Progress In Electromagnetics Research B, Vol. 53, 267–290, 2013

A MULTI-BEAM MODEL OF ANTENNA ARRAY PATTERN SYNTHESIS BASED ON CONIC TRUST REGION METHOD

Tiao-Jun Zeng and Quan-Yuan Feng*

School of Information Science and Technology, Southwest Jiaotong University, Sichuan 610031, China

Abstract—In this paper, we propose a multi-beam model for antenna array pattern synthesis (AAPS) problem. The model uses a conic trust region algorithm (CTRA) similarly proposed in this paper to optimize its cost function. Undoubtedly, whole algorithm efficiency ultimately lies on the CTRA, thereof, we propose a method to improve the iterative algorithm's efficiency. Unlike traditional trust region methods that resolve sub-problems, the CTRA efficiently searches a region via solving a inequation, by which it identifies new iteration points when a trial step is rejected. Thus, the proposed algorithm improves computational efficiency. Moreover, the CTRA has strong convergence properties with the local superlinear and quadratic convergence rate under mild conditions, and exhibits high efficiency and robustness. Finally, we apply the combinative algorithm to AAPS. Numerical results show that the method is highly robust, and computer simulations indicate that the algorithm excellently performs AAPS problem.

1. INTRODUCTION

Although array pattern synthesis problems have been extensively investigated over the last several decades [1–9], most of them are single-beam methods, and very few focus on multi-beam case [10, 11]. A multi-beam antenna is an antenna which is able to create multibeams simultaneously. Since the size of antennas is fixed by physical considerations, the difficulty of accommodating them on satellites and other facilities with multiple antenna-based functions grows with the number of such functions if a separate antenna

Received 15 June 2013, Accepted 31 July 2013, Scheduled 2 August 2013
* Corresponding author: Quan-Yuan Feng (fengquanyuan@163.com).

is used for each. In this context, it is desirable for a single antenna to be able to generate multiple radiation patterns, with the excitation corresponding to each being selectable by suitable switching devices [12]. Multibeam antennas can be considered as effective approach to perform independent data streams between a transmitter and a receiver.

In this paper, we propose a model to multi-beam AAPS, and this model is the extension of our previous work [7]. Under this model, the AAPS problem can be transformed into an unconstrained optimization problem. In recent years, the global optimization techniques, such as particle swarm optimization and differential evolution, have been used for AAPS problem [6, 13]. However, these algorithms are too time-consuming and limited in their application. Therefore, it is necessary to design a time-saving algorithms. Trust-region method of quadratic model for unconstrained optimization has been studied by many researchers [14–17]. Trust-region methods are robust, can be applied to ill-conditioned problems and have strong global convergence properties.

Instead of quadratic approximation to objective function, Davidon [18] proposed conic model to approximate the objective function. Di and Sun [19] first presented a trust region method based on conic model for unconstrained optimization. The conic model has several advantages. First, if the objective function has strong non-quadratic behavior or its curvature changes severely, the quadratic model methods often produce a poor prediction of the minimizer of the function. In this case, conic model approximates the objective function better than a quadratic approximation, because it has more freedom in the model. Second, the quadratic model does not take into account the information concerning the function value in the previous iteration which is useful for algorithms. However, the conic model possesses richer interpolation information and satisfies four interpolation conditions of the function values and the gradient values at the current and the previous points. Using these rich interpolation information may improve the performance of the algorithms. Third, the initial and limited numerical results provided in [19] etc. show that the conic model methods give improvement over the quadratic model ones. Finally, the conic model methods have the similar global and local convergence properties as the quadratic model ones [20].

Recently, Many researchers have recently studied nonmonotone adaptive conic trust region methods (NACTRM) for unconstrained optimization problems [16, 19, 20]. NACTRM can automatically produce an adaptive trust region radius whenever a trial step is rejected, and decreases functional values after finite iterations. The main disadvantage of NACTRM lies in identifying new trial iterations; it requires considerable computational time to repeatedly solve sub-problems. Motivated by rectifying this shortcoming, we propose a solving-inequality conic trust region method (SICTRM) that adopts a different search approach at each iteration. The search direction $d_k = x_{k+1} - x_k$ is generated by solving the sub-problem of the cost function. If d_k is rejected, the sub-problem does not need to be resolved. The search direction is generated by solving an inequality (details in Section 3).

The rest of the paper is organized as follows. Section 2 presents the Multi-beam model. In Section 3, SICTRM is introduced and then proven as a well-defined algorithm with its convergence properties. Section 4 shows the experimental results and related discussions. Discussion and Conclusions are drawn in Sections 5 and 6, respectively. Notation: $(\cdot)^T$ is the transpose, $(\cdot)^*$ is the complex conjugate, and $\|\cdot\|$ represents the Frobenius norm.

2. MULTI-BEAM MODEL

The array factor for a linear array with N isotropic elements can be expressed as:

$$p(W,\theta) = \sum_{i=1}^{N} w_i e^{j\phi_i(\theta)} = W^T S(\theta), \quad \theta \in [-90^\circ, 90^\circ] \tag{1}$$

where θ is the direction of arrival of signal, $j = \sqrt{-1}$, $S(\theta) = \begin{bmatrix} 1, & e^{j\varphi_2(\theta)}, & \dots, & e^{j\varphi_2(\theta)} \end{bmatrix}^T$ the steering vector, $\phi_i(\theta) = 2\pi d_i \sin\theta/\lambda$ the phase delay due to propagation, λ the wavelength of the transmitted signal, d_i the position of the ith element of the antenna array ($d_1 = 0$), and $W = [\ w_1, \ w_2, \ \dots, \ w_N\ ]^T$ the complex-weight vector. Since the optimization approach is only applicable to real variable problems, a complex-to-real transform of the $p(W,\theta)$ is necessary. Firstly, introducing a power function $(P(W,\theta))$ of the AF, i.e., $P(W,\theta) = |p(W,\theta)|^2 = p(W,\theta)p^*(W,\theta)$, then denote $W_1 = [w_1^1 \ w_2^1 \ \dots \ w_N^1]^T$, $W_2 = [w_1^2 \ w_2^2 \ \dots \ w_N^2]^T$, $S(\theta) = S_1(\theta) + jS_2(\theta)$, $S_1(\theta) = [1 \ \cos(\phi_2(\theta)) \ \dots \ \cos(\phi_N(\theta))\ ]^T$, $S_2(\theta) = [0 \ \sin(\phi_2(\theta)) \ \dots \ \sin(\phi_N(\theta))]^T$, W_1, W_2, $S_1(\theta)$ and $S_2(\theta) \in \mathbb{R}^{\mathbb{N}}$. Thus, we obtain [7]

$$\begin{aligned} P(W,\theta) = & \big[(W_1^T S_1)^2 + 2(W_1^T S_1)(W_2^T S_2) + (W_2^T S_2)^2 + (W_1^T S_2)^2 \\ & -2(W_1^T S_2)(W_2^T S_1) + (W_2^T S_1)^2\big] \in \mathbb{R} \end{aligned} \tag{2}$$

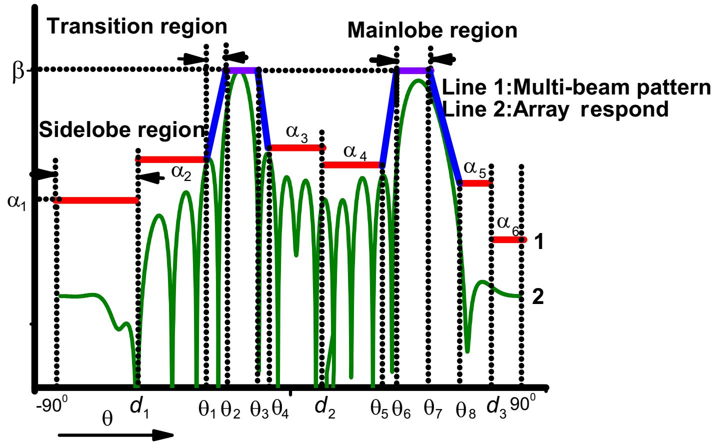


Figure 1. Definition of multi-beam synthesis.

