Thinned Array Beampattern Synthesis by Iterative Soft-Thresholding based Optimization Algorithms

Xiangrong Wang, Elias Aboutanios, Senior Member and Moeness G. Amin, Fellow

***Abstract*—In this paper we cast thinned array beampattern synthesis as a linear inverse problem (LIP) and apply three iterative soft-thresholding based optimization algorithms to solve it effectively. We combine synthesis algorithms with beampattern error reweighting and beampattern phase adjustment to achieve accurate approximation of the desired reference pattern. The proposed methods can be used to synthesize arbitrarily shaped beampatterns, including multibeam forming and radiation suppression within several angular regions. These methods offer a number of advantages in regard to array scale, beampattern specifics and computational time compared to existing methods. Since this paper focuses on synthesizing receiver arrays, there are no explicit constraints applied to the excitation weights. Numerical results demonstrate the efficiency of three proposed approaches for designing non-uniformly spaced arrays. In particular, the fast iterative soft-thresholding algorithm can synthesize of up to a few hundred antennas.**

***Index Terms*—Beampattern synthesis, Array thinning, Iterative soft-thresholding, Alternating descent,Weighted least square.**

I. Introduction

The last fifty years have seen a growing demand for large aperture arrays exhibiting increased capabilities in terms of flexibility and reconfigurability, yet simultaneously offering reduced hardware cost and computational complexity [1], [2], [3]. Thinned arrays are ideal for satisfying these requirements, as they can maintain the same mainlobe width and peak sidelobe level (PSL) with a significant reduction in cost, weight and complexity. Moreover, thinned arrays are flexible and reconfigurable due to the periodic quantization of the element positions.

Adaptive algorithms can be broadly classified into open-loop and closed-loop techniques [4], [5]. Closed-loop reconfigurable adaptive array techniques have been previously considered, e.g. [6], [7]. In this paper, we consider array reconfigurability for open-loop null-steering algorithms. The high cost of an entire front-end per antenna makes large array interference nulling quite expensive. In order to reduce the hardware cost, a smaller number of front-ends are installed in the receiver. Thinning strategies then adaptively select an optimum subset of antennas over a full array layout to connect to the following beam forming network (BFN), whereas those belonging to the complementary subset are connected to matched loads or removed. The on-off status of the array elements is controlled by acting on a set of radio frequency (RF) switches [8], [9], [10], giving an easily-reconfigurable and low-complexity antenna architecture. Therefore, we propose an adaptive thinned array framework for open-loop interference suppression, as shown in Fig. 1.

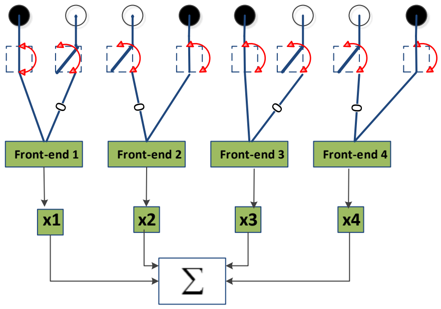


Fig. 1. The Diagram of the proposed reconfigurable array thinning strategy.

There are many effective methods for synthesizing thinned arrays existing in the literature. These include techniques such as heuristic method [11], convex optimization [8], [12], Bayesian Compressive Sensing [13], Matrix Pencil [14] to list a few. These methods, however, are prohibitively computationally expensive or can even fail when the scale of the problem becomes large. Yang [15] combined the iterative fast Fourier transform (FFT) algorithm [16], [17] and array interpolation together to synthesize large antenna arrays, but the convergence rate of his method is highly dependent on the choice of parameter values and initial search points. It is desirable to adaptively achieve arbitrary sidelobe levels and the required deep nulls for combating interferences and noise according to continuously changing environmental scenarios [18]. Thus developing fast array thinning algorithms becomes necessary for adaptive interference nulling [19].

In order to address these issues, we apply an iterative soft-thresholding based optimization method for array thinning, which has been shown to effectively address large-scale optimization problems in the areas of compressive sensing and image processing due to its simple structure. Each iteration of this algorithm comprises a multiplication by a matrix and its transpose, along with a scalar shrinkage step on the obtained result [20], [21], [22], [23]. However, when the problem scale is small or moderate with many constraints, the iterative soft-thresholding algorithms do not exhibit obvious advantages [23]. Our main contributions in this paper build on our initial work in [24] and are shown as follows: (1) We cast the beampattern synthesis problem into the appropriate form that

X. Wang and E. Aboutanios are with the School of Electrical Engineering and Telecommunication, University of New South Wales, Sydney, Australia 2052. e-mail:x.r.wang@unsw.edu.au; elias@unsw.edu.au.

