Flexible Array Response Control via Oblique Projection

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

***Abstract*—This paper presents a flexible array response control algorithm via oblique projection, abbreviated as FARCOP, and its application to array pattern synthesis. The proposed FARCOP algorithm stems from the adaptive array theory, and it can flexibly, precisely and simultaneously adjust the array response levels at multiple angles based on an arbitrarily given weight vector. Different from the existing approaches, the proposed FARCOP algorithm controls multi-point responses by linearly transferring the given weight vector, with a transformation matrix containing a set of parameters, each of which can be very easily determined by the desired response level (at the control angle). Owing to the fact that those parameters are independent of each other, the response levels at the control angles can be either individually or jointly and, therefore, flexibly adjusted. Since the parameter phases can be arbitrary, we take the beampattern into account and propose to uniquely choose the optimal parameters under the typical criterion of maximum white noise gain (WNG). Accordingly, a gradient projection (GP) algorithm is devised to achieve the optimal solution. Moreover, a closed-form solution is derived for the centro-symmetric array. In addition, the application of the FARCOP algorithm to pattern synthesis is discussed. Comparing to the state-of-the-art methods like multi-point accurate array response control (MA^{2}RC), the proposed FARCOP algorithm controls the array responses more flexibly with lower computational complexity. Representative examples are presented to demonstrate the effectiveness and superiority of the FARCOP algorithm under various situations.**

***Index Terms*—Array response control, oblique projection, array pattern synthesis, adaptive array theory.**

I. Introduction

SENSOR arrays have found numerous applications in the fields of, for example, radar, wireless communication and remote sensing [1]. In general, how to flexibly control the array response is of great significance in practice. For instance, in a radar system, it is effective to suppress the undesirable interference by shaping a beampattern with fixed nulls. While in some communication systems, it enables users to receive data more efficiently by freely adjusting the transmitted beampatterns or synthesizing multiple-beam patterns. Additionally, synthesizing a beampattern with broad mainlobe is beneficial to extend monitoring areas in satellite remote sensing.

In the past years, quite a number of approaches to array response control[1] have been developed. For example, in data-dependent scenarios, the linearly constrained minimum variance (LCMV) [2] method is able to control array responses of specific points by imposing linear constraints on the output of beamformer. A variant of LCMV is presented in [3] to adjust the mainlobe responses and control the sidelobe levels, with the aid of the semidefinite relaxation (SDR) technique [4]. In [5], the response levels at a desired range of angles are adjusted to exceed a constant value by quadratically restricting the magnitude responses of two steering vectors. The concept of spectral factorization is introduced in [6] to realize magnitude response control. In [7], a new array response control approach is presented by transforming the array output power and the magnitude response to linear functions of the autocorrelation sequence of the array weight. This method is able to control certain response region with specified beamwidth and response ripple. A strategy of accurate main beam control is developed in [8], where a class of approaches are provided to solve the resulting problem. In [9], desired sidelobe is achieved with second-order cone program (SOCP), which ensures that the sidelobes are exactly below the required level.

To control the array response in data-independent scenarios, a conjugate symmetric weight vector is obtained in [10] to make the responses satisfy specified non-convex lower bound constraints, by utilizing the symmetric geometries of linear and planar arrays. A simple iterative algorithm is proposed in [11] to precisely adjust the responses of certain points to their desired levels. The unified approach presented in [12] allows the quiescent response of the beamformer to be specified as any desirable fixed-weight response. Recently, an optimal and precise array response control (OPARC) algorithm has been reported in [13] and [14], where array responses are precisely controlled by

Manuscript received February 13, 2018; revised August 30, 2018; accepted April 8, 2019. Date of publication April 18, 2019; date of current version May 6, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pengfei Xia. This work was supported in part by the National Nature Science Foundation of China under Grants 61671139, 61671137, 61771316, 61701499, 61871085, and 61501301, in part by the Foundation of Shenzhen under Grant JCYJ20170302150044331, and in part by the China Scholarship Council. *(Corresponding author: Xuepan Zhang.)*

X. Zhang, Z. He, and Y. Yang are with the 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).

J. Zhang is with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: zhangjf@szu.edu.cn).

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

Digital Object Identifier 10.1109/TSP.2019.2912147

[1] In this work, array response control is defined as designing a weight vector such that the corresponding array response fulfills specific requirements. It includes adaptive (data-dependent) and non-adaptive (data-independent) approaches. We shall focus on the latter case.

assigning virtual interferences. Additionally, an accurate array response control (A^{2}RC) algorithm and a weight vector orthogonal decomposition (WORD) algorithm were proposed in [15] and [16], respectively, to control array response based on an arbitrarily-specified weight vector. Since these two approaches are restricted to adjust the response level of a single point, a multi-point accurate array response control (MA^{2}RC) algorithm is thus developed in [17] to precisely control array responses at multiple angles. However, a relatively high computation cost is required in MA^{2}RC algorithm, especially for large arrays.

Note that even though some of the existing methods mentioned above show some flexibilities in array response control, one may expect a more flexible approach which is able to adjust the response levels at multiple points either individually or simultaneously, and more importantly, in a simple manner. To this end, in this paper we devise a new algorithm named as flexible array response control via oblique projection (FARCOP). This algorithm stems from the adaptive array theory. More specifically, we find that the optimal weight vector can be obtained alternatively, by linearly transferring the quiescent weight. The transformation matrix is constructed with the aid of the oblique projection (OP) [18] operator and contains a set of parameters, which are related to the interference-to-noise ratios (INRs) and affect the response levels at the angles of interferences. By exploiting this concept, for an arbitrarily given weight vector, we propose to control the array response levels of multiple angles, by linearly transferring the pre-assigned weight and then selecting the parameters of the specific transformation matrix. Different from the existing approaches, in FARCOP, the array responses at the prescribed angles can be individually adjusted by simply varying the modulus of parameter. To further determine the phases of parameters, we propose to maximize the white noise gain (WNG) [19]–[21] and obtain the ultimate solution by using a computationally attractive gradient projection (GP) algorithm [22]. Moreover, we derive closed-form solutions for the centro-symmetric arrays [23]. In addition, comparisons between the proposed FARCOP algorithm and the existing MA^{2}RC algorithm are summarized, and the application of FARCOP to array pattern synthesis [24]–[26] is studied.

The rest of the paper is organized as follows. In Section II, the oblique projection is briefly introduced. The proposed FARCOP algorithm is presented in Section III and its comparison with the existing MA^{2}RC approach is provided in Section IV. In Section V, we discuss the application of FARCOP to pattern synthesis. Representative simulations are presented in Section VI and conclusions are drawn in Section VII.

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 and $\|\cdot\|_2$ denotes the l_2 norm. We use $\boldsymbol{\eta}(i)$ to represent the ith element of vector $\boldsymbol{\eta}$. $\mathrm{Diag}(\cdot)$ stands for the diagonal matrix with diagonal elements equal to the input entries. $\mathbb{R}$ and $\mathbb{C}$ denote the sets of real and complex numbers, respectively. $\mathcal{R}(\cdot)$ and $\mathcal{N}(\cdot)$ return the column space and the null space of the input matrix, respectively. $\mathcal{R}^{\perp}(\cdot)$ is the orthogonal complementary space of $\mathcal{R}(\cdot)$. $\odot$ denotes the element-wise product operator. $\angle(\cdot)$ returns the argument of the input. $(\cdot)^{\dagger}$ represents the pseudo-inverse and $\oplus$ is the direct sum operation. Finally, we use $\lambda_{\max}(\cdot)$ to stand for the maximum eigenvalue of the input matrix.

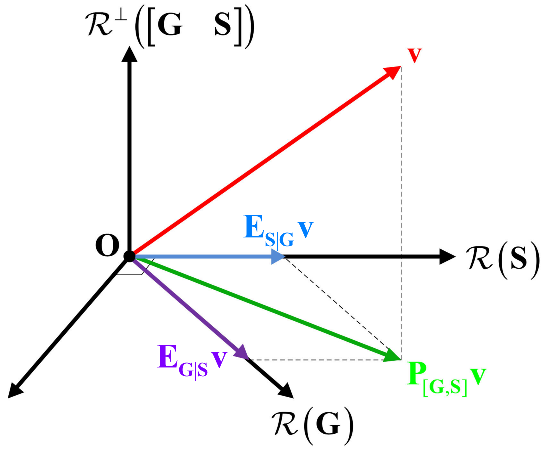

$\mathcal{R}^{\perp}([\mathbf{G} \quad \mathbf{S}])$
$\mathbf{v}$
$\mathbf{O}$
$\mathbf{E}_{\mathbf{S}|\mathbf{G}}\mathbf{v}$
$\mathcal{R}(\mathbf{S})$
$\mathbf{E}_{\mathbf{G}|\mathbf{S}}\mathbf{v}$
$\mathbf{P}_{[\mathbf{G},\mathbf{S}]}\mathbf{v}$
$\mathcal{R}(\mathbf{G})$

Fig. 1. Illustration of oblique projection.

II. Preliminaries and Motivations

A. Oblique Projection

Assume that $\mathbf{Z} \in \mathbb{C}^{m\times(p+l)}$ has a full column rank and can be partitioned as $\mathbf{Z} = \begin{bmatrix}\mathbf{G} & \mathbf{S}\end{bmatrix}$, where $\mathbf{G} \in \mathbb{C}^{m\times p}$ and $\mathbf{S} \in \mathbb{C}^{m\times l}$. It is known that the orthogonal projection whose range is $\mathcal{R}(\mathbf{Z})$ can be expressed as

$$\mathbf{P}_{\mathbf{Z}} = \mathbf{Z}(\mathbf{Z}^{\mathrm{H}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{H}} \tag{1}$$

which is termed as the orthogonal projector onto $\mathcal{R}(\mathbf{Z})$ [27]. Accordingly, the orthogonal projector whose range is $\mathcal{R}^{\perp}(\mathbf{Z})$ is given by $\mathbf{P}_{\mathbf{Z}}^{\perp} = \mathbf{I} - \mathbf{P}_{\mathbf{Z}}$.

Projection matrix that is not orthogonal is referred to as oblique projection (OP) [18]. Specifically, we can decompose $\mathbf{P}_{\mathbf{Z}}$ in (1) as

$$\mathbf{P}_{\mathbf{Z}} = \mathbf{P}_{[\mathbf{G}\,\mathbf{S}]} = \mathbf{E}_{\mathbf{G}|\mathbf{S}} + \mathbf{E}_{\mathbf{S}|\mathbf{G}} \tag{2}$$

where $\mathbf{E}_{\mathbf{G}|\mathbf{S}}$ and $\mathbf{E}_{\mathbf{S}|\mathbf{G}}$, termed as oblique projectors [18], are given by

$$\mathbf{E}_{\mathbf{G}|\mathbf{S}} = [\mathbf{G}\ \mathbf{0}](\mathbf{Z}^{\mathrm{H}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{H}} = \mathbf{G}(\mathbf{G}^{\mathrm{H}}\mathbf{P}_{\mathbf{S}}^{\perp}\mathbf{G})^{-1}\mathbf{G}^{\mathrm{H}}\mathbf{P}_{\mathbf{S}}^{\perp} \tag{3a}$$

$$\mathbf{E}_{\mathbf{S}|\mathbf{G}} = [\mathbf{0}\ \mathbf{S}](\mathbf{Z}^{\mathrm{H}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{H}} = \mathbf{S}(\mathbf{S}^{\mathrm{H}}\mathbf{P}_{\mathbf{G}}^{\perp}\mathbf{S})^{-1}\mathbf{S}^{\mathrm{H}}\mathbf{P}_{\mathbf{G}}^{\perp}. \tag{3b}$$

It can be readily verified that

$$\mathbf{E}_{\mathbf{G}|\mathbf{S}}\mathbf{G} = \mathbf{G}, \quad \mathbf{E}_{\mathbf{G}|\mathbf{S}}\mathbf{S} = \mathbf{0} \tag{4a}$$

$$\mathbf{E}_{\mathbf{S}|\mathbf{G}}\mathbf{S} = \mathbf{S}, \quad \mathbf{E}_{\mathbf{S}|\mathbf{G}}\mathbf{G} = \mathbf{0}. \tag{4b}$$

As illustrated in Fig. 1, the oblique projector $\mathbf{E}_{\mathbf{G}|\mathbf{S}}$ projects vectors onto $\mathcal{R}(\mathbf{G})$ along the direction parallel to $\mathcal{R}(\mathbf{S})$, and likewise for $\mathbf{E}_{\mathbf{S}|\mathbf{G}}$. Note that oblique projectors are idempotent but not Hermitian symmetric.