In the case where phase constraint is not considered, an optimal weight vector, W_{opt}, is determined so that the pown (or amplitude) response $P(W,\theta)$ (or $|p(W,\theta)|$) best approximates the desired pattern. The definition of multi-beam synthesis is illustrated in Figure 1, we propose a method for synthesizing multi-beam pattern of antenna array, it is formulated by a discreted angle penalty function based on pown response of antenna array as following:

$$F(W,\mathrm{A}) = F_1(W,\mathrm{A}) + F_2(W,\mathrm{A}) \tag{3}$$

$$\begin{aligned} F_1(W,\mathrm{A}) = &\sum_{-90}^{d_1-\Delta\theta}\left[(\alpha_1-P)^2+\left(\frac{P}{\alpha_1}\right)^m\right]+\sum_{d_1}^{\theta_1-\Delta\theta}\left[(\alpha_2-P)^2+\left(\frac{P}{\alpha_2}\right)^m\right] \\ &+\sum_{\theta_1}^{\theta_2-\Delta\theta}\left[(f_1-P)^2+\left(\frac{P}{f_1}\right)^m\right]+\sum_{\theta_2}^{\theta_3}(\beta-P)^2 \\ &+\sum_{\theta_3-\Delta\theta}^{\theta_4-\Delta\theta}\left[(f_2-P)^2+\left(\frac{P}{f_2}\right)^m\right]+\sum_{\theta_4}^{d_2-\Delta\theta}\left[(\alpha_3-P)^2+\left(\frac{P}{\alpha_3}\right)^m\right] \end{aligned} \tag{4}$$

$$\begin{aligned} F_2(W,\mathrm{A}) = &\sum_{d_2}^{\theta_5-\Delta\theta}\left[(\alpha_4-P)^2+\left(\frac{P}{\alpha_4}\right)^m\right]+\sum_{\theta_5}^{\theta_6-\Delta\theta}\left[(f_3-P)^2+\left(\frac{P}{f_3}\right)^m\right] \\ &+\sum_{\theta_6}^{\theta_7}(\beta-P)^2+\sum_{\theta_7-\Delta\theta}^{\theta_8-\Delta\theta}\left[(f_4-P)^2+\left(\frac{P}{f_4}\right)^m\right]+\sum_{\theta_8}^{d_3-\Delta\theta}\left[(\alpha_5-P)^2\right. \end{aligned}$$

$$+\left(\frac{P}{\alpha_5}\right)^m\Bigg]+\sum_{d_3}^{90}\left[(\alpha_6-P)^2+\left(\frac{P}{\alpha_6}\right)^m\right]-\lambda\sum_{i=1}^{6}\lg\alpha_i \qquad (5)$$

where $\Delta\theta$ is the angle resolution, $P = P(W,\theta)$ defined by Eq. (2), constant $m > 0$, $\beta = r\sum\alpha_i$ (r is a given value, in general, and $r \geqslant 20$ will satisfy the condition: $\beta \geqslant 100\alpha_i$), $f_1 = \frac{(\beta-\alpha_2)}{(\theta_2-\theta_1)}(\theta-\theta_1)+\alpha_2$, $f_2 = \frac{(\beta-\alpha_3)}{(\theta_3-\theta_4)}(\theta-\theta_4)+\alpha_3$, $f_3 = \frac{(\beta-\alpha_4)}{(\theta_6-\theta_5)}(\theta-\theta_4)+\alpha_4$, $f_4 = \frac{(\beta-\alpha_5)}{(\theta_7-\theta_8)}(\theta-\theta_8)+\alpha_5$. Similar roles of penalty terms f_i, $\frac{P}{\alpha_j}$ and $\lg\alpha_j$ are mentioned in reference [7]. When $F(W,\theta)\to\varepsilon$ (ε is a very small constant), it obtain

$$\forall\theta\in[-90,\theta_1)+(\theta_4,\theta_5)+(\theta_8,90],\quad \left[(\alpha_i-P(W,\theta))^2+\left(\frac{P(W,\theta)}{\alpha_i}\right)^m\right]\to\varepsilon$$

$$\forall\omega\in[\theta_2,\theta_3]+[\theta_6,\theta_7],\quad (\beta-P(W,\omega))^2\to\varepsilon$$

i.e., $P(W,\theta) < \alpha_i$ and $P(W,\omega) = \beta$, thereof, $|PSLL| = |10*\log(\frac{P(W,\omega)}{P(W,\theta)})| > |10*\log(\frac{\beta}{\alpha_i})| > 20\,\mathrm{dB}$ ($\because \beta > 100\alpha_i$).

Above analysis indicates that $F\to\varepsilon$ is the sufficient condition of weight vector W converging to a optimal solution. In practice, the converging condition is too hard to achieve, because F is a bundle of many sub functions, every sub function has convergence errors generally. Thereof, it is necessary to introduce a more practical function, here, average cost function (ACF, i.e., $\frac{F}{(180/\Delta\theta)}$) to index algorithm convergency. The optimal value of the cost function, which is finally obtained by the optimizing method. In this paper, an algorithm based on hybrid trust region method is propped in next section, however, for applying the approach, we must first compute its gradient and Hessian matrices (see the details in Appendix A).

3. ALGORITHM BASED ON THE CONIC TRUST REGION ALGORITHM

We consider the following unconstrained optimization problem:

$$\min f(x),\quad x\in R^n \qquad (6)$$

where $f : R^n \to R$ is a continuously differentiable function. The optimization problem has become an important research focus given its wide range of potential applications. Throughout the paper, $\{x_k\}$ is a sequence of points generated by our algorithm. CTRA iteratively solves optimization problems. At each iteration, a trial step d_k is generated by solving the sub-problem

$$\begin{cases} \min \varphi_k(d) = f(x_k) + \dfrac{g_k^T d}{1-\alpha_k^T d} + \dfrac{1}{2}\dfrac{d^T B_k d}{\left(1-\alpha_k^T d\right)^2}, & \alpha_k, d \in R^n \\ s.t.\ \|d\| \leqslant \Delta_k, & 1-\alpha_k^T d > 0 \end{cases} \tag{7}$$

where $g_k = \nabla f(x_k) \in R^n$, $B_k \in R^{n\times n}$ is an approximate Hessian matrix of $f(x_k)$, and $\Delta_k > 0$ is a trust region radius. Some criteria are used to decide on accepting a trial step and the manner by which the trust region radius is adjusted. After obtaining a trial step d_k, the trust region algorithms compute the ratio $\rho_k = \frac{f(x_k)-f(x_k+d_k)}{f(x_k)-\varphi_k(d_k)}$. Where ρ_k indicates the approximate degree of $f(x_k + d_k)$ and $\varphi_k(d_k)$, and $\rho_k \to 1$ while $\Delta_k \to 0$. In iterative computations, $\rho_k \geqslant c_0$ (a positive constant) means that $\varphi_k(d_k)$ is approximate to $f(x_k+d_k)$, and trial step d_k is accepted, so that Δ_k is enlarged to speed up the computations. On the other hand, if d_k is not accepted ($\rho_k < c_0$), some methods resolve subproblem (Eq. (7)) by reducing the trust region radius until an acceptable step is found. Therefore, the subproblem may be solved several times during each iteration before an acceptable step is found, and these repetitious processes increase the total computational cost for large-scale problems. In the current paper, we propose a new algorithm to improve the computational efficiency of trust region methods.

In the computation, B_k is generated using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula and is therefore is positive; $d_k = -\beta_k g_k$ ($\beta_k \geqslant 0$) is a descent direction of $f(x_k)$. Meanwhile, $f(x_k+d_k) \approx \varphi_k(d_k)$ when $d_k \in \{\|x - x_k\| \leqslant \Delta_k\}$, indicating that $d_k = -\beta_k g_k$ is also the descent direction of $\varphi_k(d)$ under this condition. Thus, we obtain an approximate solution of the subproblem by solving the following inequality:

$$-\frac{g_k^T(-\beta_k g_k)}{1-\alpha_k^T(-\beta_k g_k)} - \frac{1}{2}\frac{(-\beta_k g_k)^T B_k(-\beta_k g_k)}{\left(1-\alpha_k^T(-\beta_k g_k)\right)^2} \leqslant \frac{f(x_k)-f(x_k+(-\beta_k g_k))}{c_0} \tag{8}$$

Let $\beta^* = \max\{\beta_k\,|(8)\ holds\}$, then $d_k = -\beta^* g_k$ is an approximate solution of the subproblem and $\Delta_k = \|\beta^* g_k\|$ is the trust region radius of the kth iteration.

Algorithm 3.1 (conic trust region algorithm for unconstrained optimization)

Step 0. Given $0.75 < c_0 < 1$, $c_1 > 1$ and an initial symmetric positive definite matrix B_0, choose x_0 and set $k = 0$.

Step 1. Compute g_k; if $\|g_k\| < \varepsilon$ (ε is a small positive real number), then stop.

Step 2. calculating solution d_k of sub-problem (Eq. (7)).

Step 3. We compute ρ_k. If $\rho_k \geqslant \mu$, then we proceed to step 4. Otherwise, Solving inequality (8), we obtain $d_k = -\beta^* g_k$ and $x_{k+1} = x_k + d_k$. Let

$$\Delta_{k+1} = \rho_k \Delta_k$$

We then proceed to step 5.
Step 4. We set

$$x_{k+1} = x_k + d_k$$

and

$$\begin{aligned} &\Delta_{k+1} = \Delta_k, \quad \text{if } \|d_k\| < \Delta_k \\ &\Delta_{k+1} \in [\Delta_k, c_1\Delta_k], \quad \text{if } \|d_k\| = \Delta_k \end{aligned}$$

Step 5. Generate α_{k+1}, modify B_k into B_{k+1} using the BFGS formula as an approximation to $\nabla^2 f(x_{k+1})$, then return to step 1.
Remarks. The method for generating α_{k+1} and B_{k+1} was as previously reported [16–18]. We assume that the matrices B_{k+1} are uniformly bounded to prove the convergence, and $\forall k$, $\exists \sigma \in (0, 1)$: $\|\alpha_k\|\Delta_k \leqslant \sigma$ which ensures that the conic model function $\varphi_k(d)$ is bounded over the trust-region ($\|d\| \leqslant \Delta_k$). We reiterate that our algorithm is reduced to a quadratic model-based algorithm if $\alpha_k = 0$ for all k. Under the smoothness assumptions made in this paper, note that the objective function is convex quadratic around a local minimizer. Therefore, choosing $\alpha_k \simeq 0$ asymptotically is suitable when x_k is near the minimizer.