M. G. Amin is with the Center for Advanced Communications, College of Engineering, Villanova University, Villanova, PA 19085, USA. e-mail: moeness.amin@villanova.edu.

is amenable to the application of iterative soft-thresholding algorithms. We also combine beampattern error reweighting and beampattern phase adjustment with the synthesis algorithm. (2) We propose the Amplitude Sparse Optimization (ASO) algorithm. (3) We provide a mathematical interpretation of the beampattern phase adjustment that offers an important insight into complex beampattern synthesis problems. (4) We introduce a vector form thresholding in lieu of the commonly used scalar threshold in order to preserve the array aperture length. (5) We analyse the sparse property and compare the advantages and disadvantages of the three algorithms in the area of beampattern synthesis.

A brief description of the notation used in this paper is given next. Bold lowercase letters (such as $\mathbf{v}$) denote column vectors and bold uppercase letters (such as $\mathbf{V}$) are reserved for matrices. We use $\mathbf{v}(i)$ to refer to the i^{th} element of $\mathbf{v}$, $\mathbf{v}_i$ and $\mathbf{v}_i^T$ to the i^{th} column and row vectors which are usually the i^{th} column and row of a corresponding matrix $\mathbf{V}$ respectively, and $\mathbf{v}_i^{(k)}$ is the i^{th} vector at the k^{th} iteration. $\mathbf{v}^T$ and $\mathbf{v}^H$ are the usual transpose and Hermitian operations. The symbols $\circ$ and $\oslash$ are element-wise multiplication and division respectively. $\|\cdot\|_0$, $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the l_0, l_1 and l_2 norm operations respectively, while $\|\mathbf{v}\|_\mathbf{M}$ means $\mathbf{v}^H\mathbf{M}\mathbf{v}$. $\lambda_{\text{max}}(\mathbf{M})$ means the maximum eigenvalue of matrix $\mathbf{M}$. $|\mathbf{v}|$ means taking the absolute value of the vector $\mathbf{v}$ element-wise. $\text{sign}(\cdot)$ takes value 1 if $\cdot > 0$, value -1 if $\cdot < 0$ and value 0 if $\cdot = 0$. $\text{diag}(\mathbf{v})$ is a diagonal matrix with the vector $\mathbf{v}$ along the diagonal.

The paper is organised as follows. In section II, we describe the LIP model of thinned array beampattern synthesis and the three iterative soft-thresholding based optimization algorithms. In section III, we show numerical results to demonstrate the efficiency of the proposed approach. Finally, some conclusions are drawn in section IV.

II. Beampattern Synthesis Algorithms

Consider an N-antenna array in the x-y plane of a Cartesian coordinate system and let the vector $\mathbf{P} \in \mathbb{R}^{N\times 2}$ contain the positions of the array elements. A plane wave with wavelength λ is impinging on the array from a direction of arrival (DOA) that is specified by elevation angle, θ, and azimuth angle, ϕ. The steering vector of the array in the look direction (θ, ϕ) is

$$\mathbf{a}(\theta,\phi) = e^{j\frac{2\pi}{\lambda}\mathbf{P}[u_x,u_y]^T}, \tag{1}$$

where $u_x = \cos\theta\cos\phi$ and $u_y = \cos\theta\sin\phi$. We are interested in synthesizing a desired beampattern for this array using a minimum number of antennas. This is formulated in the next subsection.

A. Problem Formulation

Let us assume that we have a desired reference pattern, $\mathbf{f}_\text{d}$, over some $(K\times L)$ sampling of the elevation and azimuth ranges, where $KL > N$. That is $\mathbf{f}_\text{d}$ is available for the KL positions $\theta_1, \theta_2, ..., \theta_K$ and $\phi_1, \phi_2, ..., \phi_L$. Let $\mathbf{x} \in \mathbb{C}^N$ denote the complex excitation vector. Then the array response can be expressed as

$$\mathbf{f}_\text{d} = \mathbf{A}\mathbf{x}, \tag{2}$$

where $\mathbf{A} \in \mathbb{C}^{KL\times N}$ and

$$\mathbf{A} = [\mathbf{a}(\theta_1,\phi_1), ..., \mathbf{a}(\theta_1,\phi_L), ..., \mathbf{a}(\theta_K,\phi_1), ..., \mathbf{a}(\theta_K,\phi_L)]^T.$$