B. Adaptive Array Theory

In adaptive beamforming, the output signal-to-interference-plus-noise ratio (SINR) is maximized and the optimal weight vector is given by

$$\mathbf{w}_{\rm opt} = \alpha \mathbf{R}_{n+i}^{-1}\mathbf{a}(\theta_0) \tag{5}$$

where α is the normalization factor that does not affect the output SINR, $\mathbf{R}_{n+i}$ denotes the $N \times N$ noise-plus-interference covariance matrix, θ_0 is the main beam axis, $\mathbf{a}(\theta)$ stands for the steering vector at θ and is given by

$$\mathbf{a}(\theta) = [g_1(\theta)e^{-j\omega\tau_1(\theta)}, \ldots, g_N(\theta)e^{-j\omega\tau_N(\theta)}]^{\rm T} \tag{6}$$

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, $n = 1, \ldots, N$, ω denotes the operating frequency.

Suppose that the noise is spatially-white and Q interferences are independent with each other. Then, the normalized noise-plus-interference covariance matrix is given by [19]

$$\boldsymbol{\Xi}_{n+i} \triangleq \frac{\mathbf{R}_{n+i}}{\sigma_n^2} = \mathbf{I} + \sum_{q=1}^{Q} \beta_q \mathbf{a}(\theta_q)\mathbf{a}^{\rm H}(\theta_q) \tag{7}$$

where σ_n^2 is the noise power, $\beta_q \triangleq \sigma_q^2/\sigma_n^2$, σ_q^2 and θ_q represent the interference-to-noise ratio (INR), power and direction of the qth interference, respectively. Consequently, the optimal weight vector can be scaled as

$$\mathbf{w}_\star = \boldsymbol{\Xi}_{n+i}^{-1}\mathbf{a}(\theta_0) \tag{8}$$

which corresponds to the same SINR as $\mathbf{w}_{\rm opt}$ in (5).

C. Motivations

It can be seen that in the adaptive case the optimal weight vector $\mathbf{w}_\star$ depends on the data-dependent matrix $\boldsymbol{\Xi}_{n+i}$ (or $\mathbf{R}_{n+i}$). However, this quantity is generally unavailable when we design a data-independent array response pattern

$$L(\theta, \theta_0) \triangleq \frac{|\mathbf{w}^{\rm H}\mathbf{a}(\theta)|^2}{|\mathbf{w}^{\rm H}\mathbf{a}(\theta_0)|^2} \tag{9}$$

to meet specific requirements. To this end, in [13], the concept of virtual normalized noise-plus-interference covariance matrix (VCM) was introduced, and it was shown that the array responses can be adjusted by assigning virtual interferences [24] from specific directions and then selecting appropriate INRs. However, when controlling array responses at multiple points, the configuration of the associated INRs of the virtual interferences is somewhat difficult to handle. This is because the mapping of the array response at the direction of the interference to the corresponding INR is not one-to-one, but mutually affected by other interferences. Moreover, the existing methods of assigning virtual interferences cannot realize the response control based on an arbitrarily given weight vector.

To circumvent these shortcomings, an equivalent parameterization of the optimal beamformer is presented next, by mapping INRs into a new parameter vector whose elements are one-to-one related to the responses at the directions of interferences.

III. The Proposed FARCOP Algorithm

A. An Equivalent Parameterization of the Optimal Weight

To begin with, we rewrite the normalized covariance matrix in (7) as

$$\boldsymbol{\Xi}_{n+i} = \mathbf{I} + \mathbf{A}(\theta_1, \ldots, \theta_Q)\boldsymbol{\Sigma}\mathbf{A}^{\rm H}(\theta_1, \ldots, \theta_Q) \tag{10}$$

where $\mathbf{A}(\theta_i, \ldots, \theta_j)$ and $\boldsymbol{\Sigma}$ define the steering matrix and covariance matrix of the interferences, respectively, as

$$\mathbf{A}(\theta_i, \ldots, \theta_j) \triangleq [\mathbf{a}(\theta_i), \ldots, \mathbf{a}(\theta_j)] \tag{11a}$$

$$\boldsymbol{\Sigma} = {\rm Diag}\left([\beta_1, \beta_2, \ldots, \beta_Q]\right). \tag{11b}$$

Substituting (10) into (8) and applying the Woodbury matrix identity [27] to $\boldsymbol{\Xi}_{n+i}$ yields

$$\begin{aligned}\mathbf{w}_\star &= \left(\mathbf{I} + \mathbf{A}(\theta_1, \ldots, \theta_Q)\boldsymbol{\Sigma}\mathbf{A}^{\rm H}(\theta_1, \ldots, \theta_Q)\right)^{-1}\mathbf{a}(\theta_0) \\ &= \underbrace{\left[\mathbf{a}(\boldsymbol{\theta_0}) \quad \mathbf{A}(\theta_1, \ldots, \theta_Q)\right]}_{\triangleq \breve{\mathbf{A}}}\left[1 \quad \mathbf{u}^{\rm T}\right]^{\rm T} \end{aligned} \tag{12}$$

where $\breve{\mathbf{A}} \triangleq \mathbf{A}(\theta_0, \theta_1, \ldots, \theta_Q) \in \mathbb{C}^{N\times(Q+1)}$, and $\mathbf{u} \in \mathbb{C}^Q$ is given by

$$\begin{aligned}\mathbf{u} = -&\left(\mathbf{I} + \boldsymbol{\Sigma}\mathbf{A}^{\rm H}(\theta_1, \ldots, \theta_Q)\mathbf{A}(\theta_1, \ldots, \theta_Q)\right)^{-1} \\ &\boldsymbol{\Sigma}\mathbf{A}^{\rm H}(\theta_1, \ldots, \theta_Q)\mathbf{a}(\theta_0). \end{aligned} \tag{13}$$

It is found from (12) that the optimal weight vector is a linear combination of $\mathbf{a}(\theta_0), \mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_Q)$ with coefficients included in $\mathbf{u}$. Obviously, it is seen that we cannot simply set the INR β_q to adjust the response level at θ_q, while keeping the response at the other $Q - 1$ angles unaltered. The reason lies in the fact that the response level at θ_q is related to all INRs $\beta_1, \ldots, \beta_Q$. Thus, it is generally challenging to realize multi-point responses control with the above common parameterization (in terms of INR), as shown in the literatures [13]–[15].

Nevertheless, by introducing a new parameterization, we shall show that it is possible to control the responses at interference angles individually. Let us first define

$$v(i, j) \triangleq \mathbf{a}^{\rm H}(\theta_i)\mathbf{a}(\theta_j) \tag{14}$$

and obtain the following proposition where the weight vector $\mathbf{w}_\star$ in (12) are alternatively expressed.

Proposition 1: Suppose that $\mathbf{a}(\theta_0), \mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_Q)$ are linearly independent and $v(q, 0) \neq 0$ for $\forall q \in \{1, \ldots, Q\}$. Then, given $\beta_1, \ldots, \beta_Q$, there exist $\eta_1, \ldots, \eta_Q$ such that the resulting weight $\mathbf{w}_\star$ in (12) can be expressed with the oblique projectors as

$$\mathbf{w}_{\rm OP} = \left(\left(\mathbf{I} - \mathbf{E}^{\rm H}_{\breve{\mathbf{A}}_{0-}|0}\right) + \sum_{q=1}^{Q}\eta_q\mathbf{E}^{\rm H}_{q|\breve{\mathbf{A}}_{q-}}\right)\mathbf{a}(\theta_0) = c\mathbf{w}_\star \tag{15}$$

where c is a constant, $\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}$ and $\mathbf{E}_{q|\breve{\mathbf{A}}_{q-}}$ denote the oblique projectors as

$$\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0} \triangleq \mathbf{E}_{\breve{\mathbf{A}}_{0-}|\mathbf{a}(\theta_0)} \tag{16a}$$

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

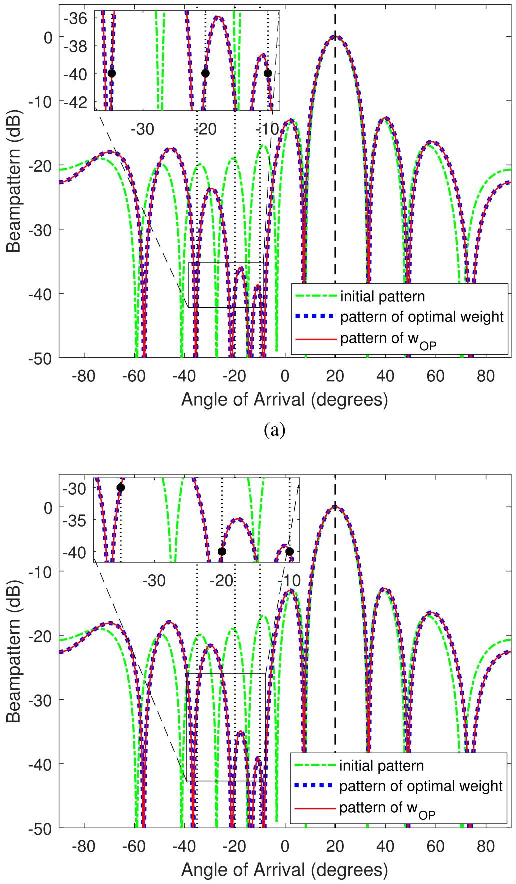
with $\breve{\mathbf{A}}_{i-}$ being the following matrix resulted by removing the column $\mathbf{a}(\theta_i)$ from $\breve{\mathbf{A}}$, $i = 0, 1, \ldots, Q$, i.e.,

$$\breve{\mathbf{A}}_{i-} \triangleq \mathbf{A}(\theta_0, \theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_Q) \in \mathbb{C}^{N\times Q}. \tag{17}$$

Proof: See Appendix A. ■

An important conclusion from Proposition 1 is that the array response of given angles can be alternatively controlled with formula (15), by selecting the new parameters η_q rather than the INRs β_q, $q = 1, \ldots, Q$.

To better explain the above proposition, let us consider a uniformly spaced linear array (ULA) of $N = 10$ elements spaced by half wavelength. We fix the beam axis at $\theta_0 = 20°$ and assume



(a)

(b)

Fig. 2. Illustrations of Proposition 1. (a) The first case: $\beta_1 = 0.7559, \beta_2 = 1.2226, \beta_3 = 1.5186$. (b) The second case: $\beta_1 = 0.1689, \beta_2 = 1.2507, \beta_3 = 1.5199$.

that $Q = 3$ interferences impinge on the array from $\theta_1 = -35^\circ$, $\theta_2 = -20^\circ$ and $\theta_3 = -10^\circ$, respectively. In the first case, the INRs are set, respectively, as $\beta_1 = 0.7559\,(-1.2155\text{ dB})$, $\beta_2 = 1.2226\,(0.8730\text{ dB})$ and $\beta_3 = 1.5186\,(1.8145\text{ dB})$. The resulting pattern of the optimal beamformer $\mathbf{w}_\star$ is shown in Fig. 2(a), from which we can clearly observe that all response levels at θ_1, θ_2 and θ_3 are -40 dB. Accordingly, if we set $\eta_1 = 0.1022$, $\eta_2 = 0.0908$, $\eta_3 = 0.0747$, then the obtained pattern of $\mathbf{w}_{\mathrm{OP}}$ is completely the same as that of $\mathbf{w}_\star$, as shown in Fig. 2(a). In fact, it can be verified that $\mathbf{w}_{\mathrm{OP}} = c\mathbf{w}_\star$ with $c = 1.0412$. The rationality of Proposition 1 can thus be demonstrated by this example.

In order to draw forth the important result of Proposition 1, we change the INRs to $\beta_1 = 0.1689\,(-7.7238\text{ dB})$, $\beta_2 = 1.2507\,(0.9717\text{ dB})$ and $\beta_3 = 1.5199\,(1.8182\text{ dB})$. In this case, the responses at θ_1, θ_2 and θ_3 are -30 dB, -40 dB and -40 dB, respectively, as shown by the resulting pattern of the optimal weight vector $\mathbf{w}_\star$ in Fig. 2(b). It can be noticed that, compared with the first case, all three INRs have been changed, although we only adjust the response level at θ_1 from -40 dB to -30 dB while retaining the responses at θ_2 and θ_3. As a matter of fact, in this case we have $\eta_1 = 0.3231$, $\eta_2 = 0.0908$, and $\eta_3 = 0.0747$ such that $\mathbf{w}_{\mathrm{OP}} = c\mathbf{w}_\star$ with $c = 1.0394$ (details of the determination of these parameters will be discussed in the following two subsections). An interesting and important observation is that the parameters η_2 and η_3 are unchanged comparing to the first case. It appears that the response at θ_1 is only affected by η_1 and unrelated to η_2 and η_3. Moreover, the adjustment of the response level at θ_1 does not cause response variations at θ_2 and θ_3. In fact, these results are not occasional, but theoretically provable as shown in the sequel.

B. *The FARCOP Algorithm*

In the preceding subsection, a new equivalent parameterization of the optimal beamformer in terms of $\{\eta_1, \ldots, \eta_Q\}$ has been presented with the aid of OP operator, and numerical examples are given to verify the equivalence. In this section, the FARCOP algorithm is introduced by extending the parameterization to general cases, rather than the optimal beamformer.