The mild following assumptions, commonly used in the convergence analysis of most optimization algorithms [16–18], are proposed to analyze the convergence properties of Algorithm 3.1.

Assumption 1. f is twice continuously differentiable and bounded below.

Assumption 2. A strong local minimizer point x^* exists, such that $\nabla^2 f(x^*)$ is positive definite.
Set $s = \frac{d}{1-\alpha_k^T d}$, $\therefore$ $d = \frac{s}{1+\alpha_k^T s}$. From $\|d\| \leqslant \Delta_k$, we obtain $\|\frac{s}{1+\alpha_k^T s}\| \leqslant \Delta_k$, $\|s\| \leqslant \Delta_k|1 + \alpha_k^T s| \leqslant \Delta_k + \Delta_k\|\alpha_k\|\|s\| \leqslant \Delta_k + \sigma\|s\|$, and $\|s\| \leqslant \frac{\Delta_k}{1-\sigma}$. We consider a sequence $\{s_k = -\tau_k \frac{\Delta_k}{1-\sigma} \frac{g_k}{\|g_k\|}\}$, where $\tau_k = \begin{cases} 1, & \text{if } g_k^T B g_k \leqslant 0 \\ \min\left\{\frac{\Delta_k}{1-\sigma}\frac{\|g_k\|^3}{g_k^T B g_k}, 1\right\}, & \text{otherwise} \end{cases}$. Clearly, $\|s_k\| \leqslant \frac{\Delta_k}{1-\sigma}$, $s_k \in B(x_k, \frac{\Delta_k}{1-\sigma})$.

Lemma 1. In equation $f(x_k) - \varphi_k(d) = -\frac{g_k^T d}{1-\alpha_k^T d} - \frac{1}{2}\frac{d^T B_k d}{(1-\alpha_k^T d)^2} = -g_k^T s - \frac{1}{2} s^T B s \geqslant \frac{1}{2}\|g_k\| \min\{\frac{\Delta_k}{1-\sigma}, \frac{\|g_k\|}{\|B\|}\}$ holds.

Proof:
Case 1: $g_k^T B g_k \leqslant 0$ means that $\tau_k = 1$.
$\therefore -g_k^T s_k - \frac{1}{2} s_k^T B s_k = \frac{\|g_k^T\|^2}{\|g_k\|} \frac{\Delta_k}{1-\sigma} - \frac{1}{2}(\frac{\Delta_k}{1-\sigma})^2 \frac{1}{\|g_k^T\|^2} g_k^T B g_k \geqslant \frac{\Delta_k}{1-\sigma}\|g_k\| \geqslant \frac{1}{2}\|g_k\| \min\{\frac{\Delta_k}{1-\sigma}, \frac{\|g_k\|}{\|B\|}\}$.
Case 2: $g_k^T B g_k > 0$.
If $\min\{\frac{\Delta_k}{1-\sigma}\frac{\|g_k\|^3}{g_k^T B g_k}, 1\} = \frac{\Delta_k}{1-\sigma}\frac{\|g_k\|^3}{g_k^T B g_k}$, then $s_k = -\frac{\|g_k^T\|^2}{g_k^T B g_k} g_k$.
$\therefore -g_k^T s_k - \frac{1}{2} s_k^T B s_k = \frac{\|g_k\|^4}{g_k^T B g_k} - \frac{1}{2}\frac{\|g_k\|^4}{(g_k^T B g_k)^2}(g_k^T B g_k) = \frac{1}{2}\frac{\|g_k\|^4}{g_k^T B g_k} \geqslant \frac{1}{2}\frac{\|g_k\|^2}{\|B\|} \geqslant \frac{1}{2}\|g_k\| \min\{\frac{\Delta_k}{1-\sigma}, \frac{\|g_k\|}{\|B\|}\}$.
If $\min\{\frac{\Delta_k}{1-\sigma}\frac{\|g_k\|^3}{g_k^T B g_k}, 1\} = 1$, then $\frac{\Delta_k}{1-\sigma}\frac{\|g_k\|^3}{g_k^T B g_k} \geqslant 1$, $g_k^T B g_k \leqslant \frac{1-\sigma}{\Delta_k}\|g_k\|^3$ and $s_k = -\frac{\|g_k^T\|^2}{g_k^T B g_k} g_k$.
Thereof,
$-g_k^T s_k - \frac{1}{2} s_k^T B s_k = \frac{\Delta_k}{1-\sigma}\frac{\|g_k\|^2}{\|g_k\|} - \frac{1}{2}(\frac{\Delta_k}{1-\sigma})^2 \frac{g_k^T B g_k}{\|g_k\|^2} \geqslant \frac{\Delta_k}{1-\sigma}\|g_k\| - \frac{1}{2}(\frac{\Delta_k}{1-\sigma})^2 \frac{1}{\|g_k\|^2}\frac{1-\sigma}{\Delta_k}\|g_k\|^3 \geqslant \frac{1}{2}(\frac{\Delta_k}{1-\sigma})\|g_k\| \geqslant \frac{1}{2}\|g_k\| \min\{\frac{\Delta_k}{1-\sigma}, \frac{\|g_k\|}{\|B\|}\}$.

Lemma 2. *Suppose d_* is a solution of Sub-problem (7); then the following inequality holds:*

$$f(x_k) - \varphi_k(d_*) \geqslant \frac{1}{2}\|g_k\| \min\left\{\frac{\Delta_k}{1-\sigma}, \frac{\|g_k\|}{\|B\|}\right\}$$

Proof. $\because$ $\varphi_k(d) \geqslant \varphi_k(d_*)$ and Lemma 1, thus, the above inequality holds.

Lemma 3. *Suppose that Assumption 2 holds, then M exists as a positive constant, such that*

$$|f(x_k) - f(x_k + d_k) - (f(x_k) - \varphi_k(d_k))| \leqslant M\|d_k\|^2, \quad \forall k$$

Proof. This proof is similar to that of Lemma 3.2 [17].

Lemma 4. *Suppose that Assumption 1 holds, then Algorithm 3.1 is well defined.*

Proof. Assuming that the algorithm does not terminate at $x_k(g_k \neq 0)$, then from Lemma 2 and 3, we obtain

$$\begin{aligned}\left|\frac{f(x_k) - f(x_k + d_k) - (f(x_k) - \varphi_k(d_k))}{f(x_k) - \varphi_k(d_k)}\right| &\leqslant \frac{2M\,\|d_k\|^2}{\|g_k\| \min\left\{\frac{\Delta_k}{1-\sigma}, \frac{\|g_k\|}{\|B\|}\right\}} \\ &\leqslant \frac{2M\,\|\Delta_k\|^2}{\|g_k\| \min\left\{\frac{\Delta_k}{1-\sigma}, \frac{\|g_k\|}{\|B\|}\right\}}\end{aligned}$$

The definition of Δ_k, $\Delta_k \to 0$ as $\|\beta^*\| \to 0$ implies that

$$\left|\frac{f(x_k)-f(x_k+d_k)-(f(x_k)-\varphi_k(d_k))}{f(x_k)-\varphi_k(d_k)}\right| \to 0, \text{ i.e., } \frac{f(x_k)-f(x_k+d_k)}{f(x_k)-\varphi_k(d_k)} \to 1$$

From the definition of ρ_k, we obtain

$$\rho_k = \frac{f(x_k)-f(x_k+d_k)}{f(x_k)-\varphi_k(d_k)} \to 1 > c_0$$

Therefore, we can obtain trial step d_k after finite computation, i.e., Algorithm 3.1 is well defined.

Lemma 5. *Suppose that Assumption 1 and 2 hold, and is the sequence generated by Algorithm 3.1, then*

$$\lim_{k\to\infty} \|g_k\| = 0$$

Proof. This proof is similar to that of Lemma 2 [7].

Lemma 6. *The sequence $\{x_k\}$ is generated by Algorithm 3.1 and converging to x^*. Suppose that $\lim\limits_{k\to\infty} \frac{\|(B_k-\nabla^2 f(x^*))d_k\|}{\|d_k\|} = 0$, where $\nabla^2 f(x^*)$ is positive definite. Therefore, the convergence rate is superlinear. Moreover, suppose that $\nabla^2 f(x)$ is Lipschitz-continuous in a neighborhood $N(x^*,\delta) = \left\{x \in R^N \,|\|x-x^*\| < \delta\right\}$, i.e., $L(\delta) > 0$ exists such that*

$$\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| \leqslant L(\delta)\,\|x-y\|\,, \quad \forall x, y \in N(x^*,\delta)$$

$\{x_k\}$ then quadratically converges to x^*.

