OPARC: Optimal and Precise Array Response Control Algorithm—Part II: Multi-Points and Applications

Xuejing Zhang, *Student Member, IEEE*, Zishu He, *Member, IEEE*, Xiang-Gen Xia, *Fellow, IEEE*, Bin Liao, *Senior Member, IEEE*, Xuepan Zhang, and Yue Yang, *Student Member, IEEE*

***Abstract*—In this paper, the optimal and precise array response control (OPARC) algorithm proposed in Part I of this two paper series is extended from single point to multi-points. Two computationally attractive parameter determination approaches are provided to maximize the array gain under certain constraints. In addition, the applications of the multi-point OPARC algorithm to array signal processing are studied. It is applied to realize array pattern synthesis (including the general array case and the large array case), multi-constraint adaptive beamforming, and quiescent pattern control, where an innovative concept of normalized covariance matrix loading is proposed. Finally, simulation results are presented to validate the effectiveness and good performance of the multi-point OPARC algorithm.**

***Index Terms*—Array response control, adaptive array theory, array pattern synthesis, adaptive beamforming, quiescent pattern control.**

I. Introduction

In THE companion paper [1], optimal and precise array response control (OPARC) algorithm was proposed and analyzed. OPARC provides a new mechanism to control array responses at a given set of angles, by simply assigning virtual interference one-by-one. The optimality (in the sense of array gain) of OPARC in each step is guaranteed. Nevertheless, OPARC only controls one point per step and may be inefficient if multiple points are needed to be precisely adjusted. Moreover, how to use the OPARC algorithm in practical cases (where real data commonly exists) remains.

This paper first extends the OPARC algorithm from single point response control per step to multi-point response control per step. Note that a multi-point accurate array response control (MA^{2}RC) algorithm has been recently developed in [2]. Nevertheless, since it is built on the basis of the accurate array response control (A^{2}RC) algorithm [3], the MA^{2}RC suffers from the similar drawbacks to A^{2}RC, i.e., a solution is empirically adopted and hence a satisfactory performance cannot be always guaranteed as analyzed in details in [1]. In this paper, we first carry out a careful investigation on the change rule of the optimal beamformer when multiple virtual interferences are simultaneously assigned. Then, a generalized methodology of the weight vector update is observed and utilized for the realization of the multi-point array response control. Similar to the OPARC in [1], we formulate a constrained optimization problem such that the array response levels of multiple points can be optimally (in the sense of array gain) and precisely controlled. Then, two different solvers, by either taking advantage of the OPARC algorithm or employing the recently developed consensus alternating direction method of multipliers (C-ADMM) approach in [4], are provided to find an approximate solution of the established optimization problem. Note that since the OPARC in [1] only optimally controls the array response at one point in each step, it has a closed-form solution, while this is not the case for the multi-point OPARC in this paper. In other words, this paper does not cover [1]. The differences between the proposed multi-point OPARC and MA^{2}RC are similar to those between OPARC and A^{2}RC as described in [1] in details. Meanwhile, for the proposed multi-point OPARC, its applications to, such as, array pattern synthesis, multi-constraint adaptive beamforming and quiescent pattern control, are also presented as detailed below.

Application to Array Pattern Synthesis: Determining the complex weights for array elements so as to achieve a desired beampattern is known as array pattern synthesis [5]–[7]. With regard to this problem, it is expected to control the sidelobe of array response to achieve a pencil beam pattern or to realize a shaped beam pattern complying to a given mask. Over the past several decades, a great number of pattern synthesis approaches have been proposed. For instance, the classical Dolph-Chebychev synthesis technique obtains an analytical expression for the

Manuscript received December 25, 2017; revised May 20, 2018 and September 17, 2018; accepted November 25, 2018. Date of publication December 7, 2018; date of current version December 21, 2018. 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 Grant 61671139, Grant 61671137, Grant 61771316, and Grant 61701499; in part by the Foundation of Shenzhen under Grant JCYJ20170302150044331; and in part by China Scholarship Council. *(Corresponding author: Xuejing Zhang.)*

X. Zhang is with the University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: xjzhang7@163.com).

Z. He and Y. Yang are with the University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: zshe@uestc.edu.cn; yueyang@std.uestc.edu.cn).

X.-G. Xia is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: xxia@ee.udel.edu).

B. Liao is with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: binliao@szu.edu.cn).

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

Digital Object Identifier 10.1109/TSP.2018.2885492

weightings that optimize the compromise between beamwidth and sidelobe level [8]. However, its application to arrays with arbitrary geometries or nonisotropic elements is not straightforward. Global optimization based methods, such as genetic algorithm (GA) [9], simulated annealing (SA) method [10] and particle swarm optimization (PSO) method [11], are applicable to non-uniformly spaced arrays. Nevertheless, their prohibitive computational complexities would limit the practical use. The approaches in [12] and [13], which are developed on the basis of the adaptive array theory [14], are applicable to arbitrary array configurations. However, some key parameters in these approaches are usually selected in an *ad hoc* way. Owing to the recent advances in convex optimization [15], optimization methods such as convex programming [16], second-order cone programming (SOCP) [17] and semidefinite relaxation (SDR) [18] have been employed in pattern synthesis. Note that such kind of approaches cannot control the beampattern precisely according to the required specifications. In this paper, the above shortcoming is overcome by synthesizing desirable patterns with the proposed multi-point OPARC algorithm. We start the synthesis procedure from the quiescent pattern, and iteratively adjust the responses of multiple angles to their desired levels. Simulation results show that it only requires a few steps of iteration to complete the syntheses of well-shaped beampatterns. In addition to the consideration for a general array, large array pattern synthesis problem [19], where the existing methods consume a large amount of computing resources or even not work at all, is particularly discussed. We will see that the large array pattern synthesis can be readily realized with the multi-point OPARC algorithm, in a computationally attractive manner.

Application to Multi-constraint Adaptive Beamforming: Adaptive beamforming plays an important role in various application areas, since it enables us to receive a desired signal from a particular direction while it simultaneously blocks undesirable interferences. Multi-constraint adaptive beamforming, i.e., designing an adaptive beamformer with several fixed directional constraints, is a common strategy to improve the robustness of the adaptive beamformer, see [20]–[22] for example. The existing methods may cause distorted beampatterns, due to their imperfections on model building or parameter optimization. Based on the proposed multi-point OPARC algorithm, a new approach to multi-constraint adaptive beamforming is presented in this paper. We modify the traditional adaptive beamformer to make the prescribed amplitude constraints satisfied by utilizing the multi-point OPARC algorithm. In the proposed algorithm, the total signal-to-interference-plus-noise ratio (SINR) (taking both real interferences and assigned virtual interferences into consideration) is maximized, and the real unexpected components can be well rejected without leading to any undesirable pattern distortion. Inspired by this, a new concept of normalized covariance matrix loading (NCL), which can be regarded as a generalization of the conventional diagonal loading (DL) in [23]–[25], is developed. Moreover, NCL is also exploited to realize quiescent pattern control as introduced next.

Application to Quiescent Pattern Control: In brief, when an adaptive array operates in the presence of white noise only, the resultant adaptive beamformer is referred to as the quiescent weight vector, and the corresponding array response is termed as the quiescent pattern. As pointed out in [26], having overall low sidelobes is important to adaptive arrays and how to specify a quiescent response pattern is worthwhile investigating. Most of the existing quiescent pattern control methods [26]–[28] are established on the foundation of the linearly constrained minimum variance (LCMV) framework, where the unnecessary phase constraints of array response are implicitly imposed. In this paper, a simple yet effective quiescent pattern control algorithm is proposed. We synthesize a satisfactory deterministic pattern, i.e., the ultimate quiescent pattern, by adopting the multi-point OPARC algorithm, and meanwhile, collect the resulting virtual normalized covariance matrix (VCM) for later use. Under the real data circumstance, the quiescent pattern control is completed by conducting a simple NCL operator to the existed VCM, and the weight vector can be obtained accordingly.

This paper is organized as follows. The proposed multi-point OPARC algorithm is presented in Section II. The three applications of the multi-point OPARC are discussed in Section III. Representative experiments are carried out in Section IV and conclusions are drawn in Section V.

Notations: Following the notations in [1], we use bold uppercase and lower-case letters to represent matrices and vectors, respectively. In particular, we use $\mathbf{I}$ to denote the identity matrix. $j \triangleq \sqrt{-1}$. $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ stand for the transpose and Hermitian transpose, respectively. $|\cdot|$ denotes the absolute value and $\|\cdot\|_2$ denotes the l_2 norm. We use $(\mathbf{g})_i$ to stand for the ith element of vector $\mathbf{g}$. $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts, respectively. $\oslash$ represents the element-wise division operator. We use $\mathrm{Diag}(\cdot)$ to stand for the diagonal matrix with the components of the input vector as the diagonal elements. $\mathbb{R}$ and $\mathbb{C}$ denote the sets of all real and all complex numbers, respectively. Finally, $\cup$ denotes the set union and $\mathrm{card}(\cdot)$ returns the number of elements in a set.

II. Multi-Point OPARC Algorithm

To present our multi-point OPARC algorithm, we first make a detailed analysis on the optimal weight vector.

A. Multi-Interference Optimal Beamformer

Consider an array with N elements. According to [1], the optimal weight vector:

$$\mathbf{w}_{\mathrm{opt}} = \mathbf{T}_{n+i}^{-1}\mathbf{a}(\theta_0) \tag{1}$$

maximizes both the output signal-to-interference-plus-noise ratio (SINR) and the array gain of an array system, where SINR and array gain are defined, respectively, as [14]

$$\mathrm{SINR} \triangleq \frac{\sigma_s^2|\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2}{\mathbf{w}^{\mathrm{H}}\mathbf{R}_{n+i}\mathbf{w}},\quad G \triangleq \frac{|\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2}{\mathbf{w}^{\mathrm{H}}\mathbf{T}_{n+i}\mathbf{w}} \tag{2}$$

where $\mathbf{a}(\theta)$ stands for the array steering vector:

$$\mathbf{a}(\theta) = [g_1(\theta)e^{-j\omega\tau_1(\theta)}, \ldots, g_N(\theta)e^{-j\omega\tau_N(\theta)}]^{\mathrm{T}} \tag{3}$$

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

notations, θ_0 is the beam axis, $\mathbf{R}_{n+i}$ denotes the $N \times N$ noise-plus-interference covariance matrix, $\mathbf{T}_{n+i}$ stands for the normalized covariance matrix satisfying

$$\mathbf{T}_{n+i} = \frac{\mathbf{R}_{n+i}}{\sigma_n^2} = \mathbf{I} + \sum_{l=1}^{Q} \beta_l \mathbf{a}(\theta_l)\mathbf{a}^{\mathrm{H}}(\theta_l) \tag{4}$$

where $\beta_l \triangleq \sigma_l^2/\sigma_n^2$ denotes the interference-to-noise ratio (INR), Q is the number of interferences, $\mathbf{a}(\theta_l)$ is the steering vector of the lth interference, σ_s^2, σ_n^2 and σ_l^2 stand for the powers of signal, noise and the lth interference, respectively. Note that G in (2) represents the amplification factor of the input signal-to-noise ratio (SNR) σ_s^2/σ_n^2, and the criterion of array gain maximization is adopted to achieve the optimal weight vector.

From (1)–(2), one can see that the optimal weight vector $\mathbf{w}_{\mathrm{opt}}$ depends on $\mathbf{R}_{n+i}$ or $\mathbf{T}_{n+i}$, which is normally data-dependent. For this reason, $\mathbf{R}_{n+i}$ or $\mathbf{T}_{n+i}$ may not be available if we need to design a data-independent array response pattern $L(\theta, \theta_0) \triangleq |\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta)|^2/|\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2$ that satisfies some specific requirements. In this case, for a given response design task, the concept of virtual normalized noise-plus-interference covariance matrix (VCM) was introduced in [1]. Moreover, it was shown in [1] that a VCM can be constructed by assigning suitable virtual interferences one-by-one. In this paper, for a given response control task, we assign multiple virtual interferences (instead of a single virtual interference) at one step, and study how the optimal weight vector in (1) changes.

We use induction to describe the problem. Suppose that we have already assigned interferences for $(k-1)$ times, the total number of interferences is accumulated as Q_{k-1} and $\mathbf{T}_{k-1}$ denotes the total VCM up to the $(k-1)$th step. The corresponding optimal weight vector at the $(k-1)$th step is given by

$$\mathbf{w}_{k-1} = \mathbf{T}_{k-1}^{-1}\mathbf{a}(\theta_0) \tag{5}$$

where the subscript $(\cdot)_{\mathrm{opt}}$ has been omitted for notational simplicity. Then, we carry out the kth step by assigning M_k interferences from directions $\theta_{k,m}$ with INR to be $\beta_{k,m}$, $m = 1, \ldots, M_k$, where $\theta_{k,m}$ are renamed from those θ_l in (4). Then,

$$\begin{aligned}\mathbf{T}_k &= \mathbf{T}_{k-1} + \sum_{m=1}^{M_k} \beta_{k,m}\mathbf{a}(\theta_{k,m})\mathbf{a}^{\mathrm{H}}(\theta_{k,m}) \\ &= \mathbf{T}_{k-1} + \mathbf{A}_k\mathbf{\Sigma}_k\mathbf{A}_k^{\mathrm{H}}\end{aligned} \tag{6}$$

where

$$\mathbf{A}_k = [\mathbf{a}(\theta_{k,1}), \ldots, \mathbf{a}(\theta_{k,M_k})] \tag{7}$$

$$\mathbf{\Sigma}_k = \mathrm{Diag}([\beta_{k,1}, \ldots, \beta_{k,M_k}]) \tag{8}$$

and $\mathbf{T}_k$ is the resulting VCM after implementing the kth step of the interference assigning. Clearly, if $M_k = 1$, (6) degenerates to Eqn. (9) of [1], and the related discussions return to our previous work in [1]. To make the discussion meaningful, the matrix $\mathbf{A}_k$ in this paper is assumed to have a full column rank.