Now we write the desired reference pattern comprised of two parts, the amplitude and phase, as follows:

$$\mathbf{f}_\text{d} = \mathbf{f}_\text{dM} \circ \mathbf{f}_\text{dP} = \mathbf{F}_\text{dM}\mathbf{f}_\text{dP}, \tag{3}$$

where $\circ$ means Hadamard product, and $\mathbf{F}_\text{dM} = \text{diag}(\mathbf{f}_\text{dM})$. The thinned array beampattern synthesis problem can be formulated as composing a subarray with as few antennas as possible that gives the desired beampattern $\mathbf{f}_\text{d}$, i.e.,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_0, \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{f}_\text{d}. \end{aligned} \tag{4}$$

The l_0 norm, $\|\mathbf{x}\|_0$, is defined as the number of nonzero entries of the excitation vector $\mathbf{x}$. So if $x_i = 0$, the i^{th} antenna is discarded; otherwise the nonzero value is taken as the corresponding complex excitation for it. The l_0-norm minimization problem is non-convex and generally very hard to solve, as its solution usually requires an intractable combinatorial search [25]. Therefore, we relax Eq. (4) by replacing the l_0 norm with the well-known l_1 norm [25], and replace the equality constraint by an inequality to give the new problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_1, \\ \text{s.t.} \quad & \|\mathbf{A}\mathbf{x} - \mathbf{f}_\text{d}\|_2 \le \epsilon, \end{aligned} \tag{5}$$

for some fidelity parameter $\epsilon \ge 0$. In order to control the trade-off among the errors in the mainlobe ripple, sidelobe level and null depth, we introduce a weight vector $\mathbf{w}$ (the corresponding weight matrix is the diagonal matrix $\mathbf{W} = \text{diag}(\mathbf{w})$) to reweight the l_2-norm of the error. Then, if we want to maintain a high fidelity (i.e. smaller error) in the synthesised beampattern in some looking directions, we make the entries of $\mathbf{w}$ that correspond to these look directions larger. Eq. (5) can be re-cast into the following unconstrained weighted least squared form,

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\beta_2}{2}\|\mathbf{A}\mathbf{x} - \mathbf{f}_\text{d}\|_\mathbf{W}^2, \tag{6}$$

where the matrix norm is defined as $\|\mathbf{A}\mathbf{x} - \mathbf{f}_\text{d}\|_\mathbf{W}^2 = (\mathbf{A}\mathbf{x} - \mathbf{f}_\text{d})^H\mathbf{W}(\mathbf{A}\mathbf{x} - \mathbf{f}_\text{d})$. $\beta_2 > 0$ is a regularization parameter that balances between the solution sparsity and the minimization of the synthesized beampattern error. Generally, a smaller β_2 yields a sparser minimizer $\mathbf{x}$, but also a larger error between the synthesised and desired beampatterns.

B. Transformation from the complex to the real domain

The problem presented above is complex. Therefore, we transform it to the real domain in order to implement the optimization algorithms directly. The transformation is defined as follows:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathcal{R}(\mathbf{A}) & -\mathcal{I}(\mathbf{A}) \\ \mathcal{I}(\mathbf{A}) & \mathcal{R}(\mathbf{A}) \end{bmatrix}, \tag{7}$$

$$\tilde{\mathbf{x}} = [\mathcal{R}(\mathbf{x})^T, \mathcal{I}(\mathbf{x})^T]^T, \tag{8}$$

$$\tilde{\mathbf{f}_\text{d}} = [\mathcal{R}(\mathbf{f}_\text{d})^T, \mathcal{I}(\mathbf{f}_\text{d})^T]^T, \tag{9}$$

$$\tilde{\mathbf{w}} = [\mathbf{w}^T, \mathbf{w}^T]^T, \tag{10}$$

$$\tilde{\mathbf{W}} = \text{diag}(\tilde{\mathbf{w}}), \tag{11}$$

where $\mathcal{R}(\cdot)$ means the real part of $\cdot$ and $\mathcal{I}(\cdot)$ its imaginary part. Using Eqs. (7) - (11), the optimization problem in Eq. (6) becomes

$$\min_{\tilde{\mathbf{x}}} \quad \|\tilde{\mathbf{x}}\|_{2,1} + \frac{\beta_2}{2}\|\tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{f}}_{\text{d}}\|^2_{\tilde{\mathbf{W}}}. \tag{12}$$