Proof. Given that

$$\lim_{k\to\infty} \frac{\left\|(B_k-\nabla^2 f(x^*))d_k\right\|}{\|d_k\|} = 0$$

therefore,

$$\left\|(B_k - \nabla^2 f(x^*))d_k\right\| = o(\|d_k\|) \tag{9}$$

Using Eq. (9) and the Taylor theorem, we obtain

$$\begin{aligned}
&|f(x_k) - f(x_k+d_k) - (f(x_k) - \varphi_k(d_k))| \\
&= \left|-g_k^T d_k - \frac{1}{2} d_k^T \nabla^2 f_k d_k - o\left(\|d_k\|^2\right)\right) + \left(\frac{g_k^T d}{1-\alpha_k^T d} + \frac{1}{2}\frac{d^T B_k d}{\left(1-\alpha_k^T d\right)^2}\right)\right| \\
&(\because \alpha_k \to 0 \;\; as \;\; x_k \to x^*) \\
&\leqslant \left|\frac{1}{2} d_k^T \left(B_k - \nabla^2 f\right) d_k\right| + o\left(\|d_k\|^2\right) \leqslant \frac{1}{2}\left\|d_k^T\right\| \left\|\left(B_k - \nabla^2 f\right) d_k\right\| + o\left(\|d_k\|^2\right) \\
&\leqslant \frac{1}{2}\left\|d_k^T\right\| o(\|d_k\|) + o\left(\|d_k\|^2\right) = o\left(\|d_k\|^2\right)
\end{aligned}$$

Thus,

$$\left|\frac{f(x_k)-f(x_k+d_k)}{f(x_k)-\varphi_k(d_k)}-1\right| = \left|\frac{f(x_k)-f(x_k+d_k)-(f(x_k)-\varphi_k(d_k))}{f(x_k)-\varphi_k(d_k)}\right|$$

$$\leqslant \frac{o(\|d_k\|^2)}{\left|\frac{g_k^T d}{1-\alpha_k^T d}+\frac{1}{2}\frac{d^T B_k d}{\left(1-\alpha_k^T d\right)^2}\right|} \leqslant \frac{o(\|d_k\|^2)}{\left|\frac{1}{2}d_k^T B_k d_k\right|}(k\to\infty, \|g_k\|\to 0, \alpha_k\to 0)$$

$$\leqslant \frac{o(\|d_k\|^2)}{\|d_k\|^2}\to 0 \text{ as } k\to\infty.$$

Therefore,

$$\tfrac{f(x_k)-f(x_k+d_k)}{f(x_k)-\varphi_k(d_k)} \geqslant \mu$$

$x_{k+1} = x_k + d_k$ holds for all sufficiently large k, and Algorithm 3.1 becomes the Newton method or quasi-Newton method that superlinearly converges. Therefore, the sequence $\{x_k\}$ converges to x^* superlinearly. $\because x_k \to x^*$, $\therefore \exists \tilde{k} > 0, \forall k > \tilde{k}, x_k \in N(x^*, \delta)$. Set $\delta_k = x_k - x^*$. Using the mean value theorem, we obtain

$$\begin{aligned}
\delta_{k+1} &= x_{k+1} - x^* = x_k - x^* + d_k = \delta_k - \beta_k g_k \\
&= \delta_k - \beta_k(g_k - g^*) - \beta_k g^* \quad (g^* = g(x^*)) \\
&= \beta_k\left[\nabla^2 f(x_k)\delta_k - \int_0^1 \nabla^2 f(x^*+t\delta_k)\delta_k dt\right] - \beta_k g^* \\
&= \beta_k\left[\int_0^1 \left[\nabla^2 f(x_k) - \nabla^2 f(x^*+t\delta_k)\right]\delta_k dt\right] - \beta_k g^*
\end{aligned}$$

Thus,

$$\begin{aligned}
\|\delta_{k+1}\| &= \left\|\beta_k\left[\int_0^1 \left[\nabla^2 f(x_k) - \nabla^2 f(x^*+t\delta_k)\right]\delta_k dt\right] - \beta_k g^*\right\| \\
&\leqslant \|\beta_k\| \int_0^1 \left\|\nabla^2 f(x_k) - \nabla^2 f(x^*+t\delta_k)\right\| \|\delta_k\| dt + \|\beta_k g^*\| \\
&\leqslant \|\beta_k\| L(\delta) \|\delta_k\|^2 \int_0^1 t dt + \left\|\nabla^2 f(x_k)^{-1}\right\| \|g^*\|
\end{aligned}$$

$$\therefore \lim_{k\to\infty}\frac{\|\delta_{k+1}\|}{\|\delta_k\|^2} \leqslant \frac{1}{2}L(\delta)\|\beta_k\| \quad \left(\lim_{k\to\infty}\|g(x_k)\| = \|g^*\| = 0\right)$$

implying that $\{x_k\}$ converges to x^* quadratically. Finally, the general algorithm is derived by minimizing the cost function, obtaining optimal weight vector W_o, and outputing amplitude pattern $|p(W_o, \theta)|$. The algorithm is described as follows:

Step 1. The cost function $F(W, \mathrm{A})$ is constructed as Eq. (3).

Step 2. Let $X = (W_1 \ W_2 \ \mathrm{A})$ then, computing $\nabla_X F$ and $\nabla_X^2 F$

Eqs. (A21) and (A43) respectively (see the details in Appendix A). The two formulas are essential to optimizing algorithm. Finally, Algorithm 3.1 is used to minimize the cost function $F(X)$.

Step 3. Base on the previous analysis, vector W converges a optimal solution, if $\frac{F}{(180/\Delta\theta)} \rightarrow 0$, then array amplitude pattern is outputed.

4. SIMULATION RESULTS

In this section, several simulations are performed to test the multi-beam synthesizing patterns validity.

4.1. Simulation 1: Comparison with NACTRM [16]

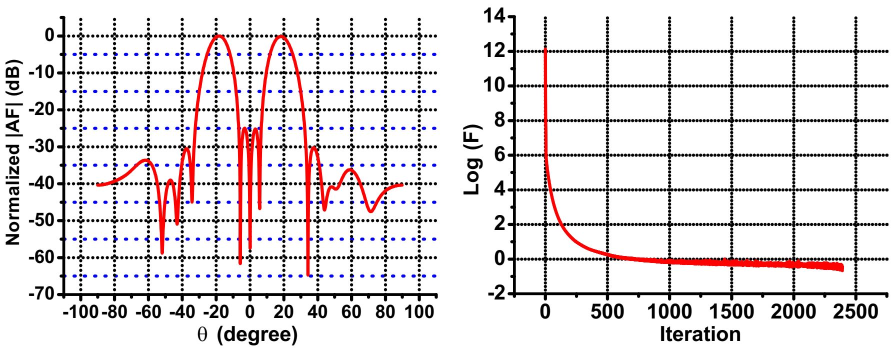
Consider a synthesis problem using a 12-element half-wave-length spacing linear array with the following configurations: $\theta_1 = -26.0$, $\theta_2 = -24.5$, $\theta_3 = -15.5$, $\theta_4 = -14.0$, $\theta_5 = 14.0$, $\theta_6 = 15.5$, $\theta_7 = 24.5$

Table 1. Comparison of the SICTRM and NACTRM algorithms.

No.	Algorithm	r	m	$\Delta\theta$	ACF	DPMLL (dB)	PSLL (dB)	Time (s)
1	**SICTRM**	**23.0**	**0.5**	**0.2**	**0.3028**	**0.1845**	**−20.23**	**23.35**
	NACTRM	23.0	0.5	0.2	0.9219	0.03314	−15.16	50.43
2	**SICTRM**	**25.0**	**0.5**	**0.2**	**0.2687**	**0.2188**	**−19.33**	**20.65**
	NACTRM	25.0	0.5	0.2	0.8534	0.2127	−18.16	51.24
3	**SICTRM**	**25.0**	**0.5**	**0.3**	**0.2334**	**0.1741**	**−19.58**	**21.06**
	NACTRM	25.0	0.5	0.3	0.7644	0.0189	−19.37	45.61
4	**SICTRM**	**20.0**	**0.4**	**0.3**	**0.4261**	**0.1341**	**−21.86**	**23.78**
	NACTRM	20.0	0.4	0.3	0.2254	0.4850	−20.04	47.53
5	**SICTRM**	**20.0**	**0.4**	**0.3**	**0.3404**	**0.2610**	**−22.40**	**23.56**
	NACTRM	20.0	0.4	0.3	0.6873	0.0293	−16.46	49.57
6	**SICTRM**	**22.0**	**0.5**	**0.3**	**0.3252**	**0.2955**	**−22.58**	**23.68**
	NACTRM	22.0	0.5	0.3	0.2818	0.3398	−19.82	44.72
7	**SICTRM**	**20.0**	**0.4**	**0.1**	**0.4477**	**0.2155**	**−21.37**	**32.02**
	NACTRM	20.0	0.4	0.1	0.3815	0.2191	−18.71	58.12
8	**SICTRM**	**22.0**	**0.3**	**0.1**	**0.7938**	**0.0599**	**−23.35**	**25.33**
	NACTRM	22.0	0.3	0.1	0.8815	0.1273	−18.71	50.32
9	**SICTRM**	**22.0**	**0.3**	**0.1**	**0.7128**	**0.0966**	**−24.32**	**23.31**
	NACTRM	22.0	0.3	0.1	0.7923	0.2247	−20.68	49.84
10	**SICTRM**	**23.0**	**0.2**	**0.1**	**0.5263**	**0.1236**	**−24.97**	**23.97**
	NACTRM	23.0	0.2	0.1	0.7128	0.0966	−24.19	54.64

and $\theta_8 = 26.0$. The simulations based on the SICTRM and NACTRM algorithms were executed 10 times independently, the results are listed in Table 1.

The two iterative algorithms optimize same cost functions in every stimulation, but the results show their distinct performance. Under NACTRM, average wasted time, average DPMLL (the difference of peak mainlobe level) and average PSLL (the peak sidelobe level) are 50.20 second, 0.1786 dB and −19.13 dB, under SICTRM, they are 24.07 second, 0.1763 dB and −21.99 dB respectively. Undoubtedly, compared with NACTRM, the SICTRM algorithm exhibits stronger



(a) (b)

Figure 2. (a) Radiation pattern and (b) cost function variety of 12-element line antenna array under SICTRM algorithm.