By applying the Generalized Woodbury Lemma [29] to (6), we obtain that

$$\begin{aligned}\mathbf{T}_k^{-1} &= \mathbf{T}_{k-1}^{-1} \\ &\quad - \mathbf{T}_{k-1}^{-1}\mathbf{A}_k\left(\mathbf{I} + \mathbf{\Sigma}_k\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}\mathbf{A}_k\right)^{-1}\mathbf{\Sigma}_k\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}.\end{aligned} \tag{9}$$

Accordingly, the obtained optimal weight vector satisfies

$$\mathbf{w}_k = \mathbf{T}_k^{-1}\mathbf{a}(\theta_0) = \mathbf{w}_{k-1} + \mathbf{T}_{k-1}^{-1}\mathbf{A}_k\mathbf{h}_k \tag{10}$$

where $\mathbf{h}_k \in \mathbb{C}^{M_k}$ is

$$\mathbf{h}_k = -\left(\mathbf{I} + \mathbf{\Sigma}_k\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}\mathbf{A}_k\right)^{-1}\mathbf{\Sigma}_k\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}\mathbf{a}(\theta_0). \tag{11}$$

As shown in (10), the current optimal weight $\mathbf{w}_k$ is obtained by making a modification to the previous weight $\mathbf{w}_{k-1}$.

Recalling the adaptive array theory, the weight $\mathbf{w}_k$ performs optimally in maximizing the array gain G_k defined as

$$G_k \triangleq |\mathbf{w}_k^{\mathrm{H}}\mathbf{a}(\theta_0)|^2/|\mathbf{w}_k^{\mathrm{H}}\mathbf{T}_k\mathbf{w}_k| \tag{12}$$

although the response levels at $\theta_{k,m}$, $m = 1, \ldots, M_k$, may not reach their expected values. To precisely adjust the array responses of $\theta_{k,m}$ to their desired levels $\rho_{k,m}$, the INRs $\beta_{k,m}$, $m = 1, \ldots, M_k$, or equivalently the diagonal matrix $\mathbf{\Sigma}_k$, should be carefully selected. In the meantime, the array gain G_k in (12) should be maximized. Note also that $\mathbf{h}_k$ in (11) acts as a mapping of $\mathbf{\Sigma}_k$, and we can express $\mathbf{\Sigma}_k$ by $\mathbf{h}_k$ as

$$\mathbf{\Sigma}_k = \mathrm{Diag}\left(-\mathbf{h}_k \oslash \left(\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}\left(\mathbf{a}(\theta_0) + \mathbf{A}_k\mathbf{h}_k\right)\right)\right). \tag{13}$$

From (11) and (13), one can see that $\mathbf{\Sigma}_k$ and $\mathbf{h}_k$ are one-one mapping. Therefore, the multi-point optimal and precise array response control (OPARC) can be realized by either finding a suitable $\mathbf{\Sigma}_k$ or selecting an appropriate $\mathbf{h}_k$.

B. Multi-Point OPARC Problem Formulation

Let us first formulate the multi-point OPARC by optimizing $\mathbf{\Sigma}_k$ as:

$$\max_{\mathbf{\Sigma}_k} \; G_k = |\mathbf{w}_k^{\mathrm{H}}\mathbf{a}(\theta_0)|^2/|\mathbf{w}_k^{\mathrm{H}}\mathbf{T}_k\mathbf{w}_k| \tag{14a}$$

$$\text{subject to } \; L(\theta_{k,m}, \theta_0) = \rho_{k,m},\ m = 1, \ldots, M_k \tag{14b}$$

$$\mathbf{w}_k = \mathbf{w}_{k-1,\star} + \mathbf{T}_{k-1}^{-1}\mathbf{A}_k\mathbf{h}_k \tag{14c}$$

where $\mathbf{w}_{k-1,\star}$ is the resultant weight vector of the $(k-1)$th step (we use the star symbol to indicate it as the ultimate selection of $\mathbf{w}_{k-1}$), the vector $\mathbf{h}_k$ is given by (11). Once the optimal $\mathbf{\Sigma}_{k,\star}$ has been obtained, we can express the ultimate weight vector $\mathbf{w}_{k,\star}$ as (15) shown at the bottom of this page. To find the solution of problem (14), an iterative method is first provided below.

$$\mathbf{w}_{k,\star} = \mathbf{w}_{k-1,\star} - \mathbf{T}_{k-1}^{-1}\mathbf{A}_k\left(\mathbf{I} + \mathbf{\Sigma}_{k,\star}\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}\mathbf{A}_k\right)^{-1}\mathbf{\Sigma}_{k,\star}\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}\mathbf{a}(\theta_0) \tag{15}$$

Algorithm 1: Iterative Approach to Problem (14).

give $\mathbf{a}(\theta_0)$, $\theta_{k,m}$, $\rho_{k,m}$, $m=1,\ldots,M_k$, and $\mathbf{A}_k$, set $\beta_\epsilon > 0$, $\beta_{\rm MAX} > \beta_\epsilon$, $\mathbf{\Xi} = \mathbf{T}_{k-1}$, $\mathbf{\Sigma}_k = \mathbf{0}$
while $\beta_{\rm MAX} > \beta_\epsilon$ **do**
  for $m = 1,\ldots,M_k$ **do**
    calculate $\beta_{k,m,\star}$ from Eqn. (42) of [1], by setting $L(\theta_{k,m},\theta_0) = \rho_{k,m}$
    update VCM $\mathbf{\Xi} = \mathbf{\Xi} + \beta_{k,m,\star}\mathbf{a}(\theta_{k,m})\mathbf{a}^{\rm H}(\theta_{k,m})$
  end for
  update $\mathbf{\Sigma}_k$ as $\mathbf{\Sigma}_k = \mathbf{\Sigma}_k + {\rm Diag}([\beta_{k,1,\star},\ldots,\beta_{k,M_k,\star}])$
  obtain $\beta_{\rm MAX} = \max\limits_{1\le m\le M_k} |\beta_{k,m,\star}|$
end while
obtain $\mathbf{\Sigma}_{k,\star} = \mathbf{\Sigma}_k$

C. Iterative Approach

The OPARC algorithm, developed in the companion paper [1], is able to optimally and precisely adjust one-point response level at a time. Thus, we may apply it to the M_k-point OPARC problem (14) as follows. For a fixed $k>0$, we apply the OPARC algorithm for M_k steps. In the mth step, OPARC is to realize $L(\theta_{k,m},\theta_0)=\rho_{k,m}$, $m=1,\ldots,M_k$. Unfortunately, OPARC brings inevitable pattern variations on the previous controlled angles as we have discussed in [1]. More specifically, the response levels of $\theta_{k,i}$, $i=1,\ldots,m-1$, vary after accurately controlling the response level of $\theta_{k,m}$ to its desired level $\rho_{k,m}$, $2\le m\le M_k$. To reduce the undesirable pattern variations on the pre-adjusted angles, we propose to iteratively apply the M_k-point OPARC for a number of times, until a certain termination criterion is met. A temporary variable $\mathbf{\Xi}=\mathbf{T}_{k-1}$ and $\mathbf{\Sigma}_k=\mathbf{0}$ are taken as the initializations in the first iteration. Then, in each iteration, an M_k-step OPARC is carried out. More specifically, in the mth step, we adjust the response level of $\theta_{k,m}$ to be $\rho_{k,m}$, by calculating the INR of the newly assigned virtual interference at $\theta_{k,m}$, denoted as $\beta_{k,m,\star}$, $m=1,\ldots,M_k$, from Eqn. (42) of [1], and then update the associated VCM as $\mathbf{\Xi}=\mathbf{\Xi}+\beta_{k,m,\star}\mathbf{a}(\theta_{k,m})\mathbf{a}^{\rm H}(\theta_{k,m})$. Once an iteration, i.e., an M_k-step OPARC, is completed, $\beta_{k,m,\star}$ is added to the mth diagonal element of $\mathbf{\Sigma}_k$, and then we set the resulting $\mathbf{\Xi}$ as the initial VCM in the next iteration. Note that $\mathbf{T}_0=\mathbf{I}$.

Naturally, whether the response levels of the adjusted angles $\theta_{k,m}$, $m=1,\ldots,M_k$, are close enough to their desired levels can be a criterion to terminate the iteration of OPARC. However, this strategy needs to calculate all the intermediate weight vectors that may be computationally inefficient. To improve the computational efficiency, we propose to terminate the iteration of OPARC by examining whether the magnitudes of INRs of the newly assigned virtual interferences approximate enough to zero, since there is no need to assign virtual interferences if their values are small enough.

Finally, we summarize the above iterative solver of problem (14) in Algorithm 1, where β_ϵ stands for a small tolerance parameter. Note that $\beta_{k,m,\star}$ in Algorithm 1 is calculated with Eqn. (42) of [1]. In addition, we can express the ultimate $\mathbf{\Sigma}_{k,\star}$ as

$$\mathbf{\Sigma}_{k,\star} = {\rm Diag}([\bar\beta_{k,1,\star},\ldots,\bar\beta_{k,M_k,\star}]) \tag{16}$$

where $\bar\beta_{k,m,\star}$ represents the total INR of the virtual interference assigned at $\theta_{k,m}$ in the kth step, and equals to the summation of all $\beta_{k,m,\star}$'s of different iterations for a fixed $m=1,\ldots,M_k$. As discussed earlier, once the optimal $\mathbf{\Sigma}_{k,\star}$ has been obtained, we can use $\mathbf{\Sigma}_{k,\star}$ to obtain the VCM $\mathbf{T}_k$ by Eqn. (6), update $\mathbf{h}_k$ in (11) and (14c), and calculate $\mathbf{w}_{k,\star}$ by Eqn. (15). It shall be noted that an inverse of normalized covariance matrix is indispensable in determining $\beta_{k,m,\star}$'s by Eqn. (42) of [1]. This may lead to a high cost in memory or/and computation especially for a large array, although it may not need a large number of iterations.

D. C-ADMM Approach

We next propose another approach to solve problem (14). We first reformulate the original problem (14) as a quadratically constrained quadratic program (QCQP) problem. Then, the recently developed consensus-ADMM (C-ADMM) [4] approach is employed to find its solution.

1) Problem Reformulation: Since $\mathbf{h}_k$ is a one-one mapping of $\mathbf{\Sigma}_k$, we can formulate the multi-point OPARC, i.e., problem (14), by finding $\mathbf{h}_k$ as

$$\max_{\mathbf{h}_k\in\mathbb{C}^{M_k}} \quad G_k = |\mathbf{w}_k^{\rm H}\mathbf{a}(\theta_0)|^2/|\mathbf{w}_k^{\rm H}\mathbf{T}_k\mathbf{w}_k| \tag{17a}$$

$$\text{subject to} \quad L(\theta_{k,m},\theta_0)=\rho_{k,m},\ m=1,\ldots,M_k \tag{17b}$$

$$\mathbf{w}_k = \mathbf{w}_{k-1,\star} + \mathbf{T}_{k-1}^{-1}\mathbf{A}_k\mathbf{h}_k. \tag{17c}$$

We substitute the constraint (17c) into G_k and obtain

$$\begin{aligned} G_k^2 &= \left|\mathbf{a}^{\rm H}(\theta_0)(\mathbf{w}_{k-1,\star}+\mathbf{T}_{k-1}^{-1}\mathbf{A}_k\mathbf{h}_k)\right|^2 \\ &= -\mathbf{h}_k^{\rm H}\widetilde{\mathbf{C}}\mathbf{h}_k + 2\Re\left(\tilde{\mathbf{c}}^{\rm H}\mathbf{h}_k\right) + |\mathbf{a}^{\rm H}(\theta_0)\mathbf{w}_{k-1,\star}|^2 \end{aligned} \tag{18}$$

where $\mathbf{w}_k=\mathbf{T}_k^{-1}\mathbf{a}(\theta_0)$ is used, $\widetilde{\mathbf{C}}$ and $\tilde{\mathbf{c}}$ are defined as

$$\widetilde{\mathbf{C}} \triangleq -(\mathbf{T}_{k-1}^{-1}\mathbf{A}_k)^{\rm H}\mathbf{a}(\theta_0)\mathbf{a}^{\rm H}(\theta_0)\mathbf{T}_{k-1}^{-1}\mathbf{A}_k \in \mathbb{C}^{M_k\times M_k} \tag{19a}$$

$$\tilde{\mathbf{c}} \triangleq (\mathbf{T}_{k-1}^{-1}\mathbf{A}_k)^{\rm H}\mathbf{a}(\theta_0)\mathbf{a}^{\rm H}(\theta_0)\mathbf{w}_{k-1,\star} \in \mathbb{C}^{M_k}. \tag{19b}$$

Since $|\mathbf{a}^{\rm H}(\theta_0)\mathbf{w}_{k-1,\star}|^2$ is a constant, the maximization of G_k is thus equivalent to the minimization of $\mathbf{h}_k^{\rm H}\widetilde{\mathbf{C}}\mathbf{h}_k - 2\Re\left(\tilde{\mathbf{c}}^{\rm H}\mathbf{h}_k\right)$.