For the sake of notational simplicity, we first define

$$\boldsymbol{\Psi}(\boldsymbol{\eta}) \triangleq \left(\mathbf{I} - \mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|0}\right) + \sum_{q=1}^{Q} \eta_q \mathbf{E}^{\mathrm{H}}_{q|\breve{\mathbf{A}}_{q-}} \tag{18}$$

where the subscript $\boldsymbol{\eta}$ is defined as

$$\boldsymbol{\eta} \triangleq [1, \eta_1, \eta_2, \ldots, \eta_Q]^{\mathrm{T}} \in \mathbb{C}^{Q+1}. \tag{19}$$

Hence, we have

$$\mathbf{w}_{\mathrm{OP}} = \boldsymbol{\Psi}(\boldsymbol{\eta})\mathbf{a}(\theta_0) \tag{20}$$

from which it is seen that a weight vector ($\mathbf{w}_{\mathrm{OP}}$ or $\mathbf{w}_\star$) which can adjust the response as desired at interference angles (i.e., $\theta_1, \ldots, \theta_Q$), can be obtained by performing a linear transformation of the quiescent weight vector $\mathbf{a}(\theta_0)$. Now, what we are interested in is: 1) whether such transformation can be extended to a general case, i.e., adjust the response levels at certain angles of a (arbitrarily) given weight vector by using the transformation, and 2) whether we can retain the response levels at some angles while adjusting the response(s) at other angle(s).

The expression (20) provides us a new insight into the array response control of multiple points. It indicates that the response levels at specific angles can be adjusted by carrying out a linear transformation on the given weight vector, which is taken as $\mathbf{w}_0$ in the above discussion. On the basis of (20), we next present the FARCOP algorithm, in which the array response can be controlled from any given weight vector $\mathbf{w}_{\mathrm{pre}}$, i.e., not necessarily be $\mathbf{w}_0$.

More specifically, for a given $\mathbf{w}_{\mathrm{pre}}$, we propose to find a new weight vector $\mathbf{w}_{\mathrm{new}}$ that is able to adjust the responses at $\theta_1, \theta_2, \ldots, \theta_Q$ to be $\rho_1, \rho_2, \ldots, \rho_Q$, respectively. To make the discussion meaningful, we assume that $\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_i) \neq 0$, $i = 0, 1, \ldots, Q$. In the proposed FARCOP algorithm, we conduct a linear transformation to $\mathbf{w}_{\mathrm{pre}}$ and obtain the new weight vector $\mathbf{w}_{\mathrm{new}}$ by

$$\mathbf{w}_{\mathrm{new}} = \boldsymbol{\Psi}(\boldsymbol{\eta})\mathbf{w}_{\mathrm{pre}} \tag{21}$$

where the transformation matrix $\boldsymbol{\Psi}(\boldsymbol{\eta})$ is given in (18), and η_q, $q = 1, \ldots, Q$, needs to be determined according to the response control specification.

To see the merits of (21), we recall the property of OP and obtain that

$$\mathbf{E}_{q|\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_0)=\mathbf{0},\ q=1,\ldots,Q \tag{22}$$

$$\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}\mathbf{a}(\theta_0)=\mathbf{0}. \tag{23}$$

Then, the new weight vector $\mathbf{w}_{\mathrm{new}}$ in (21) satisfies

$$\begin{aligned}\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{a}(\theta_0)&=\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\eta})\mathbf{a}(\theta_0)\\&=\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\left(\left(\mathbf{I}-\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}\right)+\sum_{q=1}^{Q}\eta_q^*\mathbf{E}_{q|\breve{\mathbf{A}}_{q-}}\right)\mathbf{a}(\theta_0)\\&=\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_0).\end{aligned} \tag{24}$$

In other words, the array response output, i.e., $\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta)$, keeps unaltered at θ_0, after carrying out the linear transformation operator, see Eqn. (21), to the previous weight $\mathbf{w}_{\mathrm{pre}}$. This facilitates the realization of array response control.

More importantly, for the given i and q satisfying $i,q=1,2\ldots,Q$, one can see that

$$\mathbf{E}_{i|\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_q)=\begin{cases}\mathbf{a}(\theta_{\mathbf{q}}), & \text{if } i=q\\ \mathbf{0}, & \text{if } i\neq q\end{cases} \tag{25}$$

and

$$\left(\mathbf{I}-\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}\right)\mathbf{a}(\theta_q)=\mathbf{0},\ \forall q=1,\ldots,Q. \tag{26}$$

Therefore, the weight vector $\mathbf{w}_{\mathrm{new}}$ in (21) satisfies

$$\begin{aligned}\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{a}(\theta_q)&=\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\eta})\mathbf{a}(\theta_q)\\&=\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\left(\left(\mathbf{I}-\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}\right)+\sum_{i=1}^{Q}\eta_i^*\mathbf{E}_{i|\breve{\mathbf{A}}_{i-}}\right)\mathbf{a}(\theta_q)\\&=\eta_q^*\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_q),\ q=1,\ldots,Q.\end{aligned} \tag{27}$$

From (27), we know that $\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{a}(\theta_q)$ is a simple scaling of the previous $\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_q)$ with scaling factor η_q^*, $q=1,\ldots,Q$.

Combining the results of (24) and (27), we obtain that

$$\begin{aligned}L_{\mathrm{new}}(\theta_q,\theta_0)&=\frac{|\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{a}(\theta_q)|^2}{|\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2}=|\eta_q^*|^2\frac{|\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_q)|^2}{|\mathbf{w}_{\mathrm{pre}}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2}\\&=|\eta_q|^2\cdot L_{\mathrm{pre}}(\theta_q,\theta_0),\ q=1,\ldots,Q.\end{aligned} \tag{28}$$

where $L_{\mathrm{pre}}(\theta,\theta_0)$ denotes the array response pattern of $\mathbf{w}_{\mathrm{pre}}$. Clearly, Eqn. (28) indicates that if the weight vector $\mathbf{w}_{\mathrm{pre}}$ is transformed by $\boldsymbol{\Psi}(\boldsymbol{\eta})$, then the response level at θ_q will be amplified by $|\eta_q|^2$. As a result, if we wish to adjust the response at θ_q to be

$$L_{\mathrm{new}}(\theta_q,\theta_0)=\rho_q,\ q=1,\ldots,Q \tag{29}$$

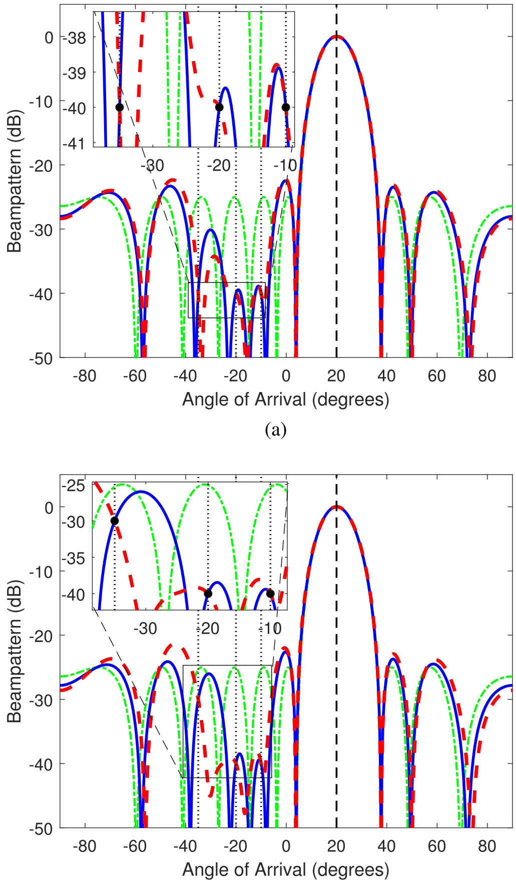
then we only need to simply select η_q as

$$|\eta_q|=\sqrt{\frac{\rho_q}{L_{\mathrm{pre}}(\theta_q,\theta_0)}},\ q=1,\ldots,Q. \tag{30}$$

Obviously, if we wish to keep the response level at θ_p unaltered after applying the transformation as (21), then η_q should be set as unitary amplitude, i.e., $|\eta_q|=1$. In addition, to make sure that $\mathbf{a}(\theta_0),\mathbf{a}(\theta_1),\ldots,\mathbf{a}(\theta_Q)$ are linearly independent, one can easily find that at most $N-1$ points can be controlled.

For illustration, we again use a 10-element ULA with half wavelength space to demonstrate the above results, by steering the beam to $\theta_0=20^\circ$ and taking the three controlled points as $\theta_1=-35^\circ$, $\theta_2=-20^\circ$ and $\theta_3=-10^\circ$. In this example, the previous weight $\mathbf{w}_{\mathrm{pre}}$ is taken as the Chebyshev weight vector with a -25 dB of sidelobe attenuation. In the first scenario, the desired levels of the controlled angles are set to be all -40 dB. According to (30), it can be readily obtained that $|\eta_1|=0.1851$, $|\eta_2|=0.1798$ and $|\eta_3|=0.1859$. Since the phases of η's can be arbitrarily set, two selections of $\{\eta_1,\eta_2,\eta_3\}$ are considered. More precisely, Fig. 3(a) shows the resulting patterns of two realizations, by taking $\eta_1=0.1851$, $\eta_2=0.1798$, $\eta_3=0.1859$ (corresponds to the blue solid line) and $\eta_1=0.1851e^{j\pi}$, $\eta_2=0.1798e^{j\pi/3}$, $\eta_3=0.1859e^{j\pi/4}$ (corresponds to the red dash line), respectively. Clearly, it is seen that these two selections of $\{\eta_1,\eta_2,\eta_3\}$ can result in identical responses at $\{\theta_1,\theta_2,\theta_3\}$, but the responses at other uncontrolled angles may be (significantly) different.

In the second scenario, the desired level of θ_1, i.e., ρ_1, is changed to be -30 dB, with all other settings staying unaltered as the first scenario. Comparing to the previous scenario, it can be computed that $|\eta_1|=0.5854$, while both $|\eta_2|$ and $|\eta_3|$ keep



(a)

(b)

Fig. 3. Illustrations of array response control initialized by a Chebyshev weight (the green lines represent the previous patterns corresponding to the given Chebyshev weights). (a) The first scenario: the blue solid line is obtained by taking $\eta_1=0.1851$, $\eta_2=0.1798$, $\eta_3=0.1859$, the red dash line is obtained by taking $\eta_1=0.1851e^{j\pi}$, $\eta_2=0.1798e^{j\pi/3}$, $\eta_3=0.1859e^{j\pi/4}$. (b) The second scenario: the blue solid line is obtained by taking $\eta_1=0.5854$, $\eta_2=0.1798$, $\eta_3=0.1859$, the red dash line is obtained by taking $\eta_1=0.5854e^{j\pi/2}$, $\eta_2=0.1798e^{j\pi/3}$, $\eta_3=0.1859e^{j\pi/4}$.

unchanged. Following the two sets of parameter selection on η_2 and η_3 in the first scenario, we examine $\eta_1 = 0.5854$ and $\eta_1 = 0.5854e^{j\pi/2}$, and plot the two resulting patterns in Fig. 3(b). One can see from Fig. 3(b) that the response control task has been fulfilled for the both sets of η_q, $q = 1, 2, 3$, but the resulting two patterns are not identical.

From the above discussion and simulation tests, a critical observation can be obtained is that the array responses at θ_q's can be readily and separately adjusted, by tuning the corresponding $|\eta_q|$'s by (30), if some of the desired levels at θ_q's, $q = 1, \ldots, Q$, are changed. Moreover, the phase of η_q affects the pattern shape of the uncontrolled angles, although it does not affect the response level at θ_q, i.e., $L_{\mathrm{new}}(\theta_q, \theta_0)$. For this reason, a more appropriate selection of the phase angles of η_q, $q = 1, \ldots, Q$, is developed under certain criterion to achieve a desired beampattern, as discussed in the next subsection.

C. Selection of $\boldsymbol{\eta}$ With Beampattern Consideration

In this subsection, we shall discuss how to select the parameter vector $\boldsymbol{\eta}$ under the consideration of beampattern, when we need to adjust the array responses at $\theta_1, \theta_2, \ldots, \theta_Q$ to their desired levels $\rho_1, \rho_2, \ldots, \rho_Q$, respectively, for a given weight vector $\mathbf{w}_{\mathrm{pre}}$ and its corresponding array response $L_{\mathrm{pre}}(\theta, \theta_0)$. From the previous analysis, it is clear that we can formulate the desired weight as $\mathbf{w}_{\mathrm{new}} = \boldsymbol{\Psi}(\boldsymbol{\eta})\mathbf{w}_{\mathrm{pre}}$ and realize the response control task by selecting a proper $\boldsymbol{\eta}$. We define a vector $\mathbf{r}$ for later use as

$$\mathbf{r} \triangleq [r_0, r_1, \ldots, r_Q]^{\mathrm{T}} \tag{31}$$

where $r_0 = 1$ and $r_q \triangleq |\eta_q|$ for $q = 1, \ldots, Q$.