Table 2. Optimal solution of 12-element line antenna array under SICTRM algorithm.

Element	W_1	W_2	Element	W_1	W_2	Element	W_1	W_2
1	−0.019499	0.007891	**2**	−0.039725	0.020580	**3**	−0.021394	0.011094
4	0.043818	−0.020758	**5**	0.086723	−0.043317	**6**	0.045457	−0.022992
7	−0.042839	0.020464	**8**	−0.087299	0.043599	**9**	−0.044590	0.022598
10	0.019820	−0.009886	**11**	0.041093	−0.020637	**12**	0.019018	−0.008572
$\alpha_{1,2}$	0.009870	0.008293	$\alpha_{3,4}$	0.007958	0.007903	$\alpha_{5,6}$	0.008166	0.009430

search ability in these simulations. The reported minimum PSLL for synthesis of 12-element multi-beam array is −24.76 dB in [11], our results is better than their result. We show the optimal normalized AF amplitude pattern, ACF variety and optimal solution in Figures 2(a), (b) and Table 2 respectively.

4.2. Simulation 2: Comparison with DA Proposed in [7] and Dynamic Differential Evolution (DDE) [13]

The DA method, a gradient-based optimizing algorithm, adopts a computing inequality to define step length for improving computation efficiency, the method has exhibited strong convergence properties. DDE is a new version of differential evolution algorithm, which used only one array to search the optimization. DDE significantly outperforms the differential evolution strategy in efficiency, robustness, and memory requirement [21].

Consider a synthesis problem using a 17-element half-wave-length spacing linear array with the following configurations: $\theta_1 = -24.0$, $\theta_2 = -23.0$, $\theta_3 = -17.0$, $\theta_4 = -16.0$, $\theta_5 = 26.0$, $\theta_6 = 27.0$, $\theta_7 = 33.0$ and $\theta_8 = 34.0$. Executing the three algorithm 10 times independently and listing results in Table 3 as following.

Under DA, average iterative time, minimum PSLL, average DPMLL and PSLL are 36.07 second, −28.75 dB, 0.3471 dB, and −22.98 dB, respectively. Similarly, under DDE, they are 1818.34 second, 26.71 dB, 0.3421 dB, and −23.96 dB, respectively. However, under SICTRM, they are 25.79 second, −30.15 dB, 0.2347 dB, and −25.88 dB, respectively. Obviously, the result indicates that the proposed algorithm has stronger ability to suppress sidelobe level. Finally, we show the optimal radiation pattern, ACF variety and optimal solution in Figures 3(a), (b), and Table 4, respectively.

4.3. Simulation 3: SICTRM Performance with 17 Mutual Coupling Elements

In this simulation, we test the SICTRM performance in the presence of antenna elements coupled to each other. Taking the mutual coupling into account, the SICTRM is closer to the actual situation. The coupling coefficient Between any two antenna elements can be calculated [22]. In the case of an uniform linear array, these coefficients can be represented by a symmetric Toeplitz matrix [23, 24]. The mutual coupling between two elements is inversely related to their distance, and thus it is assumed that when the distance between two array elements is more than k inter-element spacing, the mutual

Table 3. Comparison of the SICTRM, DA and DDE algorithms.

No.	Algorithm	r	m	$\Delta\theta$	ACF	DPMLL (dB)	PSLL (dB)	Time (s)
1	**SICTRM**	**22.0**	**0.3**	**0.1**	**0.6952**	**0.1517**	**−25.51**	**28.31**
	DA				0.6822	0.3641	−19.30	37.90
	DDE				0.7138	0.2143	−21.65	1537.82
2	**SICTRM**	**22.0**	**0.4**	**0.1**	**0.3952**	**0.2257**	**−24.10**	**22.43**
	DA				0.6865	0.3624	−23.47	32.17
	DDE				0.6457	0.3574	−23.74	1645.46
3	**SICTRM**	**22.0**	**0.4**	**0.4**	**0.7492**	**0.1919**	**−22.87**	**30.58**
	DA				0.7674	0.1616	−20.55	42.92
	DDE				0.8614	0.4827	−21.44	1785.78
4	**SICTRM**	**22.0**	**0.4**	**0.6**	**0.8206**	**0.1264**	**−21.50**	**23.29**
	DA				0.8640	0.1058	−18.44	34.48
	DDE				0.7589	0.2414	−22.37	1814.24
5	**SICTRM**	**22.0**	**0.2**	**0.1**	**0.6975**	**0.3874**	**−28.77**	**22.85**
	DA				0.7648	0.3513	−28.75	30.87
	DDE				0.5894	0.4267	−25.67	1874.67
6	**SICTRM**	**22.0**	**0.2**	**0.1**	**0.7333**	**0.2103**	**−25.31**	**32.21**
	DA				0.7449	0.5621	−27.47	38.24
	DDE				0.8457	0.5248	−26.71	1977.24
7	**SICTRM**	**24.0**	**0.1**	**0.1**	**0.7958**	**0.2118**	**−23.91**	**23.05**
	DA				0.8653	0.3822	−21.22	30.37
	DDE				0.6981	0.3140	−22.89	1867.44
8	**SICTRM**	**22.0**	**0.1**	**0.1**	**0.7946**	**0.2367**	**−28.32**	**25.35**
	DA				0.7512	0.3641	−21.74	38.24
	DDE				0.8017	0.2078	−24.75	1878.64
9	**SICTRM**	**23.0**	**0.2**	**0.1**	**0.6881**	**0.3241**	**−28.38**	**22.23**
	DA				0.6587	0.3781	−22.61	40.35
	DDE				0.7149	0.4108	−24.11	1904.67
10	**SICTRM**	**22.0**	**0.3**	**0.1**	**0.7252**	**0.2814**	**−30.15**	**27.65**
	DA				0.6004	0.4393	−26.24	35.24
	DDE				0.7089	0.2413	−26.33	1897.23

coupling coefficients are assumed to be zero. Therefor, the mutual coupling matrix can be sufficiently modeled as follows:

$$M = Topelitz\left(1,\ c_1,\ c_2,\ \ldots,\ c_{k-1},\ \mathbf{0}^{1\times(N-k)}\right) \tag{10}$$

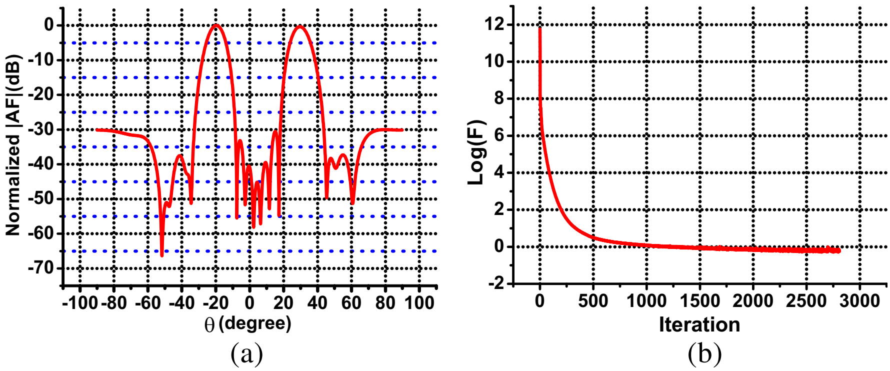


Figure 3. (a) Radiation pattern and (b) cost function variety of 17-element line antenna array under SICTRM.

Table 4. Optimal solution of 17-element line antenna array under SICTRM.

Element	W_1	W_2	Element	W_1	W_2	Element	W_1	W_2
1	−0.001240	0.000190	**2**	−0.025837	0.013814	**3**	−0.026250	0.006116
4	0.059069	0.001514	**5**	0.102814	0.027050	**6**	−0.021201	−0.015251
7	−0.135771	−0.127609	**8**	−0.038699	−0.055074	**9**	0.050524	0.174523
10	0.019190	0.165329	**11**	0.018093	−0.092873	**12**	0.076608	−0.162537
13	0.001669	−0.012582	**14**	−0.088523	0.071074	**15**	−0.046368	0.018256
16	0.035043	−0.004278	**17**	0.025260	0.001108			
$\alpha_{1,2}$	0.011523	0.012739	$\alpha_{3,4}$	0.009282	0.009612	$\alpha_{5,6}$	0.011771	0.009866

Under this model, the true steering vector should be rewritten as

$$S_t = MS(\theta) \tag{11}$$

Make minor modifications to the original program accordingly and execute 10 times. The results are shown in Table 5. In the simulation, the mutual coupling matrix

Table 5. SICTRM performance with 17 mutual coupling elements.