On the other hand, recalling the expression of $L(\theta,\theta_0)$, we can rewrite the constraint (17b) as

$$\mathbf{w}_k^{\rm H}\mathbf{S}_{k,m}\mathbf{w}_k = 0,\ m=1,\ldots,M_k \tag{20}$$

where $\mathbf{S}_{k,m}=\mathbf{a}(\theta_{k,m})\mathbf{a}^{\rm H}(\theta_{k,m})-\rho_{k,m}\mathbf{a}(\theta_0)\mathbf{a}^{\rm H}(\theta_0)$. Substituting the constraint (17c) into (20), we have

$$\mathbf{h}_k^{\rm H}\widetilde{\mathbf{D}}_m\mathbf{h}_k - 2\Re(\tilde{\mathbf{d}}_m^{\rm H}\mathbf{h}_k) = \alpha_m,\ m=1,\ldots,M_k \tag{21}$$

where

$$\widetilde{\mathbf{D}}_m = (\mathbf{T}_{k-1}^{-1}\mathbf{A}_k)^{\rm H}\mathbf{S}_{k,m}\mathbf{T}_{k-1}^{-1}\mathbf{A}_k \in \mathbb{C}^{M_k\times M_k} \tag{22a}$$

$$\tilde{\mathbf{d}}_m = -(\mathbf{T}_{k-1}^{-1}\mathbf{A}_k)^{\rm H}\mathbf{S}_{k,m}\mathbf{w}_{k-1,\star} \in \mathbb{C}^{M_k} \tag{22b}$$

$$\alpha_m = -\mathbf{w}_{k-1,\star}^{\rm H}\mathbf{S}_{k,m}\mathbf{w}_{k-1,\star} \in \mathbb{R}. \tag{22c}$$

Thus, problem (17) can be reformulated as

$$\min_{\mathbf{h}_k} \ \mathbf{h}_k^{\mathrm{H}}\widetilde{\mathbf{C}}\mathbf{h}_k - 2\Re\left(\widetilde{\mathbf{c}}^{\mathrm{H}}\mathbf{h}_k\right) \tag{23a}$$

$$\text{subject to } \ \mathbf{h}_k^{\mathrm{H}}\widetilde{\mathbf{D}}_m\mathbf{h}_k - 2\Re(\widetilde{\mathbf{d}}_m^{\mathrm{H}}\mathbf{h}_k) = \alpha_m \quad m = 1,\ldots,M_k. \tag{23b}$$

In the sequel, we adopt the newly developed C-ADMM approach [4] to solve problem (23).

2) C-ADMM Solver: We first convert (23) into its real domain as

$$\min_{\mathbf{z}} \ \mathbf{z}^{\mathrm{T}}\mathbf{C}\mathbf{z} - 2\mathbf{c}^{\mathrm{T}}\mathbf{z} \tag{24a}$$

$$\text{subject to } \ \mathbf{z}^{\mathrm{T}}\mathbf{D}_m\mathbf{z} - 2\mathbf{d}_m^{\mathrm{T}}\mathbf{z} = \alpha_m \quad m = 1,\ldots,M_k \tag{24b}$$

where

$$\mathbf{z} = \left[\,\Re(\mathbf{h}_k^{\mathrm{T}})\ \Im(\mathbf{h}_k^{\mathrm{T}})\,\right]^{\mathrm{T}} \in \mathbb{R}^{2M_k} \tag{25a}$$

$$\mathbf{c} = \left[\,\Re(\widetilde{\mathbf{c}}^{\mathrm{T}})\ \Im(\widetilde{\mathbf{c}}^{\mathrm{T}})\,\right]^{\mathrm{T}} \in \mathbb{R}^{2M_k} \tag{25b}$$

$$\mathbf{d}_m = \left[\,\Re(\widetilde{\mathbf{d}}_m^{\mathrm{T}})\ \Im(\widetilde{\mathbf{d}}_m^{\mathrm{T}})\,\right]^{\mathrm{T}} \in \mathbb{R}^{2M_k} \tag{25c}$$

$$\mathbf{C} = \begin{bmatrix} \Re(\widetilde{\mathbf{C}}) & -\Im(\widetilde{\mathbf{C}}) \\ \Im(\widetilde{\mathbf{C}}) & \Re(\widetilde{\mathbf{C}}) \end{bmatrix} \in \mathbb{R}^{2M_k\times 2M_k} \tag{25d}$$

$$\mathbf{D}_m = \begin{bmatrix} \Re(\widetilde{\mathbf{D}}_m) & -\Im(\widetilde{\mathbf{D}}_m) \\ \Im(\widetilde{\mathbf{D}}_m) & \Re(\widetilde{\mathbf{D}}_m) \end{bmatrix} \in \mathbb{R}^{2M_k\times 2M_k}. \tag{25e}$$

To tackle (24), we introduce the auxiliary variable vectors $\mathbf{p}_m$, $m = 1,\ldots,M_k$, and then formulate (24) as

$$\min_{\mathbf{z},\{\mathbf{p}_m\}_{m=1}^{M_k}} \ \mathbf{z}^{\mathrm{T}}\mathbf{C}\mathbf{z} - 2\mathbf{c}^{\mathrm{T}}\mathbf{z} \tag{26a}$$

$$\text{subject to } \ \mathbf{p}_m = \mathbf{z} \tag{26b}$$

$$\mathbf{p}_m^{\mathrm{T}}\mathbf{D}_m\mathbf{p}_m - 2\mathbf{d}_m^{\mathrm{T}}\mathbf{p}_m = \alpha_m \quad m = 1,\ldots,M_k. \tag{26c}$$

Note that the non-convex constraint in problem (26) is only imposed on $\mathbf{p}_m$ and not related to $\mathbf{z}$. Moreover, for any given $m = 1,\ldots,M_k$, the nonconvex-constraint, i.e., (26c), is a QCQP with only one constraint (QCQP-1), which can be easily solved as pointed out in [4]. Thus, the newly formulated problem (26) simplifies the original problem (24) to solve.

To see the details, we first devise the augmented Lagrangian by ignoring the constraint (26c):

$$\mathcal{L}_\eta(\mathbf{z},\mathbf{p},\boldsymbol{\lambda}) = \mathbf{z}^{\mathrm{T}}\mathbf{C}\mathbf{z} - 2\mathbf{c}^{\mathrm{T}}\mathbf{z} + \sum_{i=1}^{M_k}\boldsymbol{\lambda}_m^{\mathrm{T}}(\mathbf{z}-\mathbf{p}_m) + \sum_{i=1}^{M_k}\frac{\eta}{2}\|\mathbf{z}-\mathbf{p}_m\|_2^2 \tag{27}$$

where $\eta > 0$ is the penalty parameter, $\boldsymbol{\lambda}_m \in \mathbb{R}^{2M_k}$ are Lagrange multiplier vectors. Note that the augmented Lagrangian (27) acts as the (unaugmented) Lagrangian associated with the following problem:

$$\min_{\mathbf{z},\{\mathbf{p}_m\}_{m=1}^{M_k}} \ \mathbf{z}^{\mathrm{T}}\mathbf{C}\mathbf{z} - 2\mathbf{c}^{\mathrm{T}}\mathbf{z} + \sum_{i=1}^{M_k}\frac{\eta}{2}\|\mathbf{z}-\mathbf{p}_m\|_2^2 \tag{28a}$$

$$\text{subject to } \ \mathbf{p}_m = \mathbf{z},\ m = 1,\ldots,M_k \tag{28b}$$

which is equivalent to problem (26) in the absence of the constraint (26c), since for any feasible $\mathbf{z}$ and $\mathbf{p}_m$, $m = 1,\ldots,M_k$, the added term, i.e., the last term in (28a), to the objective function is zero. As mentioned in [30], the augmented Lagrangian brings robustness to the dual ascent method adopted later.

Since the constraints (26c) are imposed on $\mathbf{p}_m$ and not related to $\mathbf{z}$, they only play roles in finding $\mathbf{p}_m$, $m = 1,\ldots,M_k$. For this reason, we don't include (26c) in the above augmented Lagrangian intentionally. Instead, we take the constraints in (26c) into consideration when minimizing $\mathcal{L}_\eta(\mathbf{z},\mathbf{p},\boldsymbol{\lambda})$ as shown next.

The alternating direction method of multipliers (ADMM) [30], which is an operator splitting algorithm originally devised to solve convex optimization problems, has been explored as a heuristic method to solve non-convex problems [4]. Following the decomposition-coordination procedure of ADMM in [30], we can determine $\{\mathbf{z},\mathbf{p}_m,\boldsymbol{\lambda}\}$ via the alternative and iterative steps below.

Step 1: Update $\mathbf{z}$

$$\begin{aligned}\mathbf{z}(t+1) &= \arg\min_{\mathbf{z}} \mathcal{L}_\eta(\mathbf{z},\mathbf{p}(t),\boldsymbol{\lambda}(t)) \\ &= \arg\min_{\mathbf{z}} \mathbf{z}^{\mathrm{T}}\left(\mathbf{C}+\frac{\eta M_k}{2}\mathbf{I}\right)\mathbf{z} - 2\mathbf{g}^{\mathrm{T}}(t+1)\mathbf{z} \\ &= \left(\mathbf{C}+\frac{\eta M_k}{2}\mathbf{I}\right)^{-1}\mathbf{g}(t+1)\end{aligned} \tag{29}$$

where $\mathbf{g}(t+1) = \mathbf{c} - (1/2)\sum_{m=1}^{M_k}(\boldsymbol{\lambda}_m(t) - \eta\mathbf{p}_m(t))$.

Step 2: Update $\mathbf{p}$

For $m = 1,\ldots,M_k$, we update the vector $\mathbf{p}_m$ as

$$\begin{aligned}\mathbf{p}_m(t+1) &= \arg\min_{\mathbf{p}_m} \mathcal{L}_\eta(\mathbf{z}(t+1),\mathbf{p},\boldsymbol{\lambda}(t)) \\ &= \arg\min_{\mathbf{p}_m} \eta\mathbf{p}_m^{\mathrm{T}}\mathbf{p}_m - 2(\eta\mathbf{z}(t+1)+\boldsymbol{\lambda}_m(t))^{\mathrm{T}}\mathbf{p}_m \\ &= \arg\min_{\mathbf{p}_m} \|\mathbf{p}_m - \boldsymbol{\zeta}_m(t+1)\|_2^2 \qquad (30a)\end{aligned}$$

$$\text{subject to } \mathbf{p}_m^{\mathrm{T}}\mathbf{D}_m\mathbf{p}_m - 2\mathbf{d}_m^{\mathrm{T}}\mathbf{p}_m = \alpha_m \tag{30b}$$

where $\boldsymbol{\zeta}_m(t+1) = \mathbf{z}(t+1) + (1/\eta)\boldsymbol{\lambda}_m(t)$. Since the above problem is QCQP-1 which is equivalent to solving a polynomial as mentioned in [4], the bisection or Newtons method can be adopted to find its (approximate) solution, see [4] and [31] for reference.

Step 3: Update $\boldsymbol{\lambda}$

For $m = 1,\ldots,M_k$, we update the vector $\boldsymbol{\lambda}_m$ as

$$\boldsymbol{\lambda}_m(t+1) = \boldsymbol{\lambda}_m(t) + \eta(\mathbf{z}(t+1) - \mathbf{p}_m(t+1)). \tag{31}$$

The above steps 1 to 3 are repeated until a stopping criterion is reached, e.g., a maximum iteration number is attained and/or

$$\delta > \delta_{\mathrm{MAX}} \triangleq \max_{1\le m\le M_k} \|\mathbf{z}(t+1) - \mathbf{p}_m(t+1)\|_2 \tag{32}$$

where $\delta > 0$ is a small tolerance parameter.

Algorithm 2: C-ADMM Approach to Problem (14).

1: give $\mathbf{a}(\theta_0)$, $\mathbf{T}_{k-1}$, $\mathbf{w}_{k-1,\star} = \mathbf{T}_{k-1}^{-1}\mathbf{a}(\theta_0)$, $\theta_{k,m}$, $\rho_{k,m}$, $m = 1, \ldots, M_k$, and $\mathbf{A}_k$, obtain $\mathbf{c}$, $\mathbf{C}$, $\mathbf{d}_m$, $\mathbf{D}_m$ from (25), initialize $\mathbf{p}_m$, $m = 1, \ldots, M_k$, by (34), set $\delta_{\mathrm{MAX}} > \delta > 0$ and $\eta > 0$
2: **while** $\delta_{\mathrm{MAX}} > \delta$ **do**
3: update $\mathbf{z}$ by (29)
4: update $\mathbf{p}_m$, $m = 1, \ldots, M_k$, by (30)
5: update $\boldsymbol{\lambda}_m$, $m = 1, \ldots, M_k$, by (31)
6: calculate δ_{MAX} by (32)
7: **end while**
8: obtain $\mathbf{z}_\star = \mathbf{z}$
9: obtain $\mathbf{h}_{k,\star}$ by (25a)

3) Initialization of C-ADMM: Note that due to the non-convexity of problem (26), typical convergence results on ADMM do not apply and the ultimate $\mathbf{z}$ is not guaranteed to be optimal. Nevertheless, an appropriate initialization makes the above iterative algorithm [4] work well and even converge to a stationary point. Following [4], we initialize $\mathbf{p}_m$ as

$$\mathbf{p}_m = \left[\Re(\widetilde{\mathbf{p}}_m^{\mathrm{T}})\ \Im(\widetilde{\mathbf{p}}_m^{\mathrm{T}})\right]^{\mathrm{T}},\ m = 1, \ldots, M_k \tag{33}$$

where

$$\widetilde{\mathbf{p}}_m = [\underbrace{0, \ldots, 0}_{m-1}, \gamma_{m,\star}, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{C}^{M_k}. \tag{34}$$

In (34), $\gamma_{m,\star}$ is obtained by the OPARC algorithm and satisfies

$$\frac{\left|(\mathbf{w}_{k-1,\star} + \gamma_{m,\star}\mathbf{T}_{k-1}^{-1}\mathbf{a}(\theta_{k,m}))^{\mathrm{H}}\mathbf{a}(\theta_{k,m})\right|^2}{\left|(\mathbf{w}_{k-1,\star} + \gamma_{m,\star}\mathbf{T}_{k-1}^{-1}\mathbf{a}(\theta_{k,m}))^{\mathrm{H}}\mathbf{a}(\theta_0)\right|^2} = \rho_{k,m}. \tag{35}$$

It can be verified that, the constraints (26c) can be satisfied if the initial settings $\mathbf{p}_m$, $m = 1, \ldots, M_k$, take (33). This makes it easier to find an approximate solution of problem (26).