In this formulation the $l_{2,1}$ norm is defined as

$$\begin{aligned} \|\tilde{\mathbf{x}}\|_{2,1} &= \|\mathbf{x}\|_1 \\ &= \sum_{i=1}^{N} \sqrt{\mathcal{R}^2(\mathbf{x}(i)) + \mathcal{I}^2(\mathbf{x}(i))} \\ &= \sum_{i=1}^{N} \|\tilde{\mathbf{x}}_{g_i}\|_2, \end{aligned} \tag{13}$$

where

$$\tilde{\mathbf{x}}_{g_i} = [\tilde{\mathbf{x}}(i), \tilde{\mathbf{x}}(i+N)]^T, i = 1, ..., N. \tag{14}$$

The weighted l_2 norm, on the other hand, remains the same:

$$\|\tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{f}}_{\text{d}}\|^2_{\tilde{\mathbf{W}}} = \|\mathbf{A}\mathbf{x} - \mathbf{f}_{\text{d}}\|^2_{\mathbf{W}}. \tag{15}$$

C. Setting the desired beampattern phase

In most applications of antenna arrays in communications and radar [26], we do not have a phase requirement on the directional response of the array. Therefore, we can use the phase of the desired beampattern, $\mathbf{f}_{\text{dP}}$, as an extra free parameter in the optimisation in order to make the synthesized beampattern approach the desired one. Substituting Eq. (3) into the objective function in Eq. (6) yields,

$$\|\mathbf{x}\|_1 + \frac{\beta_2}{2}\|\mathbf{A}\mathbf{x} - \mathbf{f}_{\text{d}}\|^2_{\mathbf{W}} = \|\mathbf{x}\|_1 + \frac{\beta_2}{2}\|\mathbf{A}\mathbf{x} - \mathbf{F}_{\text{dM}}\mathbf{f}_{\text{dP}}\|^2_{\mathbf{W}}. \tag{16}$$

It is clear that Eq. (16) is non-convex with respect to both variables $\mathbf{x}$ and $\mathbf{f}_{\text{dP}}$. Thus we utilise an alternating descent method that iteratively shifts between the two variables $\mathbf{x}$ and $\mathbf{f}_{\text{dP}}$. The l_1 norm in the above equation does not depend on $\mathbf{f}_{\text{dP}}$. Thus, we choose $\mathbf{f}_{\text{dP}}$ that minimises the re-weighted l_2 norm, which reduces the error between the desired and synthesised beampattern. Given the excitation vector $\mathbf{x}$, we take the derivative of Eq. (16) with respect to $\mathbf{f}_{\text{dP}}$ and set it to zero to give

$$\mathbf{f}^{(u)}_{\text{dP}} = \mathbf{F}^{-1}_{\text{dM}}\mathbf{A}\mathbf{x}. \tag{17}$$

Since the entries of the phase vector $\mathbf{f}_{\text{dP}}$ have unit magnitude, and noting that the i^{th} entry of $\mathbf{A}\mathbf{x}$ is $\mathbf{a}_i^T\mathbf{x}$, we normalise each entry of $\mathbf{f}^{(u)}_{\text{dP}}$ as follows:

$$\mathbf{f}_{\text{dP}}(i) = \frac{\mathbf{f}^{(u)}_{\text{dP}}(i)}{|\mathbf{f}^{(u)}_{\text{dP}}(i)|} = \frac{\mathbf{a}_i^T\mathbf{x}}{|\mathbf{a}_i^T\mathbf{x}|}. \tag{18}$$

Eq. (18) reveals an interesting insight of the complex beampattern synthesis problem, which should be formulated as the following power pattern synthesis,

$$\min_{\mathbf{x}} \; \|\mathbf{x}\|_1 + \frac{\beta_2}{2}\||\mathbf{A}\mathbf{x}| - \mathbf{f}_{\text{dM}}\|^2_{\mathbf{W}}. \tag{19}$$