No.	Algorithm	r	m	$\Delta\theta$	ACF	DPMLL (dB)	PSLL (dB)	Time (s)
1	**SICTRM**	**22.0**	**0.3**	**0.1**	**0.8641**	**0.1231**	**−25.47**	**19.14**
2	**SICTRM**	**22.0**	**0.4**	**0.1**	**0.7080**	**0.2823**	**−24.04**	**22.46**
3	**SICTRM**	**22.0**	**0.4**	**0.4**	**0.7995**	**0.3248**	**−19.13**	**29.48**
4	**SICTRM**	**22.0**	**0.4**	**0.6**	**0.8633**	**0.3561**	**−20.86**	**26.34**
5	**SICTRM**	**22.0**	**0.2**	**0.1**	**0.8813**	**0.3119**	**−23.57**	**32.16**
6	**SICTRM**	**22.0**	**0.2**	**0.1**	**0.8233**	**0.1766**	**−20.84**	**35.41**
7	**SICTRM**	**24.0**	**0.1**	**0.1**	**0.6569**	**0.2319**	**−19.60**	**25.61**
8	**SICTRM**	**22.0**	**0.1**	**0.1**	**0.8361**	**0.3594**	**−19.98**	**29.44**
9	**SICTRM**	**23.0**	**0.2**	**0.1**	**0.7384**	**0.3549**	**−19.95**	**27.42**
10	**SICTRM**	**22.0**	**0.3**	**0.1**	**0.7742**	**0.2330**	**−22.34**	**32.17**

$$M = Topelitz\left(1,\ 0.2+0.25j,\ 0.1+0.2j,\ 0.09+0.1j,\ 0^{1\times 13}\right)$$

Other parameters are same as those of simulation 2.

Under SICTRM, average iterative time, minimum PSLL, average DPMLL and PSLL are 27.96 second, −25.47 dB, 0.2754 dB and −21.57 dB, respectively. The results show that our algorithm is valid even in the presence of larger array element coupling. Finally, we show the comparing radiation patterns in isolated and coupled cases, ACF variety and optimal solution in Figures 4(a), (b) and Table 6 respectively. These two patterns are well matched.

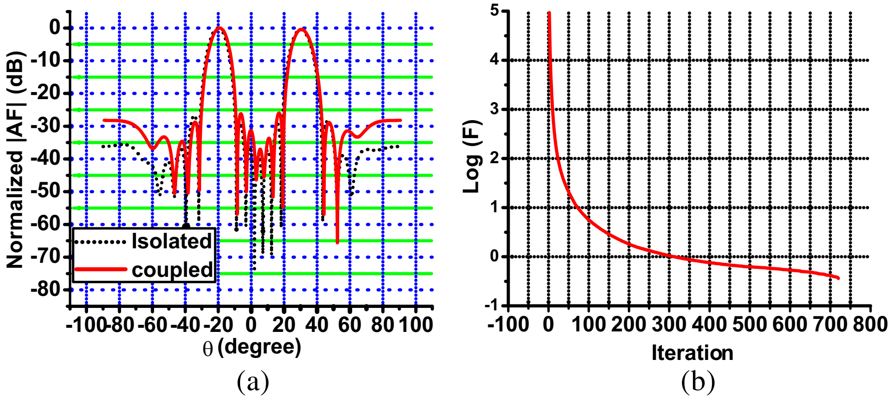


Figure 4. (a) Comparing radiation patterns and (b) cost function variety of 17-element line antenna array under SICTRM.

Table 6. Optimal solution of 17-coupled-element line antenna array under SICTRM.

Element	W_1	W_2	Element	W_1
1	0.001015	0.001615	**2**	−0.027010
4	0.042089	0.013591	**5**	0.091644
7	−0.082260	−0.134630	**8**	−0.026376
10	−0.021683	0.154256	**11**	0.025531
13	0.008818	−0.015877	**14**	−0.100060
16	0.045868	0.004018	**17**	0.033354
$\alpha_{1,2}$	0.011728	0.019533	$\alpha_{3,4}$	0.017368
W_2	Element	W_1	W_2	
0.001287	**3**	−0.040886	−0.003572	
0.056793	**6**	−0.000699	−0.002109	
−0.076570	**9**	0.006163	0.141027	
−0.057479	**12**	0.106361	−0.133212	
0.048884	**15**	−0.060390	0.013171	
0.008770				
0.018190	$\alpha_{5,6}$	0.020401	0.010581	

5. DISCUSSION

We have designed three simulations to validate our algorithm. Simulation results indicate that the proposed algorithm is valid to multi-beam antenna array pattern synthesis in both isolated and coupled cases. Of course, the performance of the algorithms relates to some pre-assumed parameters. In these parameters, the parameter r has impact on the convergence of the algorithm greatly. If r is too small ($r < 20$), antenna sidelobe level will not be suppressed effectively. However, if r is too is too large, the algorithm is difficult to converge. The simulations results show that our assumptions ($20 \leq r \leq 24$) are reasonable. Generally, parameter $m > 0$. In the three simulations, m is more than 0.1 and less than 0.4. Similarly, if it is too big, the algorithm is difficult to converge. In theory, when the angle resolution $\Delta\theta$ is smaller, the antenna sidelobe level will be suppressed more effectively. But that will increase the computational burden. In fact, $\Delta\theta = 0.1$ is a good choice. The parameter ACF is a convergence indicator and is assumed less than 1.

6. CONCLUSION

In this paper, we have presented a multi-beam model for antenna array pattern synthesis problem. Under the model, an antenna is able to create multibeams simultaneously. In order to quickly calculate the objective function to obtain the optimal solution, an iterative optimization algorithm named SICTRM is proposed. The SICTRM algorithm, unlike traditional trust region methods that resolve subproblems, can improve computational efficiency as indicated by the theoretical analysis and stimulation results. Computer simulations illustrate the general algorithm's good performance as it is applied to the AAPS problem.

APPENDIX A. THE DERIVATION OF GRADIENT AND HESSIAN MATRIX OF COST FUNCTION

The similar derivation can be seen in [7]. Let $t_1 = S_1^T W_1$, $t_2 = S_2^T W_2$, $t_3 = S_2^T W_1$, and $t_4 = S_1^T W_2$, we obtain

$$\nabla_{W_1} P = 2\left((t_1 + t_2)S_1 + (t_3 + t_4)S_2\right) \tag{A1}$$

For convenience, let $F_{11} = (\alpha_1 - P)^2 + \left(\frac{P}{\alpha_1}\right)^m$, $F_{12} = (\alpha_2 - P)^2 + \left(\frac{P}{\alpha_2}\right)^m$, $F_{13} = (f_1 - P)^2 + \left(\frac{P}{f_1}\right)^m$, $F_{14} = (\beta - P)^2$, $F_{15} = (f_2 - P)^2 + \left(\frac{P}{f_2}\right)^m$, and $F_{16} = (\alpha_3 - P)^2 + \left(\frac{P}{\alpha_3}\right)^m$. Thereof,

$$\nabla_{W_1} F_{11} = 2\left(P - \alpha_1\right)\nabla_{W_1} P + \frac{m}{\alpha_1}\left(\frac{P}{\alpha_1}\right)^{m-1}\nabla_{W_1} P \tag{A2}$$

Similarly,

$$\nabla_{W_1} F_{12} = 2\left(P - \alpha_2\right)\nabla_{W_1} P + \frac{m}{\alpha_2}\left(\frac{P}{\alpha_2}\right)^{m-1}\nabla_{W_1} P \tag{A3}$$

$$\nabla_{W_1} F_{13} = 2\left(P - f_1\right)\nabla_{W_1} P + \frac{m}{f_1}\left(\frac{P}{f_1}\right)^{m-1}\nabla_{W_1} P \tag{A4}$$

$$\nabla_{W_1} F_{14} = 2\left(P - \beta\right)\nabla_{W_1} P \tag{A5}$$

$$\nabla_{W_1} F_{15} = 2\left(P - f_2\right)\nabla_{W_1} P + \frac{m}{f_2}\left(\frac{P}{f_2}\right)^{m-1}\nabla_{W_1} P \tag{A6}$$

$$\nabla_{W_1} F_{16} = 2\left(P - \alpha_3\right)\nabla_{W_1} P + \frac{m}{\alpha_3}\left(\frac{P}{\alpha_3}\right)^{m-1}\nabla_{W_1} P \tag{A7}$$

From Eqs. (A1)–(A7), we obtain,

$$\nabla_{W_1} F_1(W, \mathrm{A}) = \sum_1 \nabla_{W_1} F_{11} + \sum_2 \nabla_{W_1} F_{12} + \sum_3 \nabla_{W_1} F_{13} + \sum_4 \nabla_{W_1} F_{14} + \sum_5 \nabla_{W_1} F_{15} + \sum_6 \nabla_{W_1} F_{16} \quad \text{(A8)}$$

Similarly,

$$\nabla_{W_1} F_2(W, \mathrm{A}) = \sum_7 \nabla_{W_1} F_{21} + \sum_8 \nabla_{W_1} F_{22} + \sum_9 \nabla_{W_1} F_{23} + \sum_{10} \nabla_{W_1} F_{24} + \sum_{11} \nabla_{W_1} F_{25} + \sum_{12} \nabla_{W_1} F_{26} \quad \text{(A9)}$$

where

$$\nabla_{W_1} F_{21} = 2(P - \alpha_4) \nabla_{W_1} P + \frac{m}{\alpha_4} \left(\frac{P}{\alpha_4}\right)^{m-1} \nabla_{W_1} P \quad \text{(A10)}$$

$$\nabla_{W_1} F_{22} = 2(P - f_3) \nabla_{W_1} P + \frac{m}{f_3} \left(\frac{P}{f_3}\right)^{m-1} \nabla_{W_1} P \quad \text{(A11)}$$

$$\nabla_{W_1} F_{23} = 2(P - \beta) \nabla_{W_1} P \quad \text{(A12)}$$

$$\nabla_{W_1} F_{24} = 2(P - f_4) \nabla_{W_1} P + \frac{m}{f_4} \left(\frac{P}{f_4}\right)^{m-1} \nabla_{W_1} P \quad \text{(A13)}$$

$$\nabla_{W_1} F_{25} = 2(P - \alpha_5) \nabla_{W_1} P + \frac{m}{\alpha_5} \left(\frac{P}{\alpha_5}\right)^{m-1} \nabla_{W_1} P \quad \text{(A14)}$$

$$\nabla_{W_1} F_{26} = 2(P - \alpha_6) \nabla_{W_1} P + \frac{m}{\alpha_6} \left(\frac{P}{\alpha_6}\right)^{m-1} \nabla_{W_1} P \quad \text{(A15)}$$