Once the solution $\mathbf{z}_\star$ has been obtained, we can reconstruct $\mathbf{h}_{k,\star}$ by (25a) and obtain $\mathbf{w}_{k,\star}$ as

$$\mathbf{w}_{k,\star} = \mathbf{w}_{k-1,\star} + \mathbf{T}_{k-1}^{-1}\mathbf{A}_k\mathbf{h}_{k,\star}. \tag{36}$$

The INRs of the newly assigned virtual interferences can by calculated via

$$\boldsymbol{\Sigma}_{k,\star} = \mathrm{Diag}\left(-\mathbf{h}_{k,\star} \oslash \left(\mathbf{A}_k^{\mathrm{H}}\mathbf{T}_{k-1}^{-1}\left(\mathbf{a}(\theta_0) + \mathbf{A}_k\mathbf{h}_{k,\star}\right)\right)\right). \tag{37}$$

To make the above procedure clear, we summarize the C-ADMM approach to solve problem (14) in Algorithm 2. Notice from [4] that the C-ADMM approach is memory-efficient and can be implemented in a parallelized or distributed manner. Thus, for a large array, the C-ADMM approach in Algorithm 2 may be a better choice to solve problem (14) compared to the iterative approach in Algorithm 1, although more iterations may be needed.

E. Update of Covariance Matrix

Similar to the OPARC algorithm, the VCM $\mathbf{T}_k$ also needs to be renewed so as to facilitate the next execution of multi-point

Algorithm 3: Multi-point OPARC Algorithm.

1: give $\mathbf{a}(\theta_0)$, $\mathbf{T}_{k-1}$ and the weight vector $\mathbf{w}_{k-1,\star} = \mathbf{T}_{k-1}^{-1}\mathbf{a}(\theta_0)$, prescribe the angle $\theta_{k,m}$ and the corresponding desired level $\rho_{k,m}$, $m = 1, \ldots, M_k$
2: calculate $\boldsymbol{\Sigma}_{k,\star}$ or $\mathbf{h}_{k,\star}$ using Algorithm 1 or Algorithm 2
3: obtain $\mathbf{T}_k$ by (38) and calculate $\mathbf{w}_{k,\star}$ by (15) or (36)

OPARC. From the above discussions, $\mathbf{T}_k$ is updated as

$$\mathbf{T}_k = \mathbf{T}_{k-1} + \mathbf{A}_k\boldsymbol{\Sigma}_{k,\star}\mathbf{A}_k^{\mathrm{H}}. \tag{38}$$

Accordingly, the weight vector is

$$\mathbf{w}_{k,\star} = \mathbf{T}_k^{-1}\mathbf{a}(\theta_0). \tag{39}$$

This completes the procedure of multi-point OPARC. Finally, we describe the steps of multi-point OPARC in Algorithm 3.

Note that in our proposed multi-point OPARC algorithm, we carry out the parameter determination in a subspace with dimension M_k, not in the whole space of dimension N. The benefit is the reduced amount of calculation. In addition, one can see that at most $M_{\max} = N - 1$ points can be precisely controlled, due to the limited degrees of freedom in problem (14) or (17).

As a remark, the differences between the recent MA^{2}RC in [2] and the proposed multi-point OPARC in this paper are similar to those between A^{2}RC and OPARC described in [1] in details.

F. Computational Complexities

We now analyze the computational complexities of the proposed algorithms and compare them with the existing MA^{2}RC algorithm in [2]. Note that two algorithms have been presented to solve the problem (14). In the iterative approach (see Algorithm 1), an M_k-step OPARC is carried out in each iteration, and the main computational cost lies in the calculation of $\mathbf{T}_{k-1}^{-1}$, which takes a computational complexity of $O(N^3)$. For the C-ADMM approach (see Algorithm 2), the main computational costs are the updates of $\mathbf{z}$ and $\mathbf{p}_m$'s, $m = 1, \ldots, M_k$, in (29) and (30), respectively. The update of $\mathbf{z}$ requires a computational complexity of $O(M_k^3)$. To update $\mathbf{p}_m$'s in (30), the eigen-decompositions of $\mathbf{D}_m$'s, $m = 1, \ldots, M_k$, are necessary [4] and the total computational complexity is $O(M_k^4)$. Therefore, the computational complexities of the iterative approach and the C-ADMM approach are $O(N^3)$ and $O(M_k^4)$, respectively, in each iteration step. In addition, the C-ADMM approach can be implemented in parallel/distributed manner [4], which can further increase the computational efficiency. For the MA^{2}RC algorithm in [2], its computational complexity is $O(M_k N^3)$, which is greater than those of the proposed algorithms.

G. Convergence Properties

In this subsection, convergence properties of the proposed algorithms are provided. We have presented two approaches to solve the problem (14). For the first approach (see Algorithm 1), we substitute the constraint (14c) into (14a) and reformulate the

problem (14) as

$$\min_{\mathbf{\Sigma}_k} \ \mathbf{h}_k^{\mathrm{H}}\widetilde{\mathbf{C}}\mathbf{h}_k - 2\Re\left(\widetilde{\mathbf{c}}^{\mathrm{H}}\mathbf{h}_k\right) \tag{40a}$$

$$\text{subject to } \ L(\theta_{k,m},\theta_0)=\rho_{k,m},\ m=1,\ldots,M_k \tag{40b}$$

where $\mathbf{h}_k$ is a mapping of $\mathbf{\Sigma}_k$ as shown in (11), $\widetilde{\mathbf{C}}$ and $\widetilde{\mathbf{c}}$ have been given in (19). The above problem (40) can be further transformed as the following un-constrained one:

$$\min_{\beta_{k,1},\ldots,\beta_{k,M_k}} f(\beta_{k,1},\ldots,\beta_{k,M_k}) \tag{41}$$

where

$$f(\beta_{k,1},\ldots,\beta_{k,M_k}) \triangleq \mathbf{h}_k^{\mathrm{H}}\widetilde{\mathbf{C}}\mathbf{h}_k - 2\Re\left(\widetilde{\mathbf{c}}^{\mathrm{H}}\mathbf{h}_k\right) + \sum_{m=1}^{M_k}\varsigma\left(L(\theta_{k,m},\theta_0)-\rho_{k,m}\right) \tag{42}$$

with $\varsigma(\cdot)$ denoting the penalty function satisfying

$$\varsigma(x)=\begin{cases}0, & \text{if } x=0\\ \nu, & \text{if } x\neq 0\end{cases} \tag{43}$$

where ν is a large enough constant. Recalling the first approach, denote the resulting $\beta_{k,m,\star}$ (in Line 4 of Algorithm 1) of the rth step as $\beta_{k,m,\star}^{(r)}$, one knows that $\beta_{k,m,\star}^{(r)}$ can make $L(\theta_{k,m},\theta_0)=\rho_{k,m}$ satisfied. Thus, it's not hard to learn that $\beta_{k,m,\star}^{(r)}$ is actually the optimal value as

$$\beta_{k,m,\star}^{(r)} = \arg\min_{\beta} f(\beta_{k,1}^{(r)},\ldots,\beta_{k,m-1}^{(r)},\beta,\beta_{k,m+1}^{(r-1)},\ldots,\beta_{k,M_k}^{(r-1)}).$$

This indicates that the first approach is essentially a block coordinate descent (BCD) method [32], in which one block variable (or few blocks of variables) is optimized at each iteration while holding the remaining variables fixed. Although the function f in (42) is non-convex and non-differentiable, we can know from [33] that the sequence $\{(\beta_{k,1,\star}^{(r)},\ldots,\beta_{k,M_k,\star}^{(r)})\}_{r=0,1,\ldots}$ (generated by the BCD method) converges to a stationary point of f, provided that the cyclic rule is used. In Algorithm 1, we cycle through all coordinates or variables sequentially. This is the cheapest coordinate selection strategy and it satisfies the cyclic rule presented in [33]. Therefore, we know that the first approach, i.e., Algorithm 1, will lead to a stationary point.

For the second approach (see Algorithm 2), it falls into the C-ADMM framework, which provides a universal method to solve a general QCQP. The convergence properties of C-ADMM approach have been presented in [4]. Following the result obtained in [4], we know that if

$$\lim_{t\to+\infty}\left(\mathbf{p}_m(t)-\mathbf{z}(t)\right)=\mathbf{0},\ \forall\, m=1,\ldots,M_k \tag{44}$$

and

$$\lim_{t\to+\infty}\left(\mathbf{z}(t+1)-\mathbf{z}(t)\right)=\mathbf{0} \tag{45}$$

then any limit point of $\{\mathbf{z}(t)\}$ is a stationary point of problem (23), and the corresponding $\mathbf{\Sigma}_k$ will be a stationary point of the original problem (14). Moreover, as mentioned in [4], an appropriate initialization is important for the convergence of C-ADMM. It has been found empirically in [4] that a feasible initialization point makes C-ADMM work well in most cases, see the selection of (33) for our problem.

In addition, it should be emphasized that the problem (14) is a QCQP, which is in general NP-hard [4] under normal circumstances. Therefore, the proposed two approaches in Algorithm 1 and Algorithm 2 may provide a sub-optimal solution, which may not be globally optimal. Nevertheless, the proposed algorithms work well and have satisfactory performance improvement in the applications as presented next.

III. Applications of Multi-Point OPARC

In this section, we present three applications of multi-point OPARC to array signal processing.

A. Array Pattern Synthesis

Given the beam axis θ_0, the problem of array pattern synthesis is to find an appropriate $N\times 1$ weight vector that makes the response $L(\theta,\theta_0)$ meet some specific requirements. For simplicity, we denote the desired pattern as $L_d(\theta)$. Basically, the proposed algorithm herein shares a similar concept of pattern synthesis using A^{2}RC in [3]. However, it is able to significantly reduce the number of iterations and improve the performance.

1) General Case: Generally, the array pattern synthesis can be started by setting $k=0$ and the initial weight as $\mathbf{w}_{0,\star}=\mathbf{a}(\theta_0)$. For $k>0$, multiple directions are selected by comparing $L_{k-1}(\theta,\theta_0)$:

$$L_{k-1}(\theta,\theta_0)\triangleq|\mathbf{w}_{k-1}^{\mathrm{H}}\mathbf{a}(\theta)|^2\big/|\mathbf{w}_{k-1}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2 \tag{46}$$

with the desired pattern $L_d(\theta)$ as follows. These angles can be in either the sidelobe region or the mainlobe region. For sidelobe synthesis, we only choose the peak angles in the set

$$\Omega_{k,S}=\big\{\theta\big|L_{k-1}(\theta,\theta_0)>L_{k-1}(\theta-\varepsilon,\theta_0)\text{ and } L_{k-1}(\theta,\theta_0)>L_{k-1}(\theta+\varepsilon,\theta_0),\ \theta\in\Omega_S\big\} \tag{47}$$

where ε is a small positive quantity, Ω_S denotes the sidelobe sector of the desired pattern. Different from the angle selection method in A^{2}RC where the chosen peak angles have larger response levels than their desired values, a selected peak angle in set $\Omega_{k,S}$ may have a less response level than its desired one. For mainlobe synthesis, some discrete angles where the responses deviate considerably from the desired ones are chosen, and we denote the set of selected angles in the mainlobe region as $\Omega_{k,M}$. Then, we take:

$$\Omega_k=\Omega_{k,S}\cup\Omega_{k,M}\triangleq\{\theta_{k,1},\ldots,\theta_{k,M_k}\} \tag{48}$$

where $M_k=\mathrm{card}(\Omega_k)$. The multi-point OPARC algorithm can thus be applied to adjust the corresponding responses of angles $\theta_{k,m}$ to their desired values $\rho_{k,m}=L_d(\theta_{k,m}),\ m=1,\ldots,M_k$, and the current response pattern $L_k(\theta,\theta_0)$ can be obtained by using the resulting weight of multi-point OPARC. Then, set $k=k+1$ and repeat the above process until the response is satisfactorily synthesized. Note that the above iteration procedure is different from that in Section II-C where k is fixed and an internal iteration within the kth step is conducted. To summarize, we describe the multi-point OPARC based array pattern

synthesis algorithm in Algorithm 4. As mentioned earlier, Ω_k is forced to satisfy $\mathrm{card}(\Omega_k) < N$. Otherwise, we can simply reduce $\mathrm{card}(\Omega_k)$ by modifying Ω_k similar to what is done next.