As mentioned earlier, the modulus of η_q, i.e., $|\eta_q|$, can be readily obtained from (30), what remains is the determination of the phase angles of η_q, $q = 1, \ldots, Q$, which affect the shape of the resulting pattern at the uncontrolled region. Basically, how to optimize the vector $\boldsymbol{\eta}$ relies on specific considerations and applications. We next optimize $\boldsymbol{\eta}$ to maximize the white noise gain (WNG) [19]–[21], which is commonly used to measure the performance of a weight vector or its beampattern. For other kinds of object functions, similar procedures can be straightforwardly applied.

To proceed, we denote by G the WNG as

$$G = \frac{|\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2}{\mathbf{w}^{\mathrm{H}}\mathbf{w}}. \tag{32}$$

Therefore, given $\mathbf{w}_{\mathrm{pre}}$, θ_q and ρ_q, $q = 1, \ldots, Q$, the following constrained optimization problem is formulated:

$$\underset{\mathbf{w}_{\mathrm{new}}, \boldsymbol{\eta}}{\text{maximize}} \quad \frac{|\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2}{\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{w}_{\mathrm{new}}} \tag{33a}$$

$$\text{subject to} \quad L_{\mathrm{new}}(\theta_q, \theta_0) = \rho_q, \ q = 1, \ldots, Q \tag{33b}$$

$$\mathbf{w}_{\mathrm{new}} = \boldsymbol{\Psi}(\boldsymbol{\eta})\mathbf{w}_{\mathrm{pre}} \tag{33c}$$

$$\mathbf{e}_1^{\mathrm{T}}\boldsymbol{\eta} = 1 \tag{33d}$$

where $\mathbf{e}_1$ is a $(Q+1) \times 1$ vector with its first entry equal to 1 and 0 elsewhere. Recalling (24), $|\mathbf{w}_{\mathrm{new}}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2$ is a constant if $\mathbf{w}_{\mathrm{new}} = \boldsymbol{\Psi}(\boldsymbol{\eta})\mathbf{w}_{\mathrm{pre}}$. Moreover, note that $\mathbf{w}_{\mathrm{new}} = \boldsymbol{\Psi}(\boldsymbol{\eta})\mathbf{w}_{\mathrm{pre}}$ can be reshaped as

$$\mathbf{w}_{\mathrm{new}} = \mathbf{B}\boldsymbol{\eta} \tag{34}$$

where $\mathbf{B}$ is given by

$$\mathbf{B} = \left[\left(\mathbf{I} - \mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|0}\right)\mathbf{w}_{\mathrm{pre}}, \mathbf{E}^{\mathrm{H}}_{1|\breve{\mathbf{A}}_{1-}}\mathbf{w}_{\mathrm{pre}}, \ldots, \mathbf{E}^{\mathrm{H}}_{Q|\breve{\mathbf{A}}_{Q-}}\mathbf{w}_{\mathrm{pre}}\right] \tag{35}$$

and the constraint (33b) can be replaced by $|\eta_q| = r_q$, $q = 1, \ldots, Q$. Thus, the problem (33) can be recast as

$$\underset{\boldsymbol{\eta}}{\text{minimize}} \quad \boldsymbol{\eta}^{\mathrm{H}}\mathbf{B}^{\mathrm{H}}\mathbf{B}\boldsymbol{\eta} \tag{36a}$$

$$\text{subject to} \quad \boldsymbol{\eta}(1) = 1 \tag{36b}$$

$$|\boldsymbol{\eta}(q+1)| = r_q, \ q = 1, \ldots, Q. \tag{36c}$$

Rather than solving the problem (36), we first consider the following problem

$$\underset{\mathbf{y}}{\text{minimize}} \quad \mathbf{y}^{\mathrm{H}}\mathbf{B}^{\mathrm{H}}\mathbf{B}\mathbf{y} \tag{37a}$$

$$\text{subject to} \quad |\mathbf{y}(i+1)| = r_i, \ i = 0, 1, \ldots, Q. \tag{37b}$$

Notice that in (37), we have replaced $\mathbf{y}(1) = 1$ by a stronger constraint $|\mathbf{y}(1)| = 1$. Assume that $\mathbf{y}_\star$ is the optimal solution to (37), then the optimal solution to (36) is given by

$$\boldsymbol{\eta}_\star = \mathbf{y}_\star / \mathbf{y}_\star(1). \tag{38}$$

Accordingly, the ultimate selection of the weight vector of the problem (33) can be expressed as

$$\mathbf{w}_{\mathrm{new},\star} = \mathbf{B}\boldsymbol{\eta}_\star = \mathbf{B}\mathbf{y}_\star / \mathbf{y}_\star(1). \tag{39}$$

However, problem (37) is non-convex due to its constant-modulus constraints in (37b). Several solvers have been proposed to find the globally optimal solution of problem (37), which is NP-hard. For example, the SDR approach in [4] carries out a convex relaxation operator and obtains the ultimate solution by randomization. Nevertheless, this method has a high computational complexity and its optimality may not be well guaranteed. In [28], a monotonically error-bound improving technique (MERIT) and a power method-like iteration are devised for the quadratic optimization problem with constant-modulus constraints. As mentioned in [22], MERIT provides a sub-optimality guarantee that is sometimes tighter than that provided by SDR. The power-like iteration in [28] can be used to improve any initial estimate at a relatively low (second-order) cost, but the ultimate result depends a lot on initialization. To tackle the non-convex problem (37), we next adopt the gradient projection (GP) method. The computational complexity of GP is attractive. More importantly, its convergence property can be well-guaranteed for the original NP-hard problem (37).

GP or projected gradient descent method is a modified version of the conventional gradient descent method. It has been recently applied to solve unit-modulus least squares problems [22]. Following [22], the detailed procedure of GP method to solve problem (37) is given in Algorithm 1, where ε acts as the step size along the opposite direction of the gradient. Notice that, Algorithm 1 contains a similar procedure (3rd line) as the gradient descent method, with an additional step (4th line) conducting a projection onto a non-convex feasible set. In particular, the projection onto a constant modulus constraint admits a closed-form solution, and the entire procedure can be carried out very efficiently. In fact, the GP method reduces the objective

Algorithm 1: GP Method to Solve Problem (37).

```
$j = 0$, $\delta \in (0, 1)$, $\mathbf{y}_0 = \mathbf{r}$ and $\varepsilon = \frac{\delta}{\lambda_{\max}(\mathbf{B}^{\mathrm{H}}\mathbf{B})}$
while not convergence do
    obtain $\boldsymbol{\nu}$ as $\boldsymbol{\nu}=\mathbf{y}_j - \varepsilon \cdot \mathbf{B}^{\mathrm{H}}\mathbf{B}\mathbf{y}_j$ ------(Gradient)
    update $\mathbf{y}$ as $\mathbf{y}_{j+1}=\mathbf{r} \odot e^{j\angle\boldsymbol{\nu}}$ --------(Projection)
    set $j = j + 1$
end while
output $\mathbf{y}_\star = \mathbf{y}_j$
```

at each iteration step. Moreover, authors of [22] have obtained the following convergence properties about the GP method in Algorithm 1:

- (Global Convergence) The solution sequence $\{\mathbf{y}_j\}$ converges to a set $\mathcal{K}$ which consists of all the Karush-Kuhn-Tucker (KKT) points of problem (37).
- (Iteration Complexity) The Algorithm 1 converges to a KKT point at least sub-linearly.

Therefore, the solution sequence of GP algorithm converges to a meaningful point with attractive computational complexity.

Remark 1: If $\mathbf{w}_{\mathrm{pre}}$ and $\mathbf{a}(\theta_i)$, $i = 0, 1, \ldots, Q$, are conjugate centro-symmetric vectors [23], it can be verified that $\mathbf{B}^{\mathrm{H}}\mathbf{B}$ is a (real) symmetric matrix, by utilizing the fact that the inner product of two conjugate centro-symmetric vectors returns a real number (details are omitted due to space limitation). Then, the solution of problem (36) should be a real vector. In fact, under the above situation, we can see that the $\boldsymbol{\nu}$ in Algorithm 1 maintains to be a real vector if $\mathbf{y}_0 = \mathbf{r}$ is taken. Moreover, extensive examples show that the intermediate $\mathbf{y}_{j+1}$ in Algorithm 1 always equals to $\mathbf{r}$ (although it is challenging to theoretically prove), provided that $\mathbf{y}_0 = \mathbf{r}$ and $\delta \in (0, 1)$. This is because $\mathbf{I} - \varepsilon \cdot \mathbf{B}^{\mathrm{H}}\mathbf{B}$ approximates to be a diagonal matrix with positive diagonal elements if $\delta \in (0, 1)$. Then, the signs of elements in $\boldsymbol{\nu}$ are the same as that in $\mathbf{y}_j$. Consequently, for the conjugate centro-symmetric $\mathbf{w}_{\mathrm{pre}}$ and $\mathbf{a}(\theta_i)$, $i = 0, 1, \ldots, Q$, we can set $\boldsymbol{\eta}_\star$ as

$$\boldsymbol{\eta}_\star = \mathbf{r}. \tag{40}$$

Moreover, under the same conditions, it is not difficult to verify that $\mathbf{w}_{\mathrm{new},\star} = \mathbf{B}\boldsymbol{\eta}_\star$ is also conjugate centro-symmetric, which greatly facilitates the iteration of FARCOP algorithm if needed. Note that for centro-symmetric arrays, e.g., ULA, the steering vectors are conjugate centro-symmetric. In addition, it is interesting to note that the selection of $\boldsymbol{\eta}_\star$ in (40) also coincides with the results obtained in Fig. 2, where the resulting η's are all positive real values.

Remark 2: In the above discussions, the angles where responses need to be adjusted are known exactly. In fact, our algorithm can be readily extended to the case that the corresponding directions are not precisely known. This allows us to, for example, suppress interferences with inaccurate or rough orientations. In this scenario, we can replace the steering vector $\mathbf{a}(\theta_q)$ by the new vector $\mathbf{v}_q$, $q = 1, \ldots, Q$, in the construction of oblique projectors. The vector $\mathbf{v}_q$ can be selected as the qth ($q = 1, \ldots, Q$) principal eigenvector of the following matrix $\mathbf{R}$:

$$\mathbf{R} = \int_{\Omega} \mathbf{a}(\theta)\mathbf{a}^{\mathrm{H}}(\theta)d\theta \tag{41}$$

Algorithm 2: FARCOP Algorithm.

```
give $\theta_0$, $\mathbf{w}_{\mathrm{pre}}$, $L_{\mathrm{pre}}(\theta, \theta_0)$, $\theta_q$ and $\rho_q$, $q = 1, 2, \ldots, Q$
obtain $\mathbf{B}$ from (35)
calculate $r_q$ from (30) and then obtain $\mathbf{r}$ in (31)
if a centro-symmetric array is applied and $\mathbf{w}_{\mathrm{pre}}$ has a
    conjugate centro-symmetric structure, set $\boldsymbol{\eta}_\star$ by (40)
    directly, otherwise, solve problem (37) by GP
    algorithm (see Algorithm 1), and then obtain the
    optimal $\boldsymbol{\eta}_\star$ by (38)
obtain $\mathbf{w}_{\mathrm{new},\star}$ by (39) and calculate its corresponding
    response pattern $L_{\mathrm{new}}(\theta, \theta_0)$
```

where Ω represents the possible angle sectors of interferences, and the calculation of $\mathbf{R}$ can be approximated by a finite summation if no closed-form expression is available for the integration. Since $\mathbf{v}_q$'s have extracted the principal components of $\mathbf{R}$, we can apply FARCOP to shape broad nulls at the angle sector Ω, by simply replacing $\mathbf{a}(\theta_q)$ with $\mathbf{v}_q$ and then setting $\eta_{q,\star} = 0$, $q = 1, \ldots, Q$. Note that in this case, the specific value of Q may be obtained after the calculation of $\mathbf{R}$ in (41).

Once the ultimate weight vector $\mathbf{w}_{\mathrm{new},\star}$ in (39) is obtained, the proposed array response control via oblique projection (FARCOP) algorithm is completed. To summarize, we detail the steps of FARCOP in Algorithm 2.

D. Computational Complexity

In this part, we analyze the computational complexity of the proposed FARCOP algorithm. For centro-symmetric arrays, the main computation lies in the calculations of oblique projectors, i.e., $\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}$ and $\mathbf{E}_{q|\breve{\mathbf{A}}_{q-}}$'s, $q = 1, \ldots, Q$. For each oblique projector, the computational complexity is $O(N^3)$. Therefore, the computational complexity of FARCOP is $O(N^3)$ in this case. For the general arrays, an additional implementation of GP method is required. From the description of GP method in Algorithm 1, it has a low per-iteration computational complexity with $O(N)$, whose order is lower than that of the calculation of oblique projector. Consequently, the computation complexity of the proposed FARCOP algorithm is $O(N^3)$ for both the centro-symmetric arrays and the general ones.