So,

$$\nabla_{W_1} F(W, \mathrm{A}) = \nabla_{W_1} F_1(W, \mathrm{A}) + \nabla_{W_1} F_2(W, \mathrm{A}) \quad \text{(A16)}$$

Similarly,

$$\nabla_{W_2} P = 2\left((t_4 - t_3) S_1 + (t_1 + t_2) S_2\right) \quad \text{(A17)}$$

Thereof, Eq. (A17) is substituted $\nabla_{W_1} P$ above equations correspondingly and obtains the formulations of $\nabla_{W_2} F_1(W, \mathrm{A})$ and $\nabla_{W_2} F_2(W, \mathrm{A})$. Finally, we get

$$\nabla_{W_2} F(W, \mathrm{A}) = \nabla_{W_2} F_1(W, \mathrm{A}) + \nabla_{W_2} F_2(W, \mathrm{A}) \quad \text{(A18)}$$

$$\because (F_{11})'_{\alpha_1} = 2(\alpha_1 - P) - mP^m \alpha_1^{-m-1},$$

$$(F_{13})'_{\alpha_1} = 2(f_1 - P)(f_1)'_{\alpha_1} - mP^m f_1^{-m-1} (f_1)'_{\alpha_1}, \ (F_{14})'_{\alpha_1} = 2r(\beta - P),$$

$$(F_{15})'_{\alpha_1}=2(f_2-P)(f_2)'_{\alpha_1}-mP^m f_2^{-m-1}(f_2)'_{\alpha_1},$$
$$(F_{22})'_{\alpha_1}=2(f_3-P)(f_3)'_{\alpha_1}-mP^m f_3^{-m-1}(f_3)'_{\alpha_1},$$
$$(F_{23})'_{\alpha_1}=2r(\beta-P),\ (F_{24})'_{\alpha_1}=2(f_4-P)(f_4)'_{\alpha_1}-mP^m f_4^{-m-1}(f_4)'_{\alpha_1},$$
$$(\lg\alpha_1)'=1/\alpha_1$$
$$\therefore F'_{\alpha_1}=\sum_1 (F_{11})'_{\alpha_1}+\sum_3 (F_{13})'_{\alpha_1}+\sum_4 (F_{14})'_{\alpha_1}+\sum_5 (F_{15})'_{\alpha_1}$$
$$+\sum_8 (F_{22})'_{\alpha_1}+\sum_9 (F_{23})'_{\alpha_1}+\sum_{10} (F_{24})'_{\alpha_1}-\lambda/\alpha_1 \quad \text{(A19)}$$

Similarly, it is not difficult to compute $F'_{\alpha_i}(i=2,\,3,\,\ldots,\,6)$, finally, we can obtain

$$\nabla_{\mathrm{A}}F=\begin{pmatrix}F'_{\alpha_1} & F'_{\alpha_2} & \ldots & F'_{\alpha_6}\end{pmatrix}^T \quad \text{(A20)}$$

Using Eqs. (A16), (A18) and (A20), we obtain the formulation $\nabla_X F(W,\mathrm{A})$ as following:

$$\nabla_X F=(\nabla_{W_1}F\ \ \nabla_{W_2}F\ \ \nabla_{\mathrm{A}}F) \quad \text{(A21)}$$

Next, we calculate the Hessian matrices of the cost function. First, let

$$h_{11}=(t_1+t_2)S_1+(t_3+t_4)S_2\in\mathbb{R}^{N\times 1},\ h_1=h_{11}\otimes h_{11}^T+h_{11}^T\otimes h_{11}\in\mathbb{R}^{N\times N}$$
$$h_{21}=S_1\otimes S_1^T+S_2\otimes S_2^T\in\mathbb{R}^{N\times N},\ h_2=h_{21}+h_{21}^T\in\mathbb{R}^{N\times N}$$
$$h_{31}=(t_4-t_3)S_1+(t_1+t_2)S_2\in\mathbb{R}^{N\times 1},\ h_3=h_{31}\otimes h_{31}^T+h_{31}^T\otimes h_{31}\in\mathbb{R}^{N\times N}$$

then

$$\nabla^2_{W_1}F_{11}=2\left(2+m(m-1)P^{m-2}\alpha_1^{-m}\right)h_1+\left(2(P-\alpha_1)+mP^{m-1}\alpha_1^{-m}\right)h_2 \quad \text{(A22)}$$
$$\nabla^2_{W_1}F_{12}=2\left(2+m(m-1)P^{m-2}\alpha_2^{-m}\right)h_1+\left(2(P-\alpha_2)+mP^{m-1}\alpha_2^{-m}\right)h_2 \quad \text{(A23)}$$
$$\nabla^2_{W_1}F_{13}=2\left(2+m(m-1)P^{m-2}f_1^{-m}\right)h_1+\left(2(P-f_1)+mP^{m-1}f_1^{-m}\right)h_2 \quad \text{(A24)}$$
$$\nabla^2_{W_1}F_{14}=4h_1+2(P-\beta)h_2 \quad \text{(A25)}$$
$$\nabla^2_{W_1}F_{15}=2\left(2+m(m-1)P^{m-2}f_2^{-m}\right)h_1+\left(2(P-f_2)+mP^{m-1}f_2^{-m}\right)h_2 \quad \text{(A26)}$$
$$\nabla^2_{W_1}F_{16}=2\left(2+m(m-1)P^{m-2}\alpha_3^{-m}\right)h_1+\left(2(P-\alpha_3)+mP^{m-1}\alpha_3^{-m}\right)h_2 \quad \text{(A27)}$$
$$\nabla^2_{W_1}F_{21}=2\left(2+m(m-1)P^{m-2}\alpha_4^{-m}\right)h_1+\left(2(P-\alpha_4)+mP^{m-1}\alpha_4^{-m}\right)h_2 \quad \text{(A28)}$$

$$\nabla^2_{W_1}F_{22} = 2\left(2+m(m-1)P^{m-2}f_3^{-m}\right)h_1+\left(2(P-f_3)+mP^{m-1}f_3^{-m}\right)h_2 \quad \text{(A29)}$$

$$\nabla^2_{W_1}F_{23} = 4h_1 + 2(P-\beta)h_2 \quad \text{(A30)}$$

$$\nabla^2_{W_1}F_{24} = 2\left(2+m(m-1)P^{m-2}f_4^{-m}\right)h_1+\left(2(P-f_4)+mP^{m-1}f_4^{-m}\right)h_2 \quad \text{(A31)}$$

$$\nabla^2_{W_1}F_{25} = 2\left(2+m(m-1)P^{m-2}\alpha_5^{-m}\right)h_1+\left(2(P-\alpha_5)+mP^{m-1}\alpha_5^{-m}\right)h_2 \quad \text{(A32)}$$

$$\nabla^2_{W_1}F_{26} = 2\left(2+m(m-1)P^{m-2}\alpha_6^{-m}\right)h_1+\left(2(P-\alpha_6)+mP^{m-1}\alpha_6^{-m}\right)h_2 \quad \text{(A33)}$$

$$\nabla^2_{W_1}F = \sum_i \nabla^2_{W_1}F_{1i} + \sum_i \nabla^2_{W_1}F_{2i} \quad (i=1,2,\ldots,6) \quad \text{(A34)}$$

Similarly, h_3 is substituted h_1 above Eqs. (A22)–(A33) correspondingly and obtains the formulations of $\nabla^2_{W_2}F_{1i}$ and $\nabla^2_{W_2}F_{2i}$. Finally, we get

$$\nabla^2_{W_2}F = \sum_i \nabla^2_{W_2}F_{1i} + \sum_i \nabla^2_{W_2}F_{2i} \quad (i=1,2,\ldots 6) \quad \text{(A35)}$$