2) Particular Consideration for Large Arrays: As aforementioned, the proposed multi-point OPARC algorithm operates in an M_k-dimensional subspace of the original N-dimensional space. This provides us an effective strategy to pattern synthesis for large arrays, where the traditional approaches may not work well or require extensive computation due to the large dimension.

More specifically, for a large array and a pre-determined angle set Ω_k (whose cardinality normally approaches to N) in (48), we construct a new angle set Θ_k as

$$\Theta_k = \left\{\bar{\theta}_{k,1}, \bar{\theta}_{k,2}, \ldots, \bar{\theta}_{k,C_k}\right\} \tag{49}$$

where C_k is a prescribed number that is much smaller than N, $\bar{\theta}_{k,c}$, $c = 1, \ldots, C_k$, is the cth element of the vector:

$$\mathrm{Sort}(\Omega_k) \in \mathbb{R}^{\mathrm{card}(\Omega_k)} \tag{50}$$

where $\mathrm{Sort}(\Omega_k)$ re-arranges the elements of Ω_k in the following way: the larger $|L_{k-1}(\bar{\theta}, \theta_0) - L_d(\bar{\theta})|$ for $\bar{\theta} \in \Omega_k$ is, the smaller index of $\bar{\theta}$ in $\mathrm{Sort}(\Omega_k)$ is, which makes $\bar{\theta}$ more likely to be chosen as an element in the angle set Θ_k in (49). The reason for this is that we expect to reduce the overall difference between the resulting pattern and the desired one.

Once the new angle set Θ_k is obtained, the multi-point OPARC algorithm can be applied to realize $L_k(\bar{\theta}, \theta_0) = L_d(\bar{\theta})$ for $\bar{\theta} \in \Theta_k$. Then, set $k = k + 1$ and repeat the above process until the response is satisfactorily synthesized, and the cardinality of set Θ_k, i.e., C_k, can be flexibly varied with the iteration number k. Finally, the above-described large-array pattern synthesis can be readily realized via Algorithm 4, by simply replacing Ω_k in the 4th line of Algorithm 4 with the new angle set Θ_k in (49).

Since the above proposed algorithm, in either the general case or the large-array scenario, iteratively adjusts the responses of sidelobe peaks, it is able to make all the sidelobe peaks align with the desired values. Thus, all the sidelobe responses can be well controlled to be lower than the given thresholds, and a satisfactory sidelobe shape can be well maintained. Nevertheless, array pattern synthesis works in a data-independent way, the resulting weight or its corresponding beampattern is lack of adaptivity in suppressing undesirable interference and noise, which can be well rejected by the adaptive beamformer as discussed next.

Algorithm 4: Multi-point OPARC based Array Pattern Synthesis Algorithm.

1: give $L_d(\theta)$, $\mathbf{w}_{0,\star} = \mathbf{a}(\theta_0)$, set $k = 1$, $\mathbf{T}_0 = \mathbf{I}$, $\varepsilon > 0$
2: **while** 1 **do**
3: determine Ω_k from (48)
4: apply multi-point OPARC algorithm to realize $L_k(\theta, \theta_0) = L_d(\theta)$ ($\theta \in \Omega_k$), update $\mathbf{w}_{k,\star}$ and $\mathbf{T}_k$
5: **if** $L_k(\theta, \theta_0)$ meets the requirement **then**
6: break
7: **end if**
8: set $k = k + 1$
9: **end while**
10: output $\mathbf{w}_{k,\star}$ and $L_k(\theta, \theta_0)$

B. Multi-Constraint Adaptive Beamforming

The linearly constrained minimum variance (LCMV) beamformer is commonly used to enhance the robustness of array systems [20]–[22]. In LCMV beamformer, several linear constraints are imposed when minimizing the output variance, i.e.,

$$\min_{\mathbf{w}} \ \mathbf{w}^{\mathrm{H}}\mathbf{R}_{n+i}\mathbf{w} \tag{51a}$$

$$\text{subject to } \mathbf{C}^{\mathrm{H}}\mathbf{w} = \mathbf{g} \tag{51b}$$

where $\mathbf{C}$ is the constraint matrix that consists of D spatial steering vectors corresponding to the D constrained directions θ_d, $d = 0, \ldots, D-1$, i.e., $\mathbf{C} = [\mathbf{a}(\theta_0), \mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_{D-1})]$, $\mathbf{g}$ is a prescribed D-dimensional vector usually satisfying $(\mathbf{g})_1 = 1$. The solution of problem (51) is given by

$$\mathbf{w}_{\mathrm{LCMV}} = \mathbf{R}_{n+i}^{-1}\mathbf{C}(\mathbf{C}^{\mathrm{H}}\mathbf{R}_{n+i}^{-1}\mathbf{C})^{-1}\mathbf{g}. \tag{52}$$

From (51b), we can clearly see that both the amplitude and the phase of the array output, i.e., $\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta)$, have been strictly constrained at θ_d, $d = 0, \ldots, D-1$. As a matter of fact, a less restrictive quadratically constrained minimum variance (QCMV) beamformer should be formulated by removing the unnecessary phase constraints, i.e.,

$$\min_{\mathbf{w}} \ \mathbf{w}^{\mathrm{H}}\mathbf{R}_{n+i}\mathbf{w} \tag{53a}$$

$$\text{subject to } |(\mathbf{C}^{\mathrm{H}}\mathbf{w})_d|^2 = |(\mathbf{g})_d|^2,\ d = 1, \ldots, D. \tag{53b}$$

Note that in this subsection the variable d is an index and does not mean "desired" as used previously. Comparing to the QCMV in (53), we can see that the LCMV beamformer in (51) strictly limits the optimization of the weight vector to a smaller space, although it has a closed-form solution. It, thus, may cause the output SINR of LCMV beamformer to suffer from a loss, and the resulting pattern may be distorted.

We adopt the multi-point OPARC algorithm to solve the QCMV problem (53), in the hope that the resulting output SINR can be improved (comparing to LCMV). If $D = 1$, i.e., one constraint $|\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}|^2 = 1$ is imposed in (53b), the optimal solution of (53) is given by

$$\mathbf{w} = \frac{\mathbf{R}_{n+i}^{-1}\mathbf{a}(\theta_0)}{\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{R}_{n+i}^{-1}\mathbf{a}(\theta_0)}. \tag{54}$$

If $D > 1$, based on the first constraint that $|\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}|^2 = 1$, we have $L(\theta_{d-1}, \theta_0) = |\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_{d-1})|^2$ in (53b). Then, the additional $(D-1)$ constraints can be taken into account by imposing the following constraints:

$$L(\theta_{d-1}, \theta_0) = |(\mathbf{g})_d|^2,\ d = 2, \ldots, D. \tag{55}$$

Then, the problem becomes how to realize the above described multi-point response control, starting from the optimal weight vector in (54). To apply the multi-point OPARC algorithm, we

Algorithm 5: Multi-point OPARC based Multi-constraint Adaptive Beamforming Algorithm.

1: give interference number J_r, constraint matrix $\mathbf{C}$ and vector $\mathbf{g}$, estimate $\hat{\mathbf{R}}_{n+i}$ and $\hat{\sigma}_n^2$ by (57) and (58), respectively, calculate $\mathbf{T}_{n+i} = \hat{\mathbf{R}}_{n+i}/\hat{\sigma}_n^2$ and $\mathbf{w}_0 = \mathbf{T}_{n+i}^{-1}\mathbf{a}(\theta_0)$
2: apply multi-point OPARC algorithm to realize $L(\theta_{d-1}, \theta_0) = |(\mathbf{g})_d|^2$, $d = 2, \ldots, D$, by setting $\mathbf{T}_{n+i}$ and $\mathbf{w}_0$ as the initial VCM and the initial weight vector, respectively, to obtain $\mathbf{w}_{\mathrm{QC}}$

rewrite $\mathbf{w}$ in (54) as

$$\mathbf{w} = \frac{1}{\sigma_n^2 \mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{R}_{n+i}^{-1}\mathbf{a}(\theta_0)}\mathbf{T}_{n+i}^{-1}\mathbf{a}(\theta_0) \triangleq c\mathbf{w}_0 \tag{56}$$

where c is a constant satisfying $c = (\sigma_n^2 \mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{R}_{n+i}^{-1}\mathbf{a}(\theta_0))^{-1}$, $\mathbf{T}_{n+i}$ and $\mathbf{w}_0 = \mathbf{T}_{n+i}^{-1}\mathbf{a}(\theta_0)$ act as the initial VCM in (4) and the initial weight vector in multi-point OPARC, respectively. Then, a multi-point OPARC procedure can be applied to fulfill the response requirement described in (55), and the ultimate weight vector of QCMV (denoted as $\mathbf{w}_{\mathrm{QC}}$) can be obtained accordingly.

Note that in practical applications, $\mathbf{R}_{n+i}$ can be estimated from data $\mathbf{x}(t)$:

$$\hat{\mathbf{R}}_{n+i} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t) \tag{57}$$

where T is the number of snapshots. In addition, σ_n^2 can be estimated by [34]

$$\hat{\sigma}_n^2 = \frac{1}{N - J_r}\sum_{n=J+1}^{N}\lambda_n \tag{58}$$

where J_r is the number of interferences, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ are eigenvalues of $\hat{\mathbf{R}}_{n+i}$. Replacing $\mathbf{R}_{n+i}$ and σ_n^2 with $\hat{\mathbf{R}}_{n+i}$ and $\hat{\sigma}_n^2$, respectively, we have summarized the proposed algorithm in Algorithm 5.

To have a better understanding, we denote the corresponding VCM of $\mathbf{w}_{\mathrm{QC}}$ as $\mathbf{T}_{\mathrm{QC}}$. Recalling the property (39) of multi-point OPARC, $\mathbf{w}_{\mathrm{QC}}$ and $\mathbf{T}_{\mathrm{QC}}$ satisfy

$$\mathbf{w}_{\mathrm{QC}} = \mathbf{T}_{\mathrm{QC}}^{-1}\mathbf{a}(\theta_0). \tag{59}$$

We can see that the obtained weight $\mathbf{w}_{\mathrm{QC}}$ minimizes the total variance $\mathbf{w}^{\mathrm{H}}\mathbf{T}_{\mathrm{QC}}\mathbf{w}$ with the constraints (53b), rather than minimizing $\mathbf{w}^{\mathrm{H}}\mathbf{T}_{n+i}\mathbf{w}$ or its equivalent term $\mathbf{w}^{\mathrm{H}}\mathbf{R}_{n+i}\mathbf{w}$ (for a fixed σ_n^2) in (53a). Nevertheless, we know from Proposition 7 of the companion paper [1] that the obtained weight of OPARC also minimizes the variance at the previous step. Thus, $\mathbf{w}_{\mathrm{QC}}$ is the optimal solution of problem (53) for the special case when $D = 2$, i.e., only one extra constraint is imposed besides the constraint $|\mathbf{a}^{\mathrm{H}}(\theta_0)\mathbf{w}|^2 = 1$. In addition, the obtained $\mathbf{w}_{\mathrm{QC}}$ offers the optimal solution of problem (53) if we impose null constraint at θ_{d-1}, $d = 2, \ldots, D$, based on the following argument. In this case, we set $|(\mathbf{g})_d|^2 = 0$, $d = 2, \ldots, D$, and thus obtain (60) shown at the bottom of this page, where we have used the fact that

$$\frac{|\mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\mathbf{a}(\theta_{d-1})|^2}{|\mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2} = |(\mathbf{g})_d|^2 = 0,\ d = 2, \ldots, D \tag{61}$$

and

$$\mathbf{T}_{\mathrm{QC}} = \mathbf{T}_{n+i} + \sum_{d=2}^{D}\beta_{d-1}\mathbf{a}(\theta_{d-1})\mathbf{a}^{\mathrm{H}}(\theta_{d-1}) \tag{62}$$

with β_{d-1} denoting the INR of the assigned virtual interference at θ_{d-1}. From (60) we know that $\mathbf{w}_{\mathrm{QC}}$ also minimizes $\mathbf{w}^{\mathrm{H}}\mathbf{R}_{n+i}\mathbf{w}$. The optimality (in the sense of output SINR) of the proposed algorithm is guaranteed in the above two scenarios. Otherwise, the proposed algorithm performs better than LCMV algorithm in most cases as we shall see from the simulations later.

Moreover, (59) and (62) indicate that the resulting weight vector $\mathbf{w}_{\mathrm{QC}}$ is obtained by making a normalized covariance matrix loading (NCL), which can be regarded as a generalization of the diagonal loading (DL) in [23]–[25], on the initial $\mathbf{T}_{n+i}$. The loading quantity is precisely determined by multi-point OPARC algorithm as

$$\mathbf{\Delta} = \sum_{d=2}^{D}\beta_{d-1}\mathbf{a}(\theta_{d-1})\mathbf{a}^{\mathrm{H}}(\theta_{d-1}). \tag{63}$$

Recalling Eqn. (42) of [1], one learns in OPARC that the INR of a newly assigned virtual interference depends on the previous normalized covariance matrix and also contributes to the current one. Then, revisiting Algorithm 1, where OPARC is iteratively applied, and Eqn. (16), one can see that the resulting β_{d-1}, $d = 2, \ldots, D$, depend on the initial $\mathbf{T}_{n+i}$. Thus, the loading quantity $\mathbf{\Delta}$ in (63) is related to the given constraints in (53b) and also the real data.