IV. Comparisons of MA^{2}RC and FARCOP

The MA^{2}RC algorithm [17] is able to precisely adjust array responses at multiple points, from any given weight $\mathbf{w}_{\mathrm{pre}}$. In MA^{2}RC, the weight vector is updated as

$$\widetilde{\mathbf{w}}_{\mathrm{new}} = \mathbf{Z}_1 \left[(-\mathbf{F}^{\dagger}\mathbf{b} + \mathbf{f}_n)^{\mathrm{T}}, 1\right]^{\mathrm{T}}, \forall \mathbf{f}_n \in \mathcal{N}(\mathbf{F}) \tag{42}$$

where $\mathbf{F}$ and $\mathbf{b}$ are given by

$$\mathbf{F} = \begin{bmatrix} \mathbf{P}^{\perp}_{\mathbf{Z_2}}\mathbf{U_{1,2}} \\ \mathbf{P}^{\perp}_{\mathbf{Z}_3}\mathbf{U}_{1,2} \\ \vdots \\ \mathbf{P}^{\perp}_{\mathbf{Z}_Q}\mathbf{U}_{1,2} \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} \mathbf{P}^{\perp}_{\mathbf{Z_2}}\mathbf{w_1} \\ \mathbf{P}^{\perp}_{\mathbf{Z}_3}\mathbf{w}_1 \\ \vdots \\ \mathbf{P}^{\perp}_{\mathbf{Z}_Q}\mathbf{w}_1 \end{bmatrix} \tag{43}$$

and

$$\mathbf{Z}_q = \left[\mathbf{U}_{\mathbf{q},\mathbf{2}} \quad \mathbf{w}_q\right], \, q = 1, \ldots, Q \tag{44}$$

where $\mathbf{w}_q$ is obtained via A^{2}RC [15] as $\mathbf{w}_q = \mathbf{w}_{\text{pre}} + \mu_q \mathbf{a}(\theta_q)$ with μ_q determined by ρ_q. $\mathbf{U}_{q,2}$ is a submatrix of $\mathbf{U}_q$ as

$$\mathbf{U}_q = \Big[\underbrace{\mathbf{u}_1 \;\; \mathbf{u}_2}_{\mathbf{U}_{q,1}} \;\; \underbrace{\mathbf{u}_3 \;\; \ldots \;\; \mathbf{u}_N}_{\mathbf{U}_{q,2}}\Big] \tag{45}$$

which consists of the left singular vectors of $[\mathbf{a}(\theta_0), \, \mathbf{a}(\theta_q)]$ and can be determined through singular value decomposition (SVD) as $[\mathbf{a}(\theta_0), \, \mathbf{a}(\theta_q)] = \mathbf{U}_q \mathbf{\Sigma}_q \mathbf{V}_q^{\text{H}}$. Then, we can expand the weight $\widetilde{\mathbf{w}}_{\text{new}}$ of MA^{2}RC as

$$\begin{aligned} \widetilde{\mathbf{w}}_{\text{new}} &= \left[\mathbf{U}_{\mathbf{1},\mathbf{2}} \quad \mathbf{w}_1\right] \left[(-\mathbf{F}^{\dagger}\mathbf{b} + \mathbf{f}_n)^{\text{T}}, 1\right]^{\text{T}} \\ &= \mathbf{U}_{1,2}(-\mathbf{F}^{\dagger}\mathbf{b} + \mathbf{f}_n) + \mathbf{w}_{\text{pre}} + \mu_1 \mathbf{a}(\theta_1) \\ &= \mathbf{w}_{\text{pre}} + \underbrace{\left[\mathbf{U}_{\mathbf{1},\mathbf{2}} \quad \mathbf{a}(\theta_1)\right] \left[(-\mathbf{F}^{\dagger}\mathbf{b} + \mathbf{f}_n)^{\text{T}}, \mu_1\right]^{\text{T}}}_{\triangleq \widetilde{\mathbf{\Delta}}} \end{aligned} \tag{46}$$

where $\widetilde{\mathbf{\Delta}}$ denotes the added component to $\mathbf{w}_{\text{pre}}$.

For the proposed FARCOP algorithm, we follow the derivations of (63) and (64) in Appendix A and obtain that

$$\begin{aligned} \mathbf{E}_{q|\breve{\mathbf{A}}_{q-}}^{\text{H}} \mathbf{w}_{\text{pre}} &= \xi_q (\mathbf{I} - \mathbf{P}_{\breve{\mathbf{A}}_{q-}}) \mathbf{a}(\theta_q) \mathbf{a}^{\text{H}}(\theta_q) \mathbf{w}_{\text{pre}} \\ &= \xi_q \mathbf{a}^{\text{H}}(\theta_q) \mathbf{w}_{\text{pre}} \breve{\mathbf{A}} \mathbf{h}_q, \, q = 1, \ldots, Q \end{aligned} \tag{47}$$

$$\begin{aligned} (\mathbf{I} - \mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}^{\text{H}}) \mathbf{w}_{\text{pre}} &= (\mathbf{I} - \mathbf{P}_{\breve{\mathbf{A}}}^{\text{H}} + \mathbf{E}_{0|\breve{\mathbf{A}}_0}^{\text{H}}) \mathbf{w}_{\text{pre}} \\ &= \mathbf{P}_{\breve{\mathbf{A}}}^{\perp} \mathbf{w}_{\text{pre}} + \xi_0 \mathbf{a}^{\text{H}}(\theta_0) \mathbf{w}_{\text{pre}} \breve{\mathbf{A}} \mathbf{h}_0 \end{aligned} \tag{48}$$

where both ξ_i and $\mathbf{h}_i$, $i = 0, 1, \ldots, Q$, have been specified in Appendix A, see the definitions of (55), (62), (66) and (67) for details. Then, it is not hard to rewrite the $\mathbf{w}_{\text{new}}$ in (21) as

$$\mathbf{w}_{\text{new}} = \mathbf{\Psi}(\boldsymbol{\eta}) \mathbf{w}_{\text{pre}} = \mathbf{P}_{\breve{\mathbf{A}}}^{\perp} \mathbf{w}_{\text{pre}} + \breve{\mathbf{A}} \mathbf{H} \mathbf{\Lambda}_{\text{pre}} \boldsymbol{\eta} \tag{49}$$

where $\mathbf{H}$ is given in (70) in Appendix A, $\mathbf{\Lambda}_{\text{pre}} \in \mathbb{C}^{(Q+1)\times(Q+1)}$ is a generalization of $\mathbf{\Lambda}_0$ in (71) and can be detailed as

$$\mathbf{\Lambda}_{\text{pre}} = \text{Diag}\left([\xi_0 \mathbf{a}^{\text{H}}(\theta_0) \mathbf{w}_{\text{pre}}, \ldots, \xi_Q \mathbf{a}^{\text{H}}(\theta_Q) \mathbf{w}_{\text{pre}}]\right). \tag{50}$$

From (49), one can further expand the resulting $\mathbf{w}_{\text{new}}$ of the proposed FARCOP algorithm as

$$\begin{aligned} \mathbf{w}_{\text{new}} &= \mathbf{P}_{\breve{\mathbf{A}}}^{\perp} \mathbf{w}_{\text{pre}} + \breve{\mathbf{A}} \mathbf{H} \mathbf{\Lambda}_{\text{pre}} \boldsymbol{\eta} \\ &= (\mathbf{I} - \mathbf{P}_{\breve{\mathbf{A}}}) \mathbf{w}_{\text{pre}} + \breve{\mathbf{A}} \mathbf{H} \mathbf{\Lambda}_{\text{pre}} \boldsymbol{\eta} \\ &= \mathbf{w}_{\text{pre}} + \underbrace{\breve{\mathbf{A}} \left(\mathbf{H} \mathbf{\Lambda}_{\text{pre}} \boldsymbol{\eta} - (\breve{\mathbf{A}}^{\text{H}} \breve{\mathbf{A}})^{-1} \breve{\mathbf{A}}^{\text{H}} \mathbf{w}_{\text{pre}}\right)}_{\triangleq \mathbf{\Delta}} \end{aligned} \tag{51}$$

where $\mathbf{\Delta}$ stands for the appended component.

From (46) and (51), it is not difficult to find that

$$\widetilde{\mathbf{\Delta}} \in \left[\mathcal{R}^{\perp}([\mathbf{a}(\theta_0), \, \mathbf{a}(\theta_1)]) \oplus \mathcal{R}(\mathbf{a}(\theta_1))\right] \tag{52a}$$

$$\mathbf{\Delta} \in \mathcal{R}(\breve{\mathbf{A}}) \tag{52b}$$

which indicates that the modified quantities of the resulting weights (compared to the previous $\mathbf{w}_{\text{pre}}$) locate in two different sets, for MA^{2}RC and FARCOP. Note that we have $\mathbf{P}_{\breve{\mathbf{A}}}^{\perp} \mathbf{\Delta} = \mathbf{0}$ for FARCOP, while this may be not true for MA^{2}RC. Since the component in $\mathcal{R}^{\perp}(\breve{\mathbf{A}})$ does not affect the responses of the controlled points, one can see that no redundant component (i.e., component in $\mathcal{R}^{\perp}(\breve{\mathbf{A}})$) is added in FARCOP to the given $\mathbf{w}_{\text{pre}}$. The benefit may be the less pattern variations outside $\theta_i, i = 0, 1, \ldots, Q$.

Besides the above difference, for a given set of angles to be controlled, MA^{2}RC needs to re-calculate its weight if any of the desired levels changes. While for FARCOP, the manipulation is fairly simple since the re-calculation of $\mathbf{B}$ is unnecessary in this case. Moreover, notice from (43) that the matrix $\mathbf{F}$ in MA^{2}RC has a high dimension on its row. Thus, the computation of $\mathbf{F}^{\dagger}$ is computationally inefficient especially for a large array. In addition, the vector $\mathbf{f}_n$ in MA^{2}RC is optimized to obtain a less pattern variation. While for the proposed FARCOP algorithm, the parameter vector $\boldsymbol{\eta}$ is designed to maximize the WNG.

In brief, the main differences between the proposed FARCOP algorithm and the existing MA^{2}RC algorithm include:

- Two different manners of weight vector update are adopted in FARCOP and MA^{2}RC.
- FARCOP can adjust the response at a certain angle by changing the associated parameter η only, while MA^{2}RC needs to re-conduct all calculations.
- FARCOP is computationally more attractive. More specifically, the computation complexity of FARCOP is $O(N^3)$, while the computational complexity of MA^{2}RC is $O(QN^3)$.
- FARCOP takes WNG into account for parameter optimization, while MA^{2}RC does not.

V. Pattern Synthesis via FARCOP

In this section, the application of the FARCOP algorithm to pattern synthesis is briefly introduced. Generally speaking, the strategy herein shares a similar concept of pattern synthesis using MA^{2}RC [17] or multi-point OPARC [14], however, with improved flexibility and computational efficiency.

More specifically, we first specify an initial weight vector $\mathbf{w}_0$, and set the iteration index as $k = 1$. Multiple angles are 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 [14], for sidelobe synthesis, we select Q_k ($Q_k \leq N - 1$) peak angles where the response differences (from the desired levels) are relatively large. For mainlobe synthesis, we choose few discrete angles where the responses result significant deviations from the desired values. Once those angles are picked out, the FARCOP algorithm is 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 procedure until the response pattern is satisfactorily synthesized.

Note that at most $N - 1$ points can be selected and controlled in each step, and the number of the selected angles (denoted as Q_k) can be different in each step. Moreover, following [14], we can further reduce the computation cost by decreasing the number of the controlled angles, which facilitates the pattern synthesis for large arrays as shown in simulations later. Finally, we summarize the FARCOP based pattern synthesis algorithm in Algorithm 3.