Based on the definition of Hessian matrix, we obtain $\nabla^2_{\mathrm{A}}F$ and $\nabla^2_{W_1W_2}F$ as following:

$$\nabla^2_{\mathrm{A}}F = \begin{pmatrix} \frac{\partial}{\partial\alpha_1} \\ \frac{\partial}{\partial\alpha_2} \\ \vdots \\ \frac{\partial}{\partial\alpha_6} \end{pmatrix} \begin{pmatrix} \frac{\partial F}{\partial\alpha_1} & \frac{\partial F}{\partial\alpha_2} & \cdots & \frac{\partial F}{\partial\alpha_6} \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 F}{\partial^2\alpha_1} & \frac{\partial^2 F}{\partial\alpha_1\partial\alpha_2} & \cdots & \frac{\partial^2 F}{\partial\alpha_1\partial\alpha_6} \\ \frac{\partial^2 F}{\partial\alpha_2\partial\alpha_1} & \frac{\partial^2 F}{\partial^2\alpha_2} & \cdots & \frac{\partial^2 F}{\partial\alpha_2\partial\alpha_6} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial\alpha_6\partial\alpha_1} & \frac{\partial^2 F}{\partial\alpha_6\partial\alpha_2} & \cdots & \frac{\partial^2 F}{\partial^2\alpha_6} \end{pmatrix} \quad \text{(A36)}$$

$$\nabla^2_{W_1W_2}F = \begin{pmatrix} \frac{\partial}{\partial w_1^1} \\ \frac{\partial}{\partial w_2^1} \\ \vdots \\ \frac{\partial}{\partial w_N^1} \end{pmatrix} \begin{pmatrix} \frac{\partial F}{\partial w_1^2} & \frac{\partial F}{\partial w_2^2} & \cdots & \frac{\partial F}{\partial w_N^2} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\partial}{\partial w_1^1}\left(\frac{\partial F}{\partial w_1^2}\right) & \frac{\partial}{\partial w_1^1}\left(\frac{\partial F}{\partial w_2^2}\right) & \cdots & \frac{\partial}{\partial w_1^1}\left(\frac{\partial F}{\partial w_N^2}\right) \\ \frac{\partial}{\partial w_2^1}\left(\frac{\partial F}{\partial w_1^2}\right) & \frac{\partial}{\partial w_2^1}\left(\frac{\partial F}{\partial w_2^2}\right) & \cdots & \frac{\partial}{\partial w_2^1}\left(\frac{\partial F}{\partial w_N^2}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial w_N^1}\left(\frac{\partial F}{\partial w_1^2}\right) & \frac{\partial}{\partial w_N^1}\left(\frac{\partial F}{\partial w_2^2}\right) & \cdots & \frac{\partial}{\partial w_N^1}\left(\frac{\partial F}{\partial w_N^2}\right) \end{pmatrix}$$

$$= \left(\nabla_{W_1}(\nabla_{W_2}F)_1 \quad \nabla_{W_1}(\nabla_{W_2}F)_2 \quad \ldots \quad \nabla_{W_1}(\nabla_{W_2}F)_N \right)_{N\times N}$$
$$\left(\because \nabla_{W_2}F = \left(\frac{\partial F}{\partial w_1^2} \quad \frac{\partial F}{\partial w_2^2} \ldots \frac{\partial F}{\partial w_N^2} \right)^T \right) \tag{A37}$$

Similarly,

$$\nabla^2_{W_1\mathrm{A}}F = (\nabla_{W_1}(\nabla_{\mathrm{A}}F)_1 \nabla_{W_1}(\nabla_{\mathrm{A}}F)_2 \ldots \nabla_{W_1}(\nabla_{\mathrm{A}}F)_6)_{N\times 6} \tag{A38}$$
$$\nabla^2_{W_2W_1}F = \left(\left(\nabla^2_{W_1W_2}F \right)^T \right)_{N\times N} \tag{A39}$$
$$\nabla^2_{W_2\mathrm{A}}F = (\nabla_{W_2}(\nabla_{\mathrm{A}}F)_1 \quad \nabla_{W_2}(\nabla_{\mathrm{A}}F)_2 \quad \ldots \quad \nabla_{W_2}(\nabla_{\mathrm{A}}F)_6)_{N\times 6} \tag{A40}$$
$$\nabla^2_{\mathrm{A}W_1}F = \left(\left(\nabla^2_{W_1\mathrm{A}}F \right)^T \right)_{6\times N} \tag{A41}$$
$$\nabla^2_{\mathrm{A}W_2}F = \left(\left(\nabla^2_{W_2\mathrm{A}}F \right)^T \right)_{6\times N} \tag{A42}$$

Using Eqs. (A36)–(A41), we finally obtain

$$\nabla^2_X F = \begin{pmatrix} \nabla^2_{W_1}F & \nabla^2_{W_1W_2}F & \nabla^2_{W_1\mathrm{A}}F \\ \left(\nabla^2_{W_1W_2}F\right)^T & \nabla^2_{W_2}F & \nabla^2_{W_2\mathrm{A}}F \\ \left(\nabla^2_{W_1\mathrm{A}}F\right)^T & \left(\nabla^2_{W_2\mathrm{A}}F\right)^T & \nabla^2_{\mathrm{A}}F \end{pmatrix}_{(2N+6)\times(2N+6)} \tag{A43}$$

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NNSF) under Grants 60990320, 60990323 and 61271090, the National 863 Project of China under Grant 2012AA012305, Sichuan Provincial Science and technology support Project under Grant 2012GZ0101, and Chengdu Science and technology support Project under Grant 12DXYB347JH-002.

REFERENCES

1. Li, X., W.-T. Li, X.-W. Shi, J. Yang, and J.-F. Yu, "Modified differential evolution algorithm for pattern synthesis of antenna arrays," *Progress In Electromagnetics Research*, Vol. 137, 371–388, 2013.
2. Lin, C., A.-Y. Qing, and Q.-Y. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Transactions on Antennas and Propagation*, Vol. 58, 2553–2561, 2010.
3. Li, R., L. Xu, X.-W. Shi, N. Zhang, and Z.-Q. Lv, "Improved differential evolution strategy for antenna array pattern synthesis

problems," *Progress In Electromagnetics Research*, Vol. 113, 429–441, 2011.

4. Liu, D., Q.-Y. Feng, W.-B. Wang, and X. Yu, "Synthesis of unequally spaced antenna arrays by using inheritance learning particle swarm optimization," *Progress In Electromagnetics Research*, Vol. 118, 205–221, 2011.
5. Liu, Y., Z.-P. Nie, and Q. H. Liu, "A new method for the synthesis of non-uniform linear arrays with shaped power patterns," *Progress In Electromagnetics Research*, Vol. 107, 349–363, 2010.
6. Wang, W.-B., Q.-Y. Feng, and D. Liu, "Application of chaotic particle swarm optimization algorithm to pattern synthesis of antenna arrays," *Progress In Electromagnetics Research*, Vol. 115, 173–189, 2011.
7. Zeng, T. J. and Q. Feng, "Penalty function solution to pattern synthesis of antenna array by a descent algorithm," *Progress In Electromagnetics Research B*, Vol. 49, 281–300, 2013.
8. Caorsi, S., et al., "Peak sidelobe level reduction with a hybrid approach based on GAs and difference sets," *IEEE Transactions on Antennas and Propagation*, Vol. 52, 1116–1121, 2004.
9. Lizzi, L., G. Oliveri, and A. Massa, "A time-domain approach to the synthesis of UWB antenna systems," *Progress In Electromagnetics Research*, Vol. 122, 557–575, 2012.
10. Oliveri, G., "Multibeam antenna arrays with common subarray layouts," *IEEE Antennas and Wireless Propagation Letters*, Vol. 9, 1190–1193, 2010.
11. Manica, L., et al., "Synthesis of multi-beam sub-arrayed antennas through an excitation matching strategy," *IEEE Transactions on Antennas and Propagation*, Vol. 59, 482–492, 2011.
12. Bregains, J., et al., "Synthesis of multiple-pattern planar antenna arrays with single prefixed or jointly optimized amplitude distributions," *Microwave and Optical Technology Letters*, Vol. 32, 74–78, 2002.
13. Rocca, P., et al., "Differential evolution as applied to electromagnetics," *IEEE Antennas and Propagation Magazine*, Vol. 53, 38–49, 2011.
14. Shi, Z. and S. Wang, "Nonmonotone adaptive trust region method," *European Journal of Operational Research*, Vol. 208, 28–36, 2011.
15. Ahookhosh, M., et al., "A nonmonotone trust-region line search method for large-scale unconstrained optimization," *Applied Mathematical Modelling*, Vol. 36, 478–487, 2012.

16. Zhang, J., et al., "A nonmonotone adaptive trust region method for unconstrained optimization based on conic model," *Applied Mathematics and Computation*, Vol. 217, 4265–4273, 2010.
17. Nocedal, J. and Y.-X. Yuan, "Combining trust region and line search techniques," Department of Electrical Engineering and Computer Science, Northwestern University, 1992.
18. Davidon, W. C., "Conic approximations and collinear scalings for optimizers," *SIAM Journal on Numerical Analysis*, Vol. 17, 268–281, 1980.
19. Di, S. and W. Sun, "A trust region method for conic model to solve unconstraind optimizaions," *Optimization methods and software*, Vol. 6, 237–263, 1996.
20. Ji, Y., et al., "A new nonmonotone trust-region method of conic model for solving unconstrained optimization," *Journal of Computational and Applied Mathematics*, Vol. 233, 1746–1754, 2010.
21. Deng, C., et al., "Modified dynamic differential evolution for 0-1 knapsack problems," *International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, 1–4, 2009.
22. Luebbers, R. and K. Kunz, "Finite difference time domain calculations of antenna mutual coupling," *IEEE Transactions on Electromagnetic Compatibility*, Vol. 34, 357–359, 1992.
23. Ye, Z. and C. Liu, "Non-sensitive adaptive beamforming against mutual coupling," *IET Signal Processing*, Vol. 3, 1–6, 2009.
24. Liao, B. and S.-C. Chan, "Adaptive beamforming for uniform linear arrays with unknown mutual coupling," *IEEE Antennas and Wireless Propagation Letters*, Vol. 11, 464–467, 2012.