Note that the above-described multi-constraint adaptive beamforming algorithm improves the robustness of array systems while blocking the unexpected interference and noise. Our algorithm removes the unnecessary phase constraints on the beamformer output and can make the prescribed amplitude constraints satisfied. This is not true for the traditional adaptive array approaches, such as, generalized sidelobe cancellation (GSC) technique [35] and the minimum variance distortionless

$$\mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\mathbf{T}_{\mathrm{QC}}\mathbf{w}_{\mathrm{QC}} = \mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\left(\mathbf{T}_{n+i} + \sum_{d=2}^{D}\beta_{d-1}\mathbf{a}(\theta_{d-1})\mathbf{a}^{\mathrm{H}}(\theta_{d-1})\right)\mathbf{w}_{\mathrm{QC}} = \mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\mathbf{T}_{n+i}\mathbf{w}_{\mathrm{QC}} + \underbrace{|\mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\mathbf{a}(\theta_0)|^2\left(\sum\nolimits_{d=2}^{D}\beta_{d-1}|(\mathbf{g})_d|^2\right)}_{=0}$$

$$= \mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\mathbf{T}_{n+i}\mathbf{w}_{\mathrm{QC}} = \frac{\mathbf{w}_{\mathrm{QC}}^{\mathrm{H}}\mathbf{R}_{n+i}\mathbf{w}_{\mathrm{QC}}}{\sigma_n^2} \tag{60}$$

response (MVDR) beamformer [14], which can be seen as an equivalent and a simplified version of the LCMV beamformer, respectively. However, different from the method in the preceding subsection where the sidelobe peaks can be controlled iteratively, the algorithm in this subsection only has constraints on the response levels of several pre-assigned angles $\theta_0, \theta_1, \ldots, \theta_{D-1}$. It cannot control/guarantee an overall sidelobe pattern.

C. Quiescent Pattern Control

In adaptive beamforming, weight vector is designed in a data-dependent manner. However, the traditional adaptive beamforming methods usually yield a beampattern with high sidelobes. To obtain low sidelobes in adaptive arrays, the concept of quiescent pattern control is introduced in [26], by combining the adaptive beamforming and deterministic pattern synthesis techniques. In brief, when an adaptive array operates in the presence of white noise only, the resultant adaptive beamformer is named as the quiescent weight vector, and the corresponding array response is termed as the quiescent pattern. Following the concept of quiescent pattern control in [26]–[28], it is required to find a mechanism to design a beamformer having the ability to reject an interference (if it exists) and noise, and meanwhile, maintaining the desirable shape of the quiescent pattern when only white noise is present.

Note that the quiescent weight vector of LCMV beamformer in (52) is $\mathbf{w}_q = \mathbf{C}(\mathbf{C}^{\mathrm{H}}\mathbf{C})^{-1}\mathbf{g}$ that can be readily obtained by setting $\mathbf{R}_{n+i} = \sigma_n^2\mathbf{I}$. Unfortunately, for a given desired quiescent pattern, which usually has specific constraints on the upper level of sidelobes, it is not easy to have a satisfactory quiescent pattern via LCMV by specifying $\mathbf{C}$ and $\mathbf{g}$, since LCMV only imposes constraints on a fixed set of pre-assigned finite angles as mentioned at the end of Section III-B. This is similarly true for the multi-point OPARC algorithm presented in the preceding Section III-B. Moreover, if we employ the iterative approach adopted in deterministic pattern synthesis in Section III-A to modify the shape of the obtained beampattern, nulls may not be always formed at the directions of unknown real interferences, and the adaptivity in suppressing undesirable components is thus not well guaranteed.

In this subsection, a systematic approach to quiescent pattern control is proposed. A two-stage procedure is developed, by taking advantage of the deterministic pattern synthesis approach in Section III-A and also the concept of NCL mentioned in Section III-B. More specifically, given a desired quiescent pattern, denoted as $L_d(\theta)$, the multi-point OPARC based pattern synthesis algorithm in Section III-A, see, Algorithm 4, is adopted in the first stage to design a desirable quiescent pattern off-line. Denote by $\mathbf{w}_q$, $\mathbf{T}_q$ and $L_q(\theta, \theta_0)$ the obtained (quiescent) weight vector, the associated VCM and the resulting response pattern, respectively. One can readily know that

$$\mathbf{w}_q = \mathbf{T}_q^{-1}\mathbf{a}(\theta_0). \tag{64}$$

As mentioned earlier, the resulting $L_q(\theta, \theta_0)$ performs well in maintaining the shape of $L_d(\theta)$, however, the above weight $\mathbf{w}_q$ has no ability to reject the potential interferences and noise. A strategy of finding weight vector is thus required in quiescent pattern control to, not only maintain the shape of $L_d(\theta)$ if only white noise exists, but also suppress a possible real interference and noise. From the adaptive array theory, a data-dependent loading quantity $\mathbf{\Delta}$ needs to be added to the VCM $\mathbf{T}_q$, such that the potential interferences and noise can be rejected. Moreover, in the white noise only case, $\mathbf{\Delta}$ should be zero such that the weight $\mathbf{w}_q$ in (64) can be retrieved. To do so, we carry out the second stage, by taking real data into consideration and carrying out an NCL operator to the VCM $\mathbf{T}_q$ via setting the associated loading quantity $\mathbf{\Delta}$ as

$$\mathbf{\Delta} = -\mathbf{I} + \mathbf{T}_{n+i} \tag{65}$$

where $\mathbf{T}_{n+i} = \mathbf{R}_{n+i}/\sigma_n^2$. The ultimate (adaptive) weight vector is thus calculated as

$$\mathbf{w}_a = (\mathbf{T}_q - \mathbf{I} + \mathbf{T}_{n+i})^{-1}\mathbf{a}(\theta_0). \tag{66}$$

The corresponding response pattern of $\mathbf{w}_a$ (denoted as $L_a(\theta, \theta_0)$) can be obtained accordingly.

One can see that there are two components being suppressed by $\mathbf{w}_a$ in (66). The first one is the component of the virtual interference which corresponds to $\mathbf{T}_q - \mathbf{I}$ and helps to maintain the shape of $L_d(\theta)$. The second component is $\mathbf{T}_{n+i}$, which contains the real interference and noise that need to be rejected. In the noise only scenario, the loading quantity $\mathbf{\Delta}$ offsets zero automatically and the quiescent weight vector $\mathbf{w}_q$ in (64) appears, provided that the real noise shares the same structure as the virtual noise, i.e., $\mathbf{R}_{n+i} = \sigma_n^2\mathbf{I}$ or $\mathbf{T}_{n+i} = \mathbf{I}$. Therefore, we can see that the weight vector $\mathbf{w}_q$ in (64) and its corresponding beampattern $L_q(\theta, \theta_0)$ are exactly the quiescent weight vector and quiescent pattern, respectively. Also, we should replace the unknown $\mathbf{R}_{n+i}$ and σ_n^2 with $\hat{\mathbf{R}}_{n+i}$ in (57) and $\hat{\sigma}_n^2$ in (58), respectively, and set $\mathbf{T}_{n+i} = \hat{\mathbf{R}}_{n+i}/\hat{\sigma}_n^2$ in practical applications.

It should be emphasized that we do not impose extra constraints (e.g., fixed null constraints considered in [26]) on the resulting response pattern $L_a(\theta, \theta_0)$, since such kind of constraints can be aforehand considered in the first stage of the above procedure. In addition, we can also make the fixed constraints satisfied by performing the multi-point OPARC algorithm starting from the obtained $\mathbf{w}_a$ in (66) and its corresponding normalized covariance matrix $\mathbf{T} = \mathbf{T}_q - \mathbf{I} + \mathbf{T}_{n+i}$. This is similar to the idea used in the preceding subsection. To make it clear, we have summarized the multi-point OPARC based quiescent pattern control algorithm in Algorithm 6.

Algorithm 6: Multi-point OPARC based Quiescent Pattern Control Algorithm.

1: give $L_d(\theta)$, synthesize a desirable quiescent pattern $L_q(\theta, \theta_0)$ using Algorithm 4, obtain $\mathbf{w}_q$ and $\mathbf{T}_q$
2: estimate $\hat{\mathbf{R}}_{n+i}$ and $\hat{\sigma}_n^2$ by (57) and (58), respectively, set $\mathbf{T}_{n+i} = \hat{\mathbf{R}}_{n+i}/\hat{\sigma}_n^2$
3: obtain adaptive weight vector $\mathbf{w}_a$ by Eqn. (66)
4: if extra constraints needed, modify $\mathbf{w}_a$ by conducting the multi-point OPARC algorithm
5: output the obtained weight $\mathbf{w}_a$ and its corresponding response pattern $L_a(\theta, \theta_0)$

TABLE I
OBTAINED PARAMETER COMPARISON

	MA^{2}RC	Multi-point OPARC
$D_{2,1}$(dB)	21.3110	5.7620
$D_{2,2}$(dB)	13.5149	10.0816
J_2(dB)	17.4130	7.9218
G_1(dB)	10.0078	10.0078
G_2(dB)	9.9192	11.2550

IV. NUMERICAL RESULTS

We next present some simulations to demonstrate the proposed multi-point OPARC algorithm and its applications. Unless otherwise specified, we set $\omega = 6\pi \times 10^8$ rad/s and consider an 11-element non-uniformly spaced linear array with nonisotropic elements. Both the element locations x_n and the element patterns $g_n(\theta)$ are listed in Table I in Part I [1], and the same array configuration has been adopted in Part I [1]. The beam axis is steered to $\theta_0 = 20^\circ$. We set $\beta_\epsilon = 10^{-10}$ in conducting the iterative approach, and take $\delta = 10^{-15}$ and $\eta = 900$ for the C-ADMM approach. In addition, $\mathbf{f}_n$ is specified as the all-zero vector for the MA^{2}RC algorithm in [2] for comparison, SNR is taken as 10dB when it applies.

A. Illustration of Multi-Point OPARC

In this subsection, we demonstrate the multi-point OPARC algorithm. Both the iterative approach and the C-ADMM approach are conducted, and then compared with the MA^{2}RC algorithm. As mentioned in [1], an ideal criterion of array response control is to achieve the array response levels as desired, while keeping the responses at any other directions unchanged. Two metrics are developed in [1] to measure the performance of array response control. Following the evaluation strategy adopted in [1], we define

$$D_{k,m} \triangleq |L_k(\theta_{k-1,m}, \theta_0) - L_{k-1}(\theta_{k-1,m}, \theta_0)| \tag{67}$$

to measure the response level differences between two consecutive response controls at $\theta_{k-1,m}$, $m = 1, \ldots, M_{k-1}$, where $L_k(\theta, \theta_0)$ represents the resultant response after finishing the k-th step of weight update. Since the response level at $\theta_{k-1,m}$ has been adjusted in the $(k-1)$th step as its desired level $\rho_{k-1,m}$, one can rewrite $D_{k,m}$ as $D_{k,m} = |L_k(\theta_{k-1,m}, \theta_0) - \rho_{k-1,m}|$, $m = 1, \ldots, M_{k-1}$. In addition, we define J_k as

$$J_k \triangleq \frac{1}{M_{k-1}} \sum_{m=1}^{M_{k-1}} \left| L_k(\theta_{k-1,m}, \theta_0) - \rho_{k-1,m} \right| \tag{68}$$

which can be re-written as $J_k = (1/M_{k-1}) \sum_{m=1}^{M_{k-1}} D_{k,m}$ and measures the average difference between the resulting response and the desired one at the previously-controlled angles $\theta_{k-1,m}$'s, $m = 1, \ldots, M_{k-1}$, after the k-th step of array response control. Note that J_k is effective on measuring the level difference between the resulting response and the desired one, even if the initial or previous responses are unqualified. Besides, the index k should be not less than 2 when the metrics D_k and J_k apply.

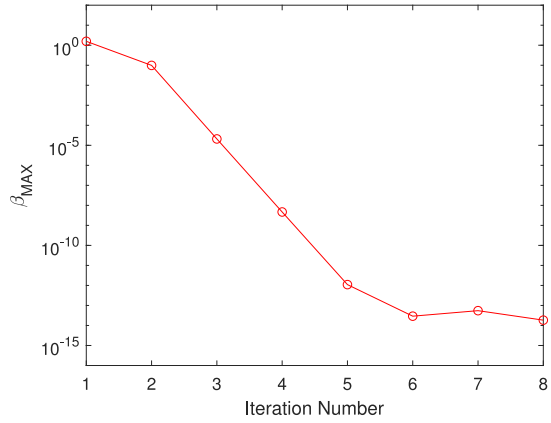


Fig. 1. Curve of β_{MAX} versus the iteration number.