Algorithm 3: FARCOP Based 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 FARCOP (see 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 $L_k(\theta, \theta_0)$
```

VI. Numerical Results

In this section, simulations are presented to demonstrate FARCOP and its application to array pattern synthesis. Unless otherwise specified, we take $\mathbf{f}_n$ as a zero vector for MA^{2}RC.

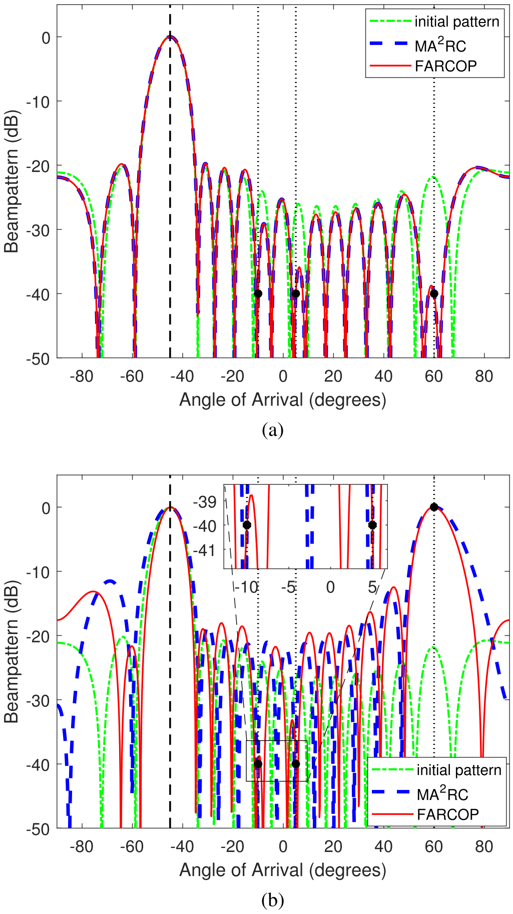
A. *Illustration of FARCOP Algorithm*

1) Precise Array Response Control Based on the Taylor Weight Vector: In this part, a linearly half-wavelength-spaced array with $N = 16$ isotropic elements is considered. We steer the beam to $\theta_0 = -45^\circ$ and set $\mathbf{w}_{\rm pre}$ as the Taylor weight vector with a -20 dB of sidelobe attenuation. In this scenario, both $\mathbf{a}(\theta)$ and $\mathbf{w}_{\rm pre}$ are conjugate centro-symmetric vectors. We optimize the parameter vector of FARCOP via GP algorithm and compare the result with MA^{2}RC in [17].

In the first case, three angles, i.e., $\theta_1 = -10^\circ$, $\theta_2 = 5^\circ$ and $\theta_3 = 60^\circ$, are expected to be all -40 dB. Fig. 4(a) depicts the results of the two algorithms. One can see that the response levels of the prescribed points have been precisely controlled to their desired values for both algorithms, with slight pattern variations (comparing to the initial pattern) at the uncontrolled regions. For the proposed FARCOP algorithm, we obtain that $\eta_{1,\star} = r_1 = 0.2051$, $\eta_{2,\star} = r_2 = 0.2205$ and $\eta_{3,\star} = r_3 = 0.1242$, which is consistent with the results in Remark 1. Interestingly, the resulting beampatterns of FARCOP and MA^{2}RC are almost coincide in this case.

To make a further comparison, we keep the initial weight and the controlled points the same as the previous example, and carry out the second case by simply setting the desired levels as $\rho_1 = -40$ dB, $\rho_2 = -40$ dB and $\rho_3 = 0$ dB, respectively. Note that both ρ_1 and ρ_2 are still the same as the previous case. The obtained patterns are displayed in Fig. 4(b), where we can see that both the two tested algorithms complete the given response control task. The resulting parameters are $\eta_{1,\star} = r_1 = 0.2051$, $\eta_{2,\star} = r_2 = 0.2205$ and $\eta_{3,\star} = r_3 = 12.4240$, which coincides with the conclusion drawn in Remark 1. More importantly, only $\eta_{3,\star}$ has varied comparing to the first case. This shows that the array response levels can be flexibly and individually adjusted by the proposed FARCOP algorithm. From Fig. 4(b), we can see that the resulting mainlobes of the MA^{2}RC method are wider than those of FARCOP. Moreover, FARCOP results in a WNG of 9.3014 dB, which is higher than the corresponding 8.5086 dB for MA^{2}RC.

To show that the FARCOP algorithm works well for an increased number of controlled points, we take $Q = N - 1 = 15$ by adding new angles as specified in Table I, and set the desired level as -40 dB for all the angles to be controlled. Fig. 5



(a)

(b)

Fig. 4. Illustration of FARCOP on array response control. (a) The first case. (b) The second case.

TABLE I
Angle Setting and the Resulting Parameters When Controlling Array Responses From the Taylor Weight Vector

q	θ_q	r_q	$\eta_{q,\star}$	q	θ_q	r_q	$\eta_{q,\star}$
1	-10°	0.2051	0.2051	9	-3°	0.2617	0.2617
2	5°	0.2205	0.2205	10	12°	0.2449	0.2449
3	60°	0.1242	0.1242	11	20°	0.2204	0.2204
4	-80°	0.1414	0.1414	12	30°	0.2242	0.2242
5	-70°	0.3123	0.3123	13	40°	0.3004	0.3004
6	-28°	0.5207	0.5207	14	50°	0.2421	0.2421
7	-23°	0.1140	0.1140	15	70°	0.2981	0.2981
8	16°	0.1233	0.1233				

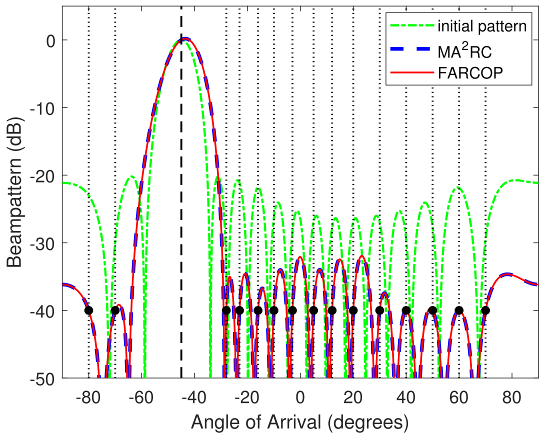


Fig. 5. Illustration of FARCOP on array response control for an increased number of controlled points.

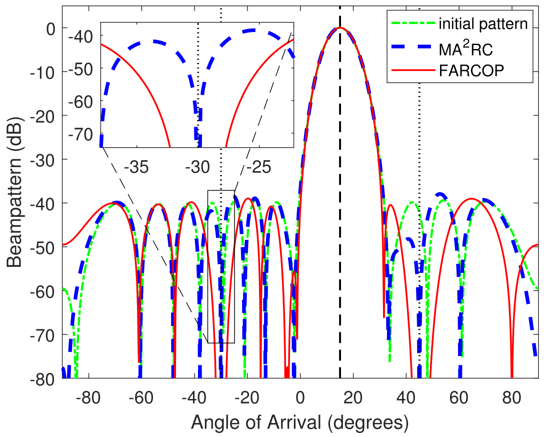


Fig. 6. Illustration of FARCOP on shaping broad nulls.

presents the resulting patterns and Table I lists the obtained parameters. Observe from Fig. 5 that the proposed algorithm has adjusted the response levels as desired. In addition, one can see from Table I that $\eta_{q,\star} = r_q$ $(q = 1, \ldots, Q)$, which follows the conclusion drawn in Remark 1. Thus, the advantage of the proposed FARCOP algorithm can be verified.

2) Shaping Broad Nulls Based on the Hamming Weight Vector: To validate that the proposed FARCOP algorithm is effective for an arbitrarily specified given weight vector, we consider the linearly half-wavelength-spaced array with $N = 16$ isotropic elements and set $\mathbf{w}_{\rm pre} = \mathbf{a}(\theta_0) \odot \mathbf{w}_h$, where $\theta_0 = 15^\circ$ is the main beam axis and $\mathbf{w}_h$ represents the 16-point Hamming window. Different from the previous example where the controlled points are precisely prescribed, in this test we consider interferences with rough directions. More specifically, all the possible interferences are assumed to locate in the sector $\Omega = [\theta_1 - 2^\circ, \theta_1 + 2^\circ] \cup [\theta_2 - 2^\circ, \theta_2 + 2^\circ]$ with $\theta_1 = -30^\circ$ and $\theta_2 = 45^\circ$. It is expected to form broad nulls at Ω with slight pattern variations (comparing to the beampattern of $\mathbf{w}_{\rm pre}$) outside Ω. Following Remark 2, we take $Q = 6$ in this scenario and obtain the ultimate pattern of FARCOP as depicted in Fig. 6. One can observe that the FARCOP algorithm results desirable beampattern with two deep broad nulls at Ω, while the MA^{2}RC method simply shapes two narrow nulls at θ_1 and θ_2, respectively.

B. Pattern Synthesis Using FARCOP

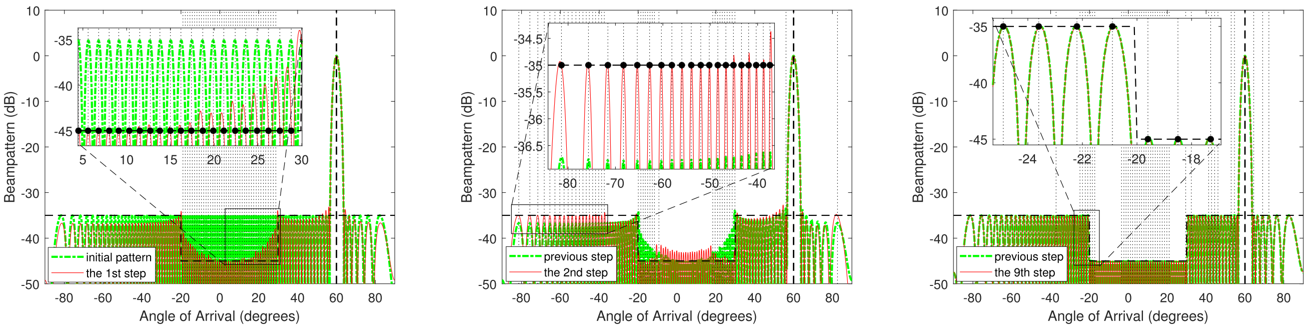
In this section, representative simulations are presented to illustrate the application of the proposed FARCOP algorithm to array pattern synthesis. Various approaches, including the convex programming (CP) method [25], the A^{2}RC method [15], the MA^{2}RC method [17] and the multiple-point OPARC method [14], are compared.

1) Nonuniform Sidelobe Synthesis for a Large ULA: In this example, pattern synthesis for a large linearly half-wavelength-spaced array with $N = 100$ isotropic elements is considered. The desired pattern steers to $\theta_0 = 60^\circ$ with a nonuniform sidelobe level. More specifically, the upper level is -45 dB in the sidelobe region $[-20^\circ, 30^\circ]$ and -35 dB in the rest of the sidelobe region. Clearly, the desired pattern is similar to the Chebyshev pattern with a -35 dB uniform sidelobe. For this reason, we take the initial weight of the FARCOP algorithm as the Chebyshev weight with a -35 dB of sidelobe attenuation, in the hope that the synthesis procedure can be simplified.

In this scenario, we select $Q_k = 41$ sidelobe peak angles in each step and then adjust their response levels to the desired ones by using the FARCOP algorithm. Since $Q \ll N$, the calculation in each step is greatly reduced. Moreover, the array considered in this case is centro-symmetric and the initial weight vector has a conjugate centro-symmetric structure. Following the analysis in Remark 1, in each step of response control we take the optimal parameter vector $\boldsymbol{\eta}_\star$ by (40) and obtain a weight vector having a closed-form expression. Thus, the computation complexity of the proposed algorithm is further reduced. Several intermediate results are presented in Fig. 7, from which one can see that only $k = 9$ steps are required to synthesize a satisfactory beampattern.

To have fairer comparisons, the same number of the iterations steps, i.e., $k = 9$ steps, is taken when conducting the MA^{2}RC method [17] and the multi-point OPARC method [14]. The number of the selected angles in each step is also set as $Q_k = 41$ for these two approaches. From the resulting sidelobe in Fig. 8, it is seen that the pattern envelope of the CP method is not aligned with the desired one, since it cannot exactly control the beampattern according to the required specifications. A careful observation shows that the A^{2}RC method (after carrying out 300 iteration steps) is outperformed by the MA^{2}RC and the FARCOP algorithms, both of which perform equally well. The obtained sidelobe of the multi-point OPARC algorithm is higher than the desired level at certain angles. This is mainly because that the multi-point OPARC algorithm is initialized by the quiescent pattern, rather than the Chebyshev pattern that is closer to the desired one. Thus, it may require more iteration steps for multi-point OPARC to synthesize a desirable beampattern.

To further assess the performance, some measurements of various algorithms are listed in Table II, where T represents the



(a)

(b)

(c)

Fig. 7. Resultant patterns at different steps when carrying out a nonuniform sidelobe synthesis for a nonuniform linear array. (a) Synthesized pattern at the 1st step. (b) Synthesized pattern at the 2nd step. (c) Synthesized pattern at the 9th step.

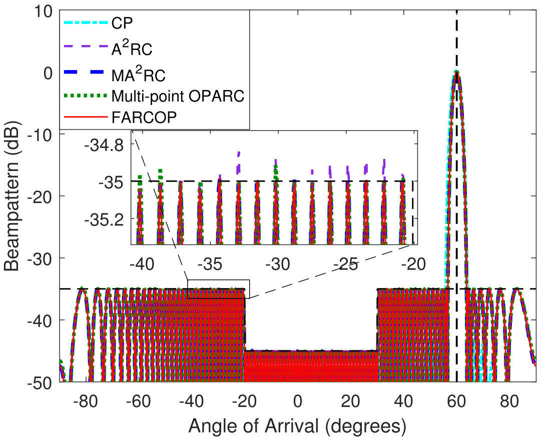


Fig. 8. Synthesized patterns for a large ULA.

TABLE II
MEASUREMENT COMPARISON WHEN CONDUCTING PATTERN SYNTHESIS FOR A LARGE ULA

	CP	A^{2}RC	MA^{2}RC	Multi-point OPARC	FARCOP
T(sec)	22.56	0.85	3.79	1.23	**0.15**
G(dB)	18.49	19.19	19.19	19.19	**19.19**

TABLE III
PARAMETERS OF THE NONISOTROPIC RANDOM ARRAY AND THE OBTAINED WEIGHTINGS BY FARCOP ALGORITHM

n	$x_n(\lambda)$	$l_n(\lambda)$	ζ_n(deg)	w_n
1	0.00	0.30	0.0	$0.2286e^{-j0.5077}$
2	0.45	0.25	-4.0	$0.4067e^{+j1.1104}$
3	0.95	0.24	5.0	$0.2989e^{+j2.2288}$
4	1.50	0.20	-32	$0.5605e^{+j3.1325}$
5	2.04	0.26	-3.2	$0.4971e^{-j1.9015}$
6	2.64	0.27	10	$0.6629e^{-j0.6966}$
7	3.09	0.23	1.0	$0.5781e^{+j0.6136}$
8	3.55	0.24	-10	$0.6496e^{+j1.2708}$
9	4.05	0.25	0.0	$0.7850e^{+j2.3602}$
10	4.55	0.21	7.0	$1.0000e^{-j2.6879}$
11	5.06	0.20	5.0	$0.9296e^{-j1.6905}$
12	5.50	0.20	5.0	$0.9463e^{-j0.7415}$
13	6.01	0.29	4.0	$0.6760e^{+j0.3764}$
14	6.53	0.20	5.0	$0.7157e^{+j1.6457}$
15	7.07	0.26	-9.0	$0.6158e^{+j2.6468}$
16	7.52	0.21	7.0	$0.6295e^{-j2.8228}$
17	8.00	0.25	10	$0.4447e^{-j1.7589}$
18	8.47	0.21	6.0	$0.5811e^{-j0.5151}$
19	8.98	0.20	-8.0	$0.3871e^{+j0.5235}$
20	9.53	0.26	0.0	$0.2498e^{+j1.3010}$
21	10.01	0.25	5.0	$0.2517e^{+j2.8088}$