However, since the function $|\mathbf{A}\mathbf{x}|$ is non-differentiable with respect to the complex vector variable $\mathbf{x}$, it is difficult to minimize Eq. (19) directly. Therefore we retain the formulation in Eq. (16) and update the desired beampattern phase $\mathbf{f}_{\text{dP}}$ in each iteration. Now let us denote $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k+1)}$ as solutions of the k^{th} and $(k+1)^{th}$ iterations respectively. Then combining Eqs. (16) and (18) in the $(k+1)^{th}$ iteration yields,

$$\|\mathbf{x}^{(k+1)}\|_1 + \frac{\beta_2}{2}\|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{F}_{\text{dM}}(\mathbf{A}\mathbf{x}^{(k)} \oslash |\mathbf{A}\mathbf{x}^{(k)}|)\|^2_{\mathbf{W}}. \tag{20}$$

Here $\oslash$ denotes the element-wise division as defined in Eq. (18). Since the distance between two successive solutions $\mathbf{x}^{(k+1)}$ and $\mathbf{x}^{(k)}$ becomes small with the convergence of the algorithm, we have that $\mathbf{x}_o = \mathbf{x}^{(k)} \simeq \mathbf{x}^{(k+1)}$. Then proceeding from Eq. (20) yields

$$\begin{aligned} &\|\mathbf{x}_o\|_1 + \frac{\beta_2}{2}\|\mathbf{A}\mathbf{x}_o - \mathbf{F}_{\text{dM}}(\mathbf{A}\mathbf{x}_o \oslash |\mathbf{A}\mathbf{x}_o|)\|^2_{\mathbf{W}} \\ = \; &\|\mathbf{x}_o\|_1 + \frac{\beta_2}{2}\||\mathbf{A}\mathbf{x}_o| - \mathbf{f}_{\text{dM}}\|^2_{\mathbf{W}}, \end{aligned} \tag{21}$$

where $\mathbf{A}\mathbf{x}_o = |\mathbf{A}\mathbf{x}_o| \circ (\mathbf{A}\mathbf{x}_o \oslash |\mathbf{A}\mathbf{x}_o|)$ with $\circ$ denoting element-wise product. We can see that Eq. (21) converges to the power pattern synthesis Eq. (19) finally.

D. Setting the beampattern weight

The challenge for the least-squared optimization is the control of sidelobe levels. As shown in [12], the shaped beampattern synthesis with PSL constraints is expressed as:

$$\begin{cases} -\delta \leq |\mathbf{A}\mathbf{x}| - \mathbf{f}_{\text{dM}} \leq \delta, & \text{for mainlobe region,} \\ |\mathbf{A}\mathbf{x}| - \mathbf{f}_{\text{dM}} \leq 0, & \text{for sidelobe and null region,} \end{cases} \tag{22}$$

where δ is a fidelity factor of the beampattern error in the mainlobe region. From Eq. (22), we can observe that the desired beampattern amplitude $\mathbf{f}_{\text{dM}}$ is actually the "mask" in the sidelobe region, as $\mathbf{f}_{\text{dM}}$ sets the upper bound on the sidelobe level and the lower bound is $-\infty$. However, for the mainlobe region, $\mathbf{f}_{\text{dM}}$ is not a "mask" but the desired reference pattern that should be achieved with a tolerance error δ. Thus the upper and lower bounds for the mainlobe region are $\mathbf{f}_{\text{dM}} + \delta$ and $\mathbf{f}_{\text{dM}} - \delta$ respectively. Therefore, large negative differences between the sidelobes of the synthesised and desired beampatterns satisfy the requirements. The least-squared optimization, however, treats both the negative and positive errors in the same way, therefore trying to match the desired sidelobe level rather than bettering it. In order to circumvent this problem, we update the weight vector $\mathbf{w}$ iteratively (or $\tilde{\mathbf{w}}$) using a similar method to [27]. We adjust the weight vector value element-wise in the $(k+1)^{th}$ iteration as follows:

Let us denote the i^{th} entry of the error $|\mathbf{A}\mathbf{x}^{(k)}| - \mathbf{f}_{\text{dM}}$ by $\mathbf{e}^{(k)}(i)$. That is $\mathbf{e}^{(k)}(i) = |\mathbf{a}_i^T\mathbf{x}^{(k)}| - \mathbf{f}_{\text{dM}}(i)$. Then for the mainlobe region,

$$\mathbf{w}^{(k+1)}(i) \quad = \quad \begin{cases} \mathbf{w}^{(k)}(i), & |\mathbf{e}^{(k)}(i)| \leq \delta \\ \mathbf{w}^{(k)}(i) + k_m|\mathbf{e}^{(k)}(i)|, & \text{otherwise.} \end{cases} \tag{23}$$