For convenience, in the first example we carry out two steps of the array response control algorithms with each step controlling two angles, i.e., $M_1 = M_2 = 2$, and denote the adjusted angles and the corresponding desired levels of the kth ($k = 1, 2$) step as $\theta_{k,m}$ and $\rho_{k,m}$, $m = 1, \ldots, M_k$, respectively. More specifically, we set $\theta_{1,1} = -45^\circ$, $\rho_{1,1} = -40$ dB, $\theta_{1,2} = -5^\circ$ and $\rho_{1,2} = -30$ dB for the first step of the response control. Note that the same settings have been adopted in Section V.A in Part I [1], where the single-point response control is realized in sequence. In this part, we first conduct multi-point OPARC algorithm by using the iterative method described in Algorithm 1. In the first iteration, the OPARC algorithm in [1] is applied to control the responses of $\theta_{1,m}$ to their desired levels $\rho_{1,m}$, $m = 1, 2$, one-by-one on m. We have $\beta_{1,1,\star} = 1.5683$, $\beta_{1,2,\star} = 0.2504$, which is the same as the results obtained in Section V.A in Part I [1]. Then, we continue our multi-point OPARC algorithm by conducting the above iteration procedure for a number of times. The curve of β_{MAX} versus the iteration number is depicted in Fig. 1. Note that the parameter β_{MAX} measures the maximal magnitudes of INRs of the newly assigned virtual interferences in the current iteration, as shown in the 8th line of Algorithm 1. From Fig. 1, one can see that β_{MAX} decreases with iteration. Moreover, observation shows that it only requires five iterations to converge, i.e., $\beta_{\mathrm{MAX}} \leq \beta_\epsilon$, and the result is $\bar{\beta}_{1,1,\star} = 1.4700$ and $\bar{\beta}_{1,2,\star} = 0.2506$, which is, respectively, close to $\beta_{1,1,\star}$ and $\beta_{1,2,\star}$. Now we test the performance of the C-ADMM approach. The obtained δ_{MAX} in (32) reduces with the iteration, i.e., the procedure described in (29)–(31), as shown in Fig. 2, and $\delta_{\mathrm{MAX}} \leq \delta$ is met after about 130 iterations. We obtain $\mathbf{h}_{1,\star} = [-0.1458 - j0.0203, -0.0687 - j0.0397]^{\mathrm{T}}$. Not surprisingly, it can be checked that the results of the above two approaches correspond to the same weight vector. Hence, the same beampatterns are synthesized for these two approaches as shown in Fig. 3(a), from which one can see that the responses of the two adjusted angles have been precisely controlled to their desired values. Interestingly, when testing the MA^{2}RC, the resulting pattern is completely the same as that of the multi-point OPARC algorithm. We believe that this occurs not accidentally but with a reason that is, unfortunately, not clear yet.

In the second step of the response control, we take $\theta_{2,1} = 7^\circ$, $\rho_{2,1} = -25$ dB, $\theta_{2,2} = 28^\circ$ and $\rho_{2,2} = 0$ dB. When conducting

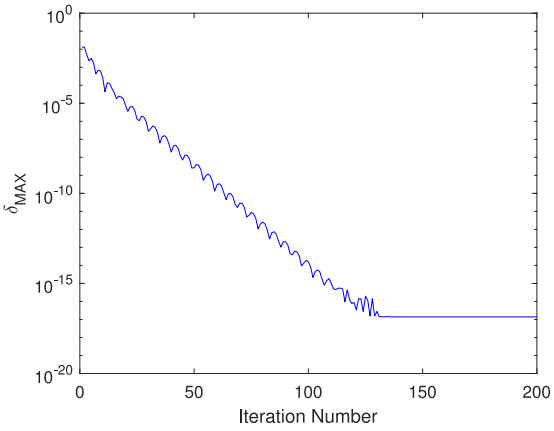
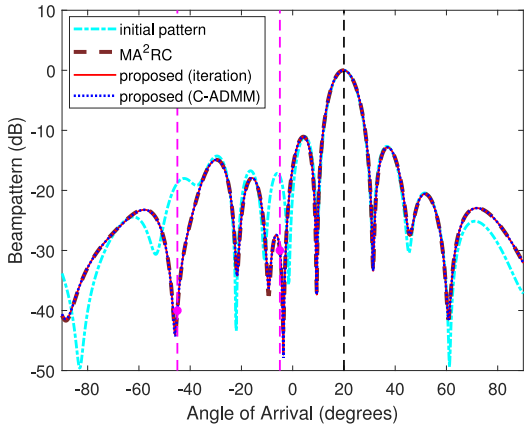
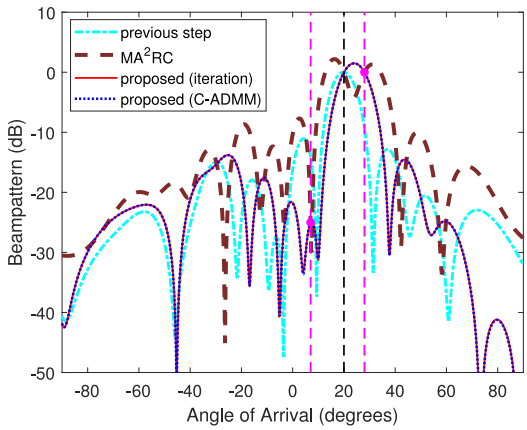


Fig. 2. Curve of δ_{MAX} versus the iteration number.



(a) The first step



(b) The second step

Fig. 3. Illustration of multi-point OPARC algorithm (the first example).

the multi-point OPARC algorithm, we obtain $\bar{\beta}_{2,1,\star} = 0.2555$ and $\bar{\beta}_{2,2,\star} = -0.0804$ for the iterative approach, and find $\mathbf{h}_{2,\star} = [-0.1803 - j0.0653, -0.5434 - j0.9252]^{\mathrm{T}}$ after implementing the C-ADMM method. Again, the above two sets of results correspond to the same beampattern as shown in Fig. 3(b), where the resulting pattern of MA^{2}RC is also displayed. From Fig. 3(b), one can see that all the adjusted angles have been accurately controlled as expected, for the three approaches. However,

TABLE II
SETTINGS OF $\theta_{k,m}$ AND $\rho_{k,m}$ $(k = 1, 2, m = 1, 2, 3, 4)$

m	$\theta_{1,m}$	$\rho_{1,m}$(dB)	m	$\theta_{2,m}$	$\rho_{2,m}$(dB)
1	$-78°$	-40	1	$7°$	0
2	$-63°$	-35	2	$25°$	-35
3	$-40°$	-30	3	$42°$	-40
4	$-22°$	-30	4	$75°$	-35

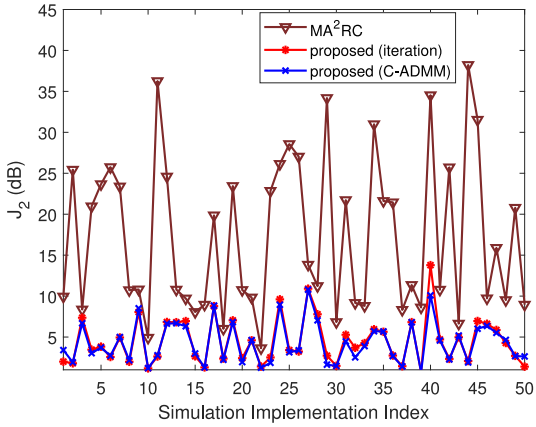
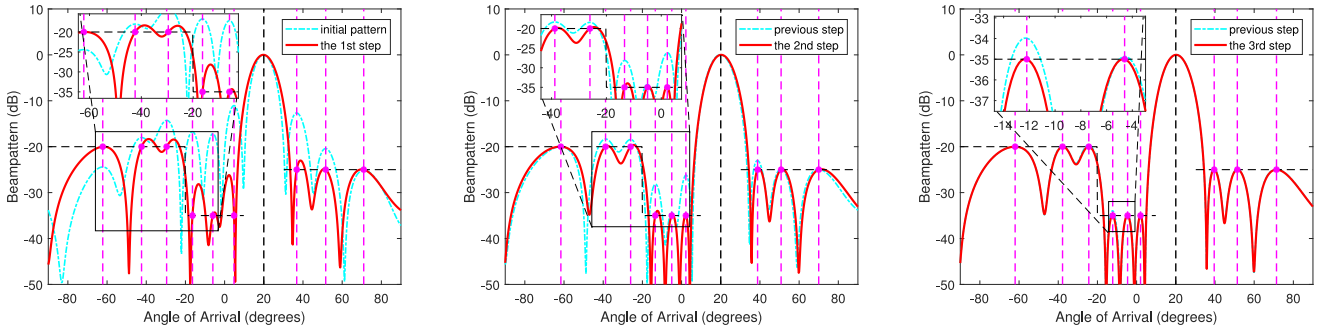


Fig. 4. Curves of J_2 versus implementation index (the second example).

the mainlobe of the ultimate pattern of MA^{2}RC is distorted and a high sidelobe level is resulted. For comparison purpose, we have listed several parameter measurements in Table I, from which one can see that the MA^{2}RC method brings large values on both $D_{2,m}$ $(m = 1, 2)$ and J_2, and results a less array gain compared to the proposed multi-point OPARC algorithm.

To show that our algorithms behave well not only under carefully chosen array configurations, we carry out the second example by randomly selecting the element number and positions, taking the influence of mutual coupling into consideration, and increasing the number of controlled points as well. In this case, we consider a linear array with isotropic elements. The beam axis is taken as $\theta_0 = 5°$. We carry out two response control steps, and pre-assign four angles and their corresponding desired levels in each procedure, see Table II for details. Following the study in Section V.C in Part I [1], we carry out Monte Carlo simulation with 50 realizations. More specifically, the element number N in each realization is randomly selected as a positive integer from 8 to 16. The element space between two adjacent sensors is distributed uniformly in the range $[0.4\lambda, 0.6\lambda]$. The mutual coupling matrix (denoted as $\mathbf{Z} \in \mathbb{C}^{N\times N}$) is complex symmetry with unit elements on diagonal. The amplitudes of other entries of $\mathbf{Z}$ are fixed as 0.1, and their phases are distributed uniformly in the range $[0, 2\pi)$. Although the configurations have been randomized, simulations show that the proposed multi-point OPARC algorithms can adjust the array response level as desired at each response control step. The curves of J_2 versus the implementation index are depicted in Fig. 4, from which one can see that our algorithms outperform MA^{2}RC method with less J_2 in each realization. Therefore, the proposed algorithms behave well not only under the circumstances of carefully chosen array configurations. In addition, observe from Fig. 4



(a) Synthesized pattern at the 1st step (b) Synthesized pattern at the 2nd step (c) Synthesized pattern at the 3rd step

Fig. 5. Resultant patterns at different steps when carrying out a nonuniform sidelobe synthesis for a nonuniform linear array.

that the results of the proposed two approaches may be slightly different in a few cases when using randomized array settings. The possible reason is that both approaches only provide suboptimal solutions but not the global one, and the result of the second approach (C-ADMM) may depend on its initial setting, as we have clarified earlier.

B. Array Pattern Synthesis Using Multi-Point OPARC

Starting from this subsection, the applications of multi-point OPARC are simulated and the iterative approach in Section II. C is adopted to illustrate the results. In this subsection, we focus upon the application of multi-point OPARC to array pattern synthesis and give two representative examples for demonstration.

1) Nonuniform Sidelobe Synthesis: In the first example, the desired pattern has nonuniform sidelobes. Fig. 5 shows the synthesized patterns of the proposed algorithm at different steps. Clearly, in each synthesis step, all the sidelobe peaks, i.e., Ω_k in (48), are first determined from the previously synthesized pattern. Notice that the response level of a selected sidelobe peak can be either higher or lower (see Fig. 5(a) for reference) than its desired level. It has been shown in Fig. 5 that it only requires 3 steps, i.e., $k = 3$, to synthesize a satisfactory beampattern.

For comparison, the resulting patterns of the proposed algorithm, Philip's method in [13], convex programming (CP) method in [16], A^{2}RC method (after carrying out 30 steps) in [3] and MA^{2}RC method (after carrying out 3 steps) in [2] are displayed in Fig. 6. As expected, we can see that the pattern envelopes of Philip's method and CP method are not aligned with the desired level, since they cannot control the beampattern precisely according to the required specifications. Although A^{2}RC and MA^{2}RC have the ability to precisely control the given array responses, the obtained sidelobe peaks are not aligned with the desired ones either, since only the sidelobe peaks higher than the desired levels are selected and adjusted in these two approaches.

2) Large Array Consideration: In this example, pattern synthesis for a large linearly half-wavelength-spaced array with $N = 80$ isotropic elements is considered. The desired pattern steers at $\theta_0 = 50^\circ$ with nonuniform sidelobes. More specifically, the upper level is -35 dB in the sidelobe region $[-90^\circ, 50^\circ)$ and -25 dB in the rest of the sidelobe region.

Fig. 7 demonstrates several intermediate results of the proposed algorithm. In every step, we select $C_k = 20$ sidelobe peak

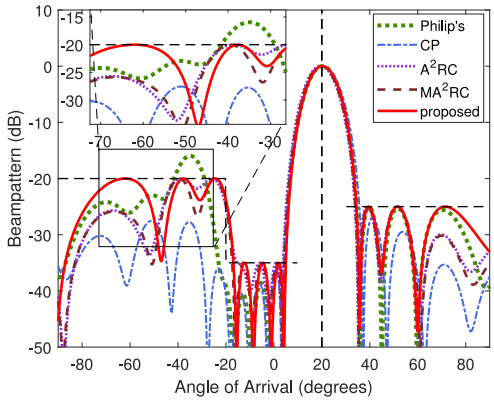


Fig. 6. Resultant pattern comparison.