execute time, G measures the resulting WNG. From Table II, one can see that the MA^{2}RC method and the CP method take long time to complete the synthesis task. In contrast, if the proposed FARCOP algorithm adopted, it converges the fastest, and meanwhile, performs well in WNG.

2) Uniform Sidelobe Synthesis for a Nonisotropic Random Array: In this example, a 21-element nonisotropic linear random array, see e.g., [11] and [15], is considered. The pattern of the nth element is given by

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

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. The beam axis is $\theta_0 = 20^\circ$ and the desired pattern has a -25 dB uniform sidelobe. Taking the quiescent weight vector $\mathbf{a}(\theta_0)$ as the initial one and implementing $k = 30$ synthesis steps, a satisfactory pattern is obtained by FARCOP and the corresponding weightings are listed in Table III.

The resulting beampatterns are depicted in Fig. 9. It is clearly seen that both the A^{2}RC algorithm (with 200 iteration steps) and the multi-point OPARC algorithm (with the same step number as FARCOP) produce responses that are higher than the desired values at certain sectors, see e.g., the region $[-5^\circ, 10^\circ]$. It may require more synthesis steps for these two methods to achieve satisfactory beampatterns. The CP method leads to a pattern with sidelobe much lower than the prescribed value. For the proposed FARCOP based pattern synthesis algorithm, Fig. 9 shows that its envelope of the synthesized pattern is aligned with the desired one. Moreover, it takes shorter time than MA^{2}RC (with the same step number as FARCOP) to synthesize a satisfactory beampattern.

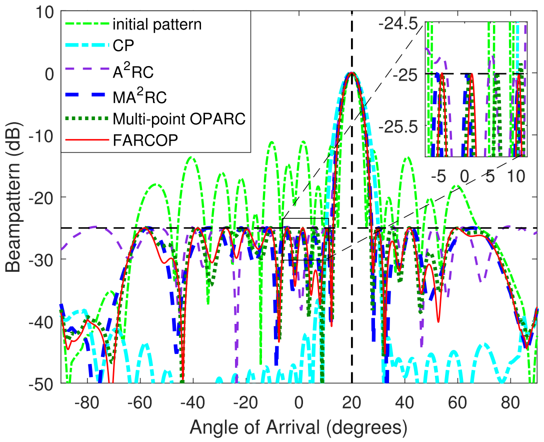


Fig. 9. Synthesized patterns with uniform sidelobe for a nonisotropic array.

VII. Conclusions

In this paper, we have presented an algorithm of flexible array response control via oblique projection (FARCOP). The proposed algorithm stems from the adaptive array theory, and is able to flexibly, precisely and simultaneously adjust the array response levels at multiple angles based on an arbitrarily given weight vector. In FARCOP, the parameters are independent of each other, and the response levels at the control angles can be either individually or jointly adjusted. We have also considered the optimization of white noise gain (WNG) by using the gradient projection (GP) algorithm to determine the parameters, and have derived a closed-form solution for the centro-symmetric array. In addition, we have discussed the application of FARCOP to array pattern synthesis. Representative simulations have been presented to verify the effectiveness and superiority of FARCOP under various scenarios.

Appendix A
Proof of Proposition 1

Suppose that $Q \leq N-1$ (otherwise, $\mathbf{a}(\theta_0),\ldots,\mathbf{a}(\theta_Q)$ are linearly dependent), then the following five steps are required in the derivation of Proposition 1.

Step 1: In the first step, $\mathbf{E}^{\mathrm{H}}_{q|\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_0)$, $q=1,\ldots,Q$, are simplified. Recalling (3), we have

$$\begin{aligned}\mathbf{E}_{q|\breve{\mathbf{A}}_{q-}} &= \mathbf{a}(\theta_q)(\mathbf{a}^{\mathrm{H}}(\theta_q)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_q))^{-1}\mathbf{a}^{\mathrm{H}}(\theta_q)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{q-}}\\ &= \xi_q\mathbf{a}(\theta_q)\mathbf{a}^{\mathrm{H}}(\theta_q)(\mathbf{I}-\mathbf{P}_{\breve{\mathbf{A}}_{q-}}) \qquad (54)\end{aligned}$$

where

$$\xi_q = (\mathbf{a}^{\mathrm{H}}(\theta_q)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_q))^{-1} \in \mathbb{R} \qquad (55)$$

$$\mathbf{P}_{\breve{\mathbf{A}}_{q-}} = \breve{\mathbf{A}}_{q-}(\breve{\mathbf{A}}^{\mathrm{H}}_{q-}\breve{\mathbf{A}}_{q-})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}_{q-}. \qquad (56)$$

From (54), we have

$$\begin{aligned}\mathbf{E}^{\mathrm{H}}_{q|\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_0) &= \xi_q(\mathbf{I}-\mathbf{P}_{\breve{\mathbf{A}}_{q-}})\mathbf{a}(\theta_q)\mathbf{a}^{\mathrm{H}}(\theta_q)\mathbf{a}(\theta_0)\\ &= \xi_q\cdot v(q,0)\cdot[\mathbf{a}(\theta_q)-\mathbf{P}_{\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_q)]\\ &= [\mathbf{a}(\theta_q),\ \breve{\mathbf{A}}_{q-}]\mathbf{d}_q \qquad (57)\end{aligned}$$

where

$$\mathbf{d}_q = \xi_q\cdot v(q,0)\begin{bmatrix}1\\ -(\breve{\mathbf{A}}^{\mathrm{H}}_{q-}\breve{\mathbf{A}}_{q-})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}_{q-}\mathbf{a}(\theta_q)\end{bmatrix} \in \mathbb{C}^{Q+1}. \qquad (58)$$

One can see that

$$\left[\mathbf{a}(\theta_q),\ \breve{\mathbf{A}}_{q-}\right] = \breve{\mathbf{A}}\mathbf{J}_q \qquad (59)$$

where $\mathbf{J}_q$ is a permutation matrix as

$$\mathbf{J}_q = \left[\begin{array}{c:c:c} & \mathbf{I}_q & \\ \hdashline 1 & & \\ \hdashline & & \mathbf{I}_{Q-q}\end{array}\right] \in \mathbb{R}^{(Q+1)\times(Q+1)}. \qquad (60)$$

Moreover, we have

$$\mathbf{J}_q\mathbf{d}_q = \xi_q\cdot v(q,0)\mathbf{h}_q \qquad (61)$$

where

$$\mathbf{h}_q = [c^{(q)}_1,\ldots,c^{(q)}_q,1,c^{(q)}_{q+1},\ldots,c^{(q)}_Q]^{\mathrm{T}} \in \mathbb{C}^{Q+1} \qquad (62)$$

with the notation $c^{(q)}_i$ standing for the ith element of the vector $-(\breve{\mathbf{A}}^{\mathrm{H}}_{q-}\breve{\mathbf{A}}_{q-})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}_{q-}\mathbf{a}(\theta_q)$, $i=1,\ldots,Q$.

Combining (57), (59) and (61), we can further simplify $\mathbf{E}^{\mathrm{H}}_{q|\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_0)$ as

$$\mathbf{E}^{\mathrm{H}}_{q|\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_0) = \breve{\mathbf{A}}\mathbf{J}_q\mathbf{d}_q = \xi_q\cdot v(q,0)\breve{\mathbf{A}}\mathbf{h}_q. \qquad (63)$$

Step 2: In this step, we attempt to make a simplification for $(\mathbf{I}-\mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|0})\mathbf{a}(\theta_0)$. Since $\mathbf{E}_{\breve{\mathbf{A}}_{0-}|0}+\mathbf{E}_{0|\breve{\mathbf{A}}_{0-}} = \mathbf{P}_{\breve{\mathbf{A}}}$, we have

$$\begin{aligned}(\mathbf{I}-\mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|0})\mathbf{a}(\theta_0) &= (\mathbf{I}-\mathbf{P}^{\mathrm{H}}_{\breve{\mathbf{A}}}+\mathbf{E}^{\mathrm{H}}_{0|\breve{\mathbf{A}}_{0-}})\mathbf{a}(\theta_0)\\ &= \mathbf{E}^{\mathrm{H}}_{0|\breve{\mathbf{A}}_{0-}}\mathbf{a}(\theta_0) \qquad (64)\end{aligned}$$

where the equality $\mathbf{P}^{\mathrm{H}}_{\breve{\mathbf{A}}}\mathbf{a}(\theta_0)=\mathbf{a}(\theta_0)$ has been used. Recalling the result of (63) in Step 1, we can similarly obtain that

$$\mathbf{E}^{\mathrm{H}}_{0|\breve{\mathbf{A}}_{0-}}\mathbf{a}(\theta_0) = \xi_0\|\mathbf{a}(\theta_0)\|^2_2\breve{\mathbf{A}}\mathbf{h}_0 \qquad (65)$$

where

$$\xi_0 = \left(\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{0-}}\mathbf{a}(\theta_0)\right)^{-1} \qquad (66)$$

$$\mathbf{h}_0 = [1,c^{(0)}_1,\ldots,c^{(0)}_Q]^{\mathrm{T}} \in \mathbb{C}^{Q+1} \qquad (67)$$

with $c^{(0)}_i$ being the ith element of $-(\breve{\mathbf{A}}^{\mathrm{H}}_{0-}\breve{\mathbf{A}}_{0-})^{-1}\breve{\mathbf{A}}^{\mathrm{H}}_{0-}\mathbf{a}(\theta_0)$, $i=1,\ldots,q$. Thus,

$$(\mathbf{I}-\mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|0})\mathbf{a}(\theta_0) = \xi_0\|\mathbf{a}(\theta_0)\|^2_2\breve{\mathbf{A}}\mathbf{h}_0. \qquad (68)$$

Step 3: In this step, a more compact expression of the $\mathbf{w}_{\mathrm{OP}}$ in (15) is obtained. More specifically, we combine (63) and (68), and obtain that

$$\begin{aligned}\mathbf{w}_{\mathrm{OP}} &= \left(\left(\mathbf{I}-\mathbf{E}^{\mathrm{H}}_{\breve{\mathbf{A}}_{0-}|0}\right)+\sum_{q=1}^{Q}\eta_q\mathbf{E}^{\mathrm{H}}_{q|\breve{\mathbf{A}}_{q-}}\right)\mathbf{a}(\theta_0)\\ &= \xi_0\|\mathbf{a}(\theta_0)\|^2_2\breve{\mathbf{A}}\mathbf{h}_0+\sum_{q=1}^{Q}\eta_q\cdot\xi_q v(q,0)\breve{\mathbf{A}}\mathbf{h}_q\\ &= \breve{\mathbf{A}}\mathbf{H}\boldsymbol{\Lambda}_0\boldsymbol{\eta} \qquad (69)\end{aligned}$$

where the coefficient matrix $\mathbf{H}\in\mathbb{C}^{(Q+1)\times(Q+1)}$, the diagonal matrix $\boldsymbol{\Lambda}_0\in\mathbb{C}^{(Q+1)\times(Q+1)}$ and the vector $\boldsymbol{\eta}\in\mathbb{C}^{Q+1}$ satisfy, respectively, as

$$\mathbf{H} = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_Q] = \begin{bmatrix} 1 & c_1^{(1)} & \cdots & c_1^{(Q)} \\ c_1^{(0)} & 1 & \cdots & c_2^{(Q)} \\ \vdots & \vdots & \ddots & \vdots \\ c_Q^{(0)} & c_Q^{(1)} & \cdots & 1 \end{bmatrix} \tag{70}$$

$$\mathbf{\Lambda}_0 = \mathrm{Diag}([\xi_0 v(0,0), \xi_1 v(1,0), \ldots, \xi_Q v(Q,0)]) \tag{71}$$

$$\boldsymbol{\eta} = [1, \eta_1, \eta_2, \ldots, \eta_Q]^{\mathrm{T}} \in \mathbb{C}^{Q+1}. \tag{72}$$

It is not hard to see that

$$\breve{\mathbf{A}}\mathbf{h}_q = (\mathbf{I} - \mathbf{P}_{\breve{\mathbf{A}}_{q-}})\mathbf{a}(\theta_q) = \mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{q-}}\mathbf{a}(\theta_q). \tag{73}$$

Then, we can rewrite $\breve{\mathbf{A}}\mathbf{H}$ as

$$\breve{\mathbf{A}}\mathbf{H} = \left[\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{0-}}\mathbf{a}(\theta_0), \mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{1-}}\mathbf{a}(\theta_1), \ldots, \mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{Q-}}\mathbf{a}(\theta_Q)\right] \triangleq \mathbf{Y}. \tag{74}$$

Step 4: In this step, we will show that $\mathbf{Y}$ in (74) has a full column rank, i.e.,

$$\mathrm{rank}(\mathbf{Y}) = Q + 1 \tag{75}$$

provided that $\mathbf{a}(\theta_0), \mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_Q)$ are linearly independent or equivalently $\breve{\mathbf{A}}$ has a full column rank.

To do so, we first assume that

$$\mathrm{rank}(\mathbf{Y}) < Q + 1 \tag{76}$$

or equivalently, the columns of $\mathbf{Y}$ are linearly dependent. It is implied from (76) that there exists a non-zero coefficient vector $\boldsymbol{\varsigma} = [\varsigma_0, \varsigma_1, \ldots, \varsigma_Q]^{\mathrm{T}}$ such that $\mathbf{Y}\boldsymbol{\varsigma} = \mathbf{0}$, or equivalently

$$\sum_{i=0}^{Q} \varsigma_i \mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i) = \mathbf{0}. \tag{77}$$