For the sidelobe region,

$$\mathbf{w}^{(k+1)}(i) = \max\{k_s\mathbf{e}^{(k)}(i), 0\}, \tag{24}$$

and for the null region,

$$\mathbf{w}^{(k+1)}(i) = \max\{k_n \mathbf{e}^{(k)}(i), 0\}, \tag{25}$$

where k_m, k_s and k_n are positive gain factors corresponding to the mainlobe, sidelobe and null respectively. Observe that for the mainlobe region, $\mathbf{w}$ never decreases from its initial value. Whereas for both the sidelobe and null, the weight $\mathbf{w}$ is zero if the synthesised beampattern is lower than the desired level.

E. Group Sparse Optimization

In this section, we will apply the Group Sparse Optimization (GSO) method to deal with the mixed $l_{2,1}$-norm regularisation in Eq. (12). The GSO approach introduced in [28], [23] is based on a variable splitting strategy and the classic alternating direction method (ADM). The convergence rate of the GSO method is guaranteed by the existing ADM theory.

Now we introduce an auxiliary variable $\mathbf{z}$ to Eq. (12), giving

$$\begin{aligned} \min_{\tilde{\mathbf{x}},\mathbf{z}} \quad & \|\mathbf{z}\|_{2,1} + \frac{\beta_2}{2} \parallel \tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{f}}_\text{d}\|^2_{\tilde{\mathbf{W}}}, \\ \text{s.t.} \quad & \mathbf{z} = \tilde{\mathbf{x}}. \end{aligned} \tag{26}$$

The augmented Lagrangian problem is of the form,

$$\min_{\tilde{\mathbf{x}},\mathbf{z}} \|\mathbf{z}\|_{2,1} - \boldsymbol{\lambda}^T(\mathbf{z} - \tilde{\mathbf{x}}) + \frac{\beta_1}{2}\|\mathbf{z} - \tilde{\mathbf{x}}\|_2^2 + \frac{\beta_2}{2}\|\tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{f}}_\text{d}\|^2_{\tilde{\mathbf{W}}}, \tag{27}$$

where $\boldsymbol{\lambda}$ is a multiplier and β_1 is a penalty parameter. Using the ADM approach, we minimize the augmented Lagrangian in Eq.(27) with respect to $\tilde{\mathbf{x}}$ and $\mathbf{z}$ alternatively. Minimizing with respect to $\tilde{\mathbf{x}}$ yields,

$$\tilde{\mathbf{x}} = (\beta_1 \mathbf{I} + \beta_2 \tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} (\beta_1 \mathbf{z} - \boldsymbol{\lambda} + \beta_2 \tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{f}}_\text{d}). \tag{28}$$

When the matrix $\tilde{\mathbf{A}}$ has more columns than rows, we can reduce the computational complexity of the matrix inversion by using the Sherman-Morrison-Woodbury formula:

$$(\beta_1 \mathbf{I} + \beta_2 \tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} = \frac{1}{\beta_1}\mathbf{I} - \frac{\beta_2}{\beta_1}\tilde{\mathbf{A}}^T(\beta_1 \tilde{\mathbf{W}}^{-1} + \beta_2 \tilde{\mathbf{A}}\tilde{\mathbf{A}}^T)^{-1}\tilde{\mathbf{A}}. \tag{29}$$

Now minimizing over $\mathbf{z}$ yields the following problem

$$\min_{\mathbf{z}} \|\mathbf{z}\|_{2,1} - \boldsymbol{\lambda}^T \mathbf{z} + \frac{\beta_1}{2}\|\mathbf{z} - \tilde{\mathbf{x}}\|_2^2. \tag{30}$$

By some simple manipulations, the problem in Eq. (30) is equivalent to

$$\min_{\mathbf{z}} \sum_{i=1}^{N} \left[\|\mathbf{z}_{g_i}\|_2 + \frac{\beta_1}{2} \|\mathbf{z}_{g_i} - \tilde{\mathbf{x}}_{g_i} - \frac{1}{\beta_1}\boldsymbol{\lambda}_{g_i}\|_2^2 \right], \tag{31}$$

where similar to the definition of $\tilde{\mathbf{x}}_{g_i}$ in Eq. (14), we have that

$$\boldsymbol{\lambda}_{g_i} = [\boldsymbol{\lambda}(i), \boldsymbol{\lambda}(i+N)]^T, i = 1, ..., N, \tag{32}$$

and

$$\mathbf{z}_{g_i} = [\mathbf{z}(i), \mathbf{z}(i+N)]^T, i = 1, ..., N. \tag{33}$$

