern synthesis with a controlled mean-square sidelobe level,” *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 977–981, Apr. 1992.

[16] C.-C. Tseng and L. J. Griffiths, “A simple algorithm to achieve desired patterns for arbitrary arrays,” *IEEE Trans. Signal Process.*, vol. 40, no. 11, pp. 2737–2746, Nov. 1992.

[17] C. A. Olen and R. T. Compton, Jr., “A numerical pattern synthesis algorithm for arrays,” *IEEE Trans. Antennas Propag.*, vol. 38, no. 10, pp. 1666–1676, Oct. 1990.

[18] P. Y. Zhou and M. A. Ingram, “Pattern synthesis for arbitrary arrays using an adaptive array method,” *IEEE Trans. Antennas Propag.*, vol. 47, no. 5, pp. 862–869, May 1999.

[19] X. Zhang, Z. He, X.-G. Xia, B. Liao, X. Zhang, and Y. Yang, “OPARC: Optimal and precise array response control algorithm—Part I: Fundamentals,” *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 652–667, Feb. 2019.

[20] X. Zhang, Z. He, X.-G. Xia, B. Liao, X. Zhang, and Y. Yang, “OPARC: Optimal and precise array response control algorithm—Part II: Multipoints and applications,” *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 668–683, Feb. 2019.

[21] H. K. Van Trees, *Optimum Array Processing*. New York, NY, USA: Wiley, 2002.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[23] H. Lebret and S. Boyd, “Antenna array pattern synthesis via convex optimization,” *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 526–532, Mar. 1997.

[24] B. Fuchs, “Application of convex relaxation to array synthesis problems,” *IEEE Trans. Antennas Propag.*, vol. 62, no. 2, pp. 634–640, Feb. 2014.

[25] X. Zhang, Z. He, X. Zhang, and W. Peng, “High-performance beampattern synthesis via linear fractional semidefinite relaxation and quasi-convex optimization,” *IEEE Trans. Antennas Propag.*, vol. 66, no. 7, pp. 3421–3431, Jul. 2018.

[26] S. E. Nai, W. Ser, Z. L. Yu, and H. Chen, “Beampattern synthesis for linear and planar arrays with antenna selection by convex optimization,” *IEEE Trans. Antennas Propag.*, vol. 58, no. 12, pp. 3923–3930, Dec. 2010.

[27] F. Wang, V. Balakrishnan, P. Y. Zhou, J. J. Chen, R. Yang, and C. Frank, “Optimal array pattern synthesis using semidefinite programming,” *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1172–1183, May 2003.

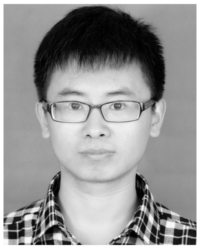
[28] X. Zhang, Z. He, B. Liao, X. Zhang, Z. Cheng, and Y. Lu, “A^2RC: An accurate array response control algorithm for pattern synthesis,” *IEEE Trans. Signal Process.*, vol. 65, no. 7, pp. 1810–1824, Apr. 2017.

[29] X. Zhang, Z. He, B. Liao, X. Zhang, and W. Peng, “Pattern synthesis for arbitrary arrays via weight vector orthogonal decomposition,” *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1286–1299, Mar. 2018.

[30] X. Zhang, Z. He, B. Liao, X. Zhang, and W. Peng, “Pattern synthesis with multipoint accurate array response control,” *IEEE Trans. Antennas Propag.*, vol. 65, no. 8, pp. 4075–4088, Aug. 2017.

[31] X. Zhang, Z. He, B. Liao, Y. Yang, J. Zhang, and X. Zhang, “Flexible array response control via oblique projection,” *IEEE Trans. Signal Process.*, vol. 67, no. 12, pp. 3126–3139, Jun. 2019.

[32] R. T. Behrens and L. L. Scharf, “Signal processing applications of oblique projection operators,” *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1413–1424, Jun. 1994.



Xuejing Zhang (S’17) was born in Shijiazhuang, Hebei, China. He received the B.S. degree in electrical engineering from Huaqiao University, Xiamen, China, in 2011 and the M.S. degree in signal and information processing from Xidian University, Xi’an, China, in 2014. He is currently pursuing the Ph.D. degree in signal and information processing with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China.

Since 2017, he has been a Visiting Student with the University of Delaware, Newark, DE, USA. His current research interests include array signal processing and wireless communications.

Zishu He (M’11) was born in Chengdu, 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, 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.

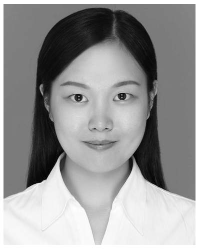
Bin Liao (S’09–M’13–SM’16) received the B.Eng. and M.Eng degrees in electronic engineering from Xidian University, Xian, China, in 2006 and 2009, respectively, and the Ph.D. degree in electronic engineering from The University of Hong Kong, Hong Kong, in 2013.

From 2013 to 2014, he was a Research Assistant with the Department of Electrical and Electronic Engineering, The University of Hong Kong. From 2016 to 2016, he was a Research Scientist with the Department of Electrical and Electronic Engineering, The University of Hong Kong. He is currently an Associate Professor with the Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen, China. His current research interests include sensor array processing, adaptive filtering, convex optimization, with applications to radar, navigation, and communications.

Dr. Liao was a recipient of the Best Paper Award at the 21st International Conference on Digital Signal Processing (2016 DSP) and 22nd International Conference on Digital Signal Processing (2017 DSP). He is an Associate Editor of IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, *IET Signal Processing*, *Multidimensional Systems and Signal Processing*, and IEEE ACCESS.

Xuepan Zhang was born in Shijiazhuang, Hebei, China. He received the B.S. and Ph.D. degrees in electrical engineering from the National Laboratory of Radar Signal Processing, Xidian University, Xi’an, China, in 2010 and 2015, respectively.

He is currently a Principal Investigator with the Qian Xuesen Laboratory of Space Technology, Beijing, China. His current research interests include synthetic aperture radar (SAR), ground moving target indication (GMTI), and deep learning.



Yue Yang (S’17) was born in Suining, Sichuan, China. She received the B.Eng. degree in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2015, where she is currently pursuing the Ph.D. degree in signal and information processing with the School of Information and Communication Engineering.

Since 2019, she has been a Visiting Student with the National University of Singapore, Singapore. Her current research interests include synthetic aperture radar (SAR) imaging, sparse signal reconstruction, and statistical signal processing.