Without loss of generalization, we assume that $\varsigma_m \neq 0$, where $m \in \{0, 1, \ldots, Q\}$. Then, we can reshape (77) as

$$-\varsigma_m \mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{m-}}\mathbf{a}(\theta_m) = \sum_{i=0}^{m-1} \varsigma_i \mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i) + \sum_{i=m+1}^{Q} \varsigma_i \mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i). \tag{78}$$

It is important to note that $\mathbf{a}(\theta_m) \in \mathcal{R}(\breve{\mathbf{A}}_{i-})$ for $0 \leq i \leq Q$ and $i \neq m$. Then, we have

$$\mathbf{a}^{\mathrm{H}}(\theta_m)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}} = \mathbf{0}, \text{ for } 0 \leq i \leq Q,\ i \neq m. \tag{79}$$

Multiplied by $\mathbf{a}^{\mathrm{H}}(\theta_m)$ to both sides from the left of (78) yields

$$-\varsigma_m \mathbf{a}^{\mathrm{H}}(\theta_m)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{m-}}\mathbf{a}(\theta_m) = 0. \tag{80}$$

Recalling the assumption that $\varsigma_m \neq 0$, Eqn. (80) implies that

$$\mathbf{a}^{\mathrm{H}}(\theta_m)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{m-}}\mathbf{a}(\theta_m) = \|\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{m-}}\mathbf{a}(\theta_m)\|_2^2 = 0 \tag{81}$$

which yields $\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{m-}}\mathbf{a}(\theta_m) = \mathbf{0}$ or $\mathbf{a}(\theta_m) \in \mathcal{R}(\breve{\mathbf{A}}_{m-})$, and then contradicts with the assumption that $\breve{\mathbf{A}}$ has a full column rank. Thus, we learn that (76) cannot be established. One readily derives that $\mathrm{rank}(\mathbf{Y}) = Q + 1$ since $Q + 1 \leq N$. This completes the proof of (75).

Step 5: In the last step, we will complete the proof of Proposition 1 by showing that $\mathbf{H}\mathbf{\Lambda}_0$ is invertible, provided that $\mathbf{a}(\theta_0), \mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_Q)$ are linearly independent and $v(q,0) \neq 0$ for $\forall q = 1, \ldots, Q$.

To begin with, it is noted from the result of Step 4 that $\mathbf{Y}$ has a full column rank. Since $\mathbf{Y} = \breve{\mathbf{A}}\mathbf{H}$ and $\breve{\mathbf{A}}$ has a full column rank, we learn that $\mathbf{H}$ is invertible.

On the other hand, $\breve{\mathbf{A}}$ having full column rank implies that $\mathbf{a}(\theta_i) \notin \mathcal{R}(\breve{\mathbf{A}}_{i-})$, or equivalently $\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i) \neq \mathbf{0}$, for $\forall i = 0, 1, \ldots, Q$. Then, we know that

$$\xi_i = (\mathbf{a}^{\mathrm{H}}(\theta_i)\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i))^{-1} = (\|\mathbf{P}^{\perp}_{\breve{\mathbf{A}}_{i-}}\mathbf{a}(\theta_i)\|_2^2)^{-1} > 0 \tag{82}$$

holds true for $\forall i = 0, 1, \ldots, Q$. In addition, if $v(q,0) \neq 0$ for $\forall q = 1, \ldots, Q$, all the diagonal element of $\mathbf{\Lambda}_0$ in (71) will be non-zero. Thus, $\mathbf{\Lambda}_0$ is invertible.

Thanks to the fact that both $\mathbf{H}$ and $\mathbf{\Lambda}_0$ are invertible, one can see that $\mathbf{H}\mathbf{\Lambda}_0$ is invertible.

Now recalling the weight $\mathbf{w}_{\star}$ in (12), we know that for $\forall \beta_1, \ldots, \beta_Q$, there exists a corresponding $\mathbf{u}$ in (13) such that $\mathbf{w}_{\star} = \breve{\mathbf{A}}\begin{bmatrix} 1 & \mathbf{u}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Since $\mathbf{H}\mathbf{\Lambda}_0$ is invertible, from (69) one can verify that there must exist $\eta_1, \ldots, \eta_Q$ or a vector $\boldsymbol{\eta}$ satisfying

$$\boldsymbol{\eta} = [1, \eta_1, \eta_2, \ldots, \eta_Q]^{\mathrm{T}} = h^{-1}(\mathbf{H}\mathbf{\Lambda}_0)^{-1}\begin{bmatrix} 1 & \mathbf{u}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{83}$$

and a corresponding $c = h^{-1}$, such that

$$\mathbf{w}_{\mathrm{OP}} = \breve{\mathbf{A}}\mathbf{H}\mathbf{\Lambda}_0\boldsymbol{\eta} = h^{-1}\breve{\mathbf{A}}\begin{bmatrix} 1 & \mathbf{u}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = c\mathbf{w}_{\star} \tag{84}$$

where h is the first entry of $(\mathbf{H}\mathbf{\Lambda}_0)^{-1}\begin{bmatrix} 1 & \mathbf{u}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. This completes the proof of Proposition 1.

References

[1] R. J. Mailloux, *Phased Array Antenna Handbook*. Norwood, MA, USA: Artech House, 1994.

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

[3] 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.

[4] 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.

[5] 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.

[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] 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.

[8] B. Liao, C. Guo, L. Huang, Q. Li, and H. C. So, "Robust adaptive beamforming with precise main beam control," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 345–356, Feb. 2017.

[9] J. Liu, A. B. Gershman, Z.-Q. Luo, and K. M. Wong, "Adaptive beamforming with sidelobe control: A second-order cone programming approach," *IEEE Signal Process. Lett.*, vol. 10, no. 11, pp. 331–334, Nov. 2003.

[10] 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.

[11] C. Y. 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.

[12] C. Y. Tseng and L. J. Griffiths, "A unified approach to the design of linear constraints in minimum variance adaptive beamformers," *IEEE Trans. Antennas Propag.*, vol. 40, no. 12, pp. 1533–1542, Dec. 1992.

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.

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

[20] H. Cox, R. Zeskind, and M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 10, pp. 1365–1376, Oct. 1987.

[21] R. C. Nongpiur and D. J. Shpak, "Synthesis of linear and planar arrays with minimum element selection," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5398–5410, Oct. 2014.

[22] J. Tranter, N. D. Sidiropoulos, X. Fu, and A. Swami, "Fast unit-modulus least squares with applications in beamforming," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2875–2887, Jun. 2017.

[23] M. Haardt and J. A. Nossek, "Unitary ESPRIT: how to obtain increased estimation accuracy with a reduced computational burden," *IEEE Trans. Signal Process.*, vol. 43, no. 5, pp. 1232–1242, May 1995.

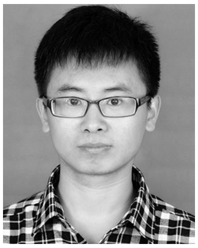
[24] P. Y. Zhou and M. A. Ingram, "Pattern synthesis for arbitrary arrays using an adaptive array method," *IEEE Trans. Antennas Propag.*, vol. 47, no. 5, pp. 862–869, May 1999.

[25] 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.

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

[27] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.

[28] M. Soltanalian and P. Stoica, "Designing unimodular codes via quadratic optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1221–1234, Mar. 2014.



Xuejing Zhang (S'17) was born in 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 working toward the Ph.D. degree in signal and information processing at the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. Since November 2017, he has been a visiting student at the University of Delaware, Newark, DE, USA. His research interests include array signal processing and wireless communications.

Zishu He (M'11) was born in Sichuan, China, in 1962. He received the B.S., M.S., and Ph.D. degrees in signal and information processing from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1984, 1988, and 2000, respectively.

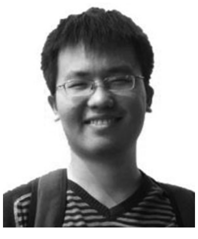
He is currently a Professor with the School of Information and Communication Engineering, UESTC. His current research interests include array signal processing, digital beam forming, the theory on multiple-input multiple-output (MIMO) communication and MIMO radar, adaptive signal processing, and interference cancellation.

Bin Liao (S'09–M'13–SM'16) received the B.Eng. and M.Eng degrees from Xidian University, Xi'an, China, in 2006 and 2009, respectively, and the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2013, all in electronic engineering. From September 2013 to January 2014, he was a Research Assistant with the Department of Electrical and Electronic Engineering, The University of Hong Kong. From August 2016 to October 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 research interests include sensor array processing, adaptive filtering, and convex optimization, with applications to radar, navigation, and communications. He is 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).

Dr. Liao is an Associate Editor of the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, *IET Signal Processing, Multidimensional Systems and Signal Processing*, and IEEE ACCESS.

Yue Yang (S'17) was born in 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. She is currently working toward the Ph.D. degree in signal and information processing at the School of Information and Communication Engineering, UESTC. Since January 2019, she has been a visiting student at the National University of Singapore, Singapore. Her research interests include synthetic aperture radar imaging, sparse signal reconstruction, and statistical signal processing.

Jinfeng Zhang received the B.S. degree in 2001, and the M.S. and the Ph.D. degrees in signal and information processing, in 2004 and 2017, respectively, all from Dalian University Technology, Dalian, China. From 2017 to 2018, she was a Visiting Scholar with the University of Delaware, Newark, DE, USA. She is currently on the Faculty of the College of Information Engineering, Shenzhen University, Shenzhen, China. She is also a member of Shenzhen Key Laboratory of Antennas and Propagation. Her research interests include array signal processing and non-Gaussian signal processing.



Xuepan Zhang was born in Hebei, China. He received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, and National Laboratory of Radar Signal Processing, Xi'an, China, in 2010 and 2015, respectively, both in electrical engineering. He is currently working as a Principal Investigator in Qian Xuesen Laboratory of Space Technology, Beijing, China. His research interests include synthetic aperture radar, ground moving target indication, and deep learning.