TABLE III
EXECUTION TIME COMPARISON WHEN CONDUCTING A LARGE-ARRAY PATTERN SYNTHESIS

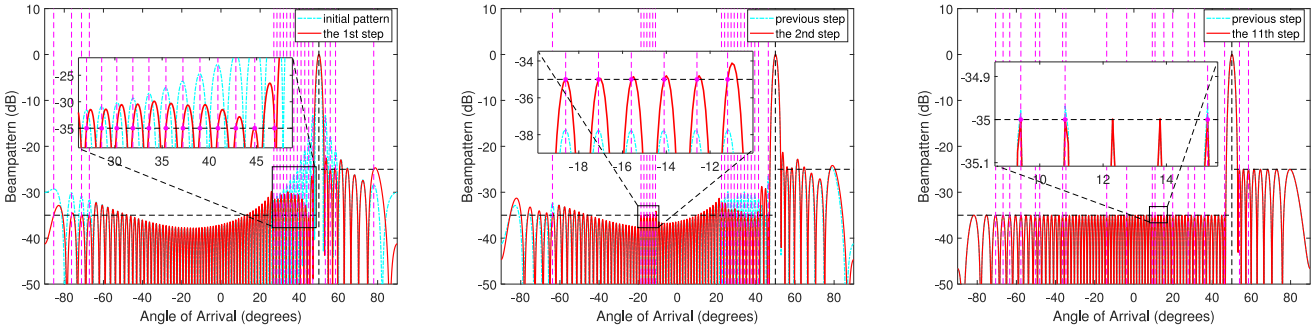
	Philip's	CP	A^{2}RC	MA^{2}RC	proposed
T(sec)	2.22	12.36	3.55	2.55	0.05

angles (see Eqn. (49) and (50) for details) and then adjust their responses to the desired levels by using multi-point OPARC algorithm. Simulation result shows that it only requires 11 steps, i.e., $k = 11$, to synthesize a qualified pattern, see the ultimate pattern in Fig. 7(c) for reference. The execution times of various methods are provided in Table III, where the superiority of the proposed algorithm can be clearly observed.

C. Multi-Constraint Adaptive Beamforming Using Multi-Point OPARC

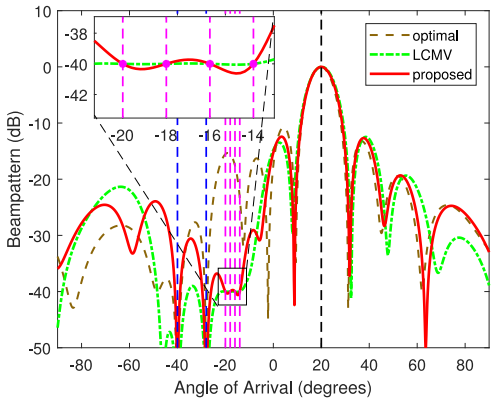
In this subsection, the multi-constraint adaptive beamforming is realized by using the multi-point OPARC algorithm. For simplicity, a perfect knowledge of the data covariance matrix is assumed.

1) Sidelobe Constraint: In the first case, four sidelobe constraints are required. More specifically, the response levels of -20°, -18°, -16° and -14° are expected to be all -40 dB. Two interferences are impinged from -40° and -28° with INRs 30 dB and 25 dB, respectively.

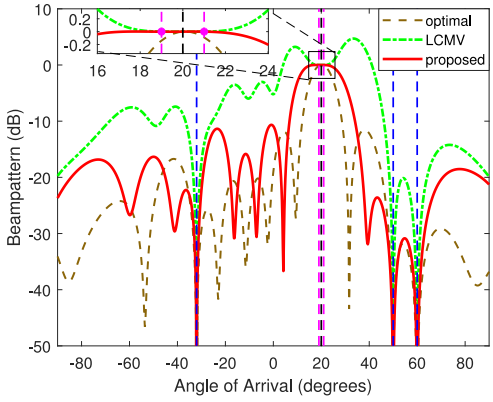


(a) Synthesized pattern at the 1st step (b) Synthesized pattern at the 2nd step (c) Synthesized pattern at the 11th step

Fig. 7. Resultant patterns at different steps when carrying out a nonuniform sidelobe synthesis for a large uniform linear array.



(a) The first case



(b) The second case

Fig. 8. Result comparison of multi-constraint adaptive beamforming.

Fig. 8(a) displays the results of the optimal beamformer with no sidelobe constraint, the LCMV method [20] and the proposed one. Clearly, both the LCMV beamformer and the proposed algorithm are able to shape deep nulls at the directions of interferences (see the blue line). Meanwhile, the given sidelobe constraints are well satisfied for both. When considering the output SINR, we have $\mathrm{SINR} = 19.5601$ dB for the LCMV method and $\mathrm{SINR} = 19.6906$ dB for the proposed one. We can see that the proposed beamformer brings an improvement on the output SINR compared to the LCMV beamformer.

2) Mainlobe Constraint: In the second case, two constraints are imposed in the mainlobe region. The constraint angles are 19° and 21°, and both of the desired levels are 0 dB. There are three interferences coming from -32°, 50° and 60° with an identical INR 30 dB.

Fig. 8(b) depicts the resultant patterns. One can see that the obtained pattern of the LCMV method is severely distorted, although the two prescribed constraints are satisfied and the three interferences are rejected. The corresponding output SINR is 11.1767 dB. Observing the resulting pattern of the proposed algorithm, the two-point constraint is well satisfied and a flat-top mainlobe is shaped with no distortion occurred. The corresponding output SINR is 17.1260 dB, which is much higher than that of the LCMV method.

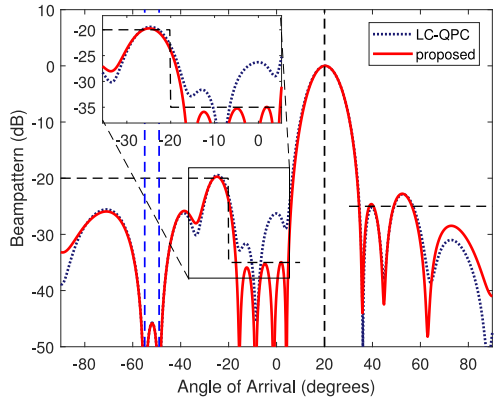
D. Quiescent Pattern Control Using Multi-Point OPARC

In this subsection, we test the performance of the multi-point OPARC based quiescent pattern control algorithm. The desired quiescent pattern has a nonuniform sidelobe level as depicted with black dash lines in Fig. 6.

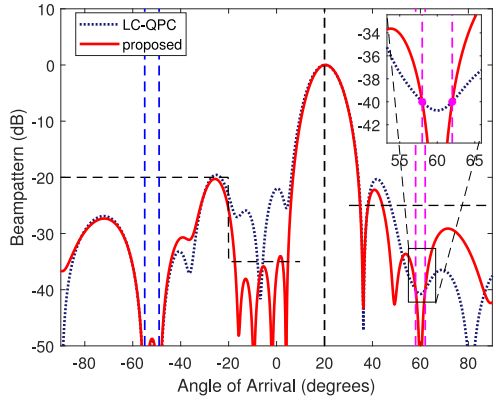
In our proposed algorithm, quiescent pattern synthesis and quiescent pattern control are jointly designed by the multi-point OPARC algorithm. We have detailed the off-line synthesis procedure in Section IV-B and illustrated the obtained quiescent pattern by red line in Fig. 6. Suppose that two interferences come from -55° and -49° with INRs 30 dB. The obtained adaptive response pattern is shown in Fig. 9(a), where we can observe that two nulls are formed at the directions of the real interferences, and the resultant sidelobe is close to the quiescent one. The obtained output SINR is 19.2984 dB for the proposed algorithm.

For comparison purpose, the classical linearly-constraint based quiescent pattern control approach (denoted as LC-QPC method for briefness) in [26] is also demonstrated, by using the same synthesized quiescent pattern in Fig. 6. The resulting pattern of LC-QPC is displayed in Fig. 9(a), where we find that an obvious perturbation is caused in the sector $[-15^\circ, 0^\circ]$ and the overall shape can not be well maintained compared to the desired one. The obtained output SINR is 19.2161 dB, which is lower than that of the proposed algorithm.

Now we take extra fixed constraints into consideration by restricting the response levels at directions 58° and 62° to be all -40 dB. The results of the proposed algorithm and the LC-QPC method are presented in Fig. 9(b), where we observe that both



(a) The first case



(b) The second case

Fig. 9. Result comparison of quiescent pattern control.

of these two methods are able to reject the undesirable interferences with the prescribed constraints being satisfied. The same as before, the proposed algorithm maintains a more desirable shape than that of the LC-QPC method. When taking the output SINR into account, the corresponding values are, respectively, 19.2382 dB (for the proposed algorithm) and 19.0967 dB (for the LC-QPC method). The advantage of the proposed algorithm is verified again.

V. Conclusions

In this paper, the optimal and precise array response control (OPARC) algorithm proposed in Part I [1] has been extended from a single point per step to a multi-points per step. Two computationally attractive multi-point OPARC algorithms have been proposed, by which the responses of multiple angles can be adjusted. In addition, several applications of the multi-point OPARC algorithm to array signal processing have been presented, and an innovate concept of normalized covariance matrix loading (NCL) has been developed. Simulation results have been provided to validate the effectiveness and superiority of the proposed algorithms under different situations.

Acknowledgment

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

References

[1] 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, 2019.

[2] 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.

[3] 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.

[4] K. Huang and N. D. Sidiropoulos, "Consensus-ADMM for general quadratically constrained quadratic programming," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5297–5310, Oct. 2016.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

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

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

[16] 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.

[17] W. Fan, 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.

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

[19] K. Yang, Z. Zhao, and Q. H. Liu, "Fast pencil beam pattern synthesis of large unequally spaced antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 61, no. pp. 627–634, Feb. 2013.

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

[21] 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.

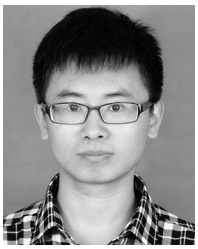
[22] C. Y. Chen and P. P. Vaidyanathan, "Quadratically constrained beamforming robust against direction-of-arrival mismatch," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4139–4150, Aug. 2007.

[23] 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.

[24] B. D. Carlson, "Covariance matrix estimation errors and diagonal loading in adaptive arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 24, no. 4, pp. 397–401, Jul. 1988.

[25] J. Li, P. Stoica, and Z. Wang, "On robust Capon beamforming and diagonal loading," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1702–1715, Jul. 2003.

[26] L. Griffiths and K. Buckley, "Quiescent pattern control in linearly constrained adaptive arrays," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 7, pp. 917–926, Jul. 1987.
[27] B. D. V. Veen, "Optimization of quiescent response in partially adaptive beamformers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 3, pp. 471–477, Mar. 1990.
[28] 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.
[29] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.
[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Ecksein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[31] J. Park and S. Boyd, "General heuristics for nonconvex quadratically constrained quadratic programming," Mar. 2017. [Online]. Available: http://stanford.edu/boyd/papers/pdf/qcqp.pdf
[32] M. Hong, M. Razaviyayn, Z. Q. Luo, and J. S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, Jan. 2016.
[33] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Math. Program.*, vol. 117, pp. 387–423, 2009.
[34] A. Liu, G. Liao, C. Zeng, Z. Yang, and Q. Xu, "An eigenstructure method for estimating DOA and sensor gain-phase errors," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 5944–5956, Dec. 2011.
[35] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 1, pp. 27–34, Jan. 1982.



Xuejing Zhang (S'17) was born in Hebei, China. He received the B.S. degree in electrical engineering from Huaqiao University, Xiamen, China, and the M.S. degree in signal and information processing from Xidian University, Xi'an, China, in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree in signal and information processing with the Department of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China. Since November 2017, he has been a Visiting Student with the University of Delaware, Newark, DE, USA. From 2014 to 2015, he was a Research Engineer with Allwinner, Inc., Zhuhai, China, where he was engaged in algorithmic research. His research interests include array signal processing, optimization theory, and machine learning.

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, Chengdu, China, in 1984, 1988, and 2000, respectively.

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

Xiang-Gen Xia (M'97–SM'00–F'09) received the B.S. degree in mathematics from Nanjing Normal University, Nanjing, China, and the M.S. degree in mathematics from Nankai University, Tianjin, China, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1983, 1986, and 1992, respectively.

He was a Senior/Research Staff Member with Hughes Research Laboratories, Malibu, CA, USA, during 1995–1996. In September 1996, he joined the Department of Electrical and Computer Engineering, University of Delaware, Newark, Delaware, where he is the Charles Black Evans Professor. His current research interests include space-time coding, multiple-input multiple-output (MIMO) and orthogonal frequency division multiplexing (OFDM) systems, digital signal processing, and synthetic aperture radar (SAR) and inverse synthetic aperture radar (ISAR) imaging. He is the author of the book *Modulated Coding for Intersymbol Interference Channels* (Marcel Dekker, 2000).

Dr. Xia was a recipient of the National Science Foundation Faculty Early Career Development (CAREER) Program Award in 1997, the Office of Naval Research Young Investigator Award in 1998, and the Outstanding Overseas Young Investigator Award from the National Nature Science Foundation of China in 2001. He is currently serving and has served as an Associate Editor for numerous international journals including IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is the Technical Program Chair of the Signal Processing Symp., Globecom 2007 in Washington, DC, and the General Co-Chair of International Conference on Acoustics, Speech, and Signal Processing 2005 in Philadelphia.

Bin Liao (S'09–M'13–SM'16) received the B.Eng. and M.Eng. degrees from Xidian University, Xian, 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, 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, Hong Kong. He is currently an Associate Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China.

Dr. Liao 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). His research interests include sensor array processing, adaptive filtering, convex optimization, with applications to radar, navigation, and communications. He is an Associate Editor for the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, *IET Signal Processing, Multidimensional Systems and Signal Processing*, and IEEE ACCESS.

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

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 electronic engineering with UESTC. Her research interests include synthetic aperture radar imaging, sparse signal reconstruction, and statistical signal processing.