Eq. (31) has a closed form solution using the vector-form soft-thresholding formula:

$$\mathbf{z}_{g_i} = \max\{\|\mathbf{r}_i\|_2 - \frac{1}{\beta_1}, 0\}\frac{\mathbf{r}_i}{\|\mathbf{r}_i\|_2}, \tag{34}$$

where

$$\mathbf{r}_i = \tilde{\mathbf{x}}_{g_i} + \frac{1}{\beta_1}\boldsymbol{\lambda}_{g_i}. \tag{35}$$

For brevity of notation, we refer to the above group-wise soft-thresholding operator by $\mathbf{z} = \text{Shrink}(\tilde{\mathbf{x}} + \frac{1}{\beta_1}\boldsymbol{\lambda}, \frac{1}{\beta_1})$. Finally, the multiplier $\boldsymbol{\lambda}$ is updated in the standard way:

$$\boldsymbol{\lambda} = \boldsymbol{\lambda} - \gamma\beta_1(\mathbf{z} - \tilde{\mathbf{x}}), \tag{36}$$

where γ is the step length. In this method, the preference of some specific antennas or the preservation of maximum aperture length can be realised by changing $\|\mathbf{z}\|_{2,1}$ to $\|\mathbf{z}\|_{\mathbf{t},2,1}$ in Eq. (26), where $\mathbf{t}(i)$ is the weight imposed on the i^{th} antennas and $\|\mathbf{z}\|_{\mathbf{t},2,1} = \sum_{i=1}^{N} \mathbf{t}(i)\|\mathbf{z}_{g_i}\|_2$. Then the group-wise soft-thresholding operator becomes $\mathbf{z} = \text{Shrink}(\tilde{\mathbf{x}} + \frac{1}{\beta_1}\boldsymbol{\lambda}, \frac{\mathbf{t}}{\beta_1})$.

Now we summarise the implementation procedure of the GSO algorithm as follows:

- **Initialisation:** Set the initial desired beampattern phase to zero, i.e. $\mathbf{f}_\text{dP} = \mathbf{1}$; Initialise $\mathbf{z}^{(0)} = \tilde{\mathbf{x}}^{(0)} = \mathbf{0}$, $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$, $\gamma = (\sqrt{5}+1)/2$, $\beta_1 = 10/\text{mean}(\mathbf{f}_\text{dM})$ and $\beta_2 = 1/\text{mean}(\mathbf{f}_\text{dM})$; Initialise k_m, k_s, k_n - for example set $k_m = 40, k_s = 4000, k_n = 10^5$;
- **Outer Iteration:** *while* the desired beampattern has not been achieved *do*
    1) **Inner Iteration:** *while* $\|\tilde{\mathbf{x}}^{(k+1)} - \tilde{\mathbf{x}}^{(k)}\|_2 > \nu$ *do*
        a) Transform the problem from the complex to the real domain according to Eqs. (7)-(11);
        b) $\mathbf{z}^{(k+1)} = \text{Shrink}(\tilde{\mathbf{x}}^{(k)} + \frac{1}{\beta_1}\boldsymbol{\lambda}^{(k)}, \frac{1}{\beta_1})$;
        c) $\tilde{\mathbf{x}}^{(k+1)} = (\beta_1\mathbf{I} + \beta_2\tilde{\mathbf{A}}^T\tilde{\mathbf{W}}\tilde{\mathbf{A}})^{-1}(\beta_1\mathbf{z}^{(k+1)} - \boldsymbol{\lambda}^{(k)} + \beta_2\tilde{\mathbf{A}}^T\tilde{\mathbf{W}}\tilde{\mathbf{f}}_\text{d})$;
        d) $\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} - \gamma_1\beta_1(\mathbf{z}^{(k+1)} - \tilde{\mathbf{x}}^{(k+1)})$;
        e) Transform the real solution $\tilde{\mathbf{x}}^{(k+1)}$ back to a complex solution $\mathbf{x}^{(k+1)}$ by $\mathbf{x}^{(k+1)}(i) = \tilde{\mathbf{x}}^{(k+1)}(i) + j\tilde{\mathbf{x}}^{(k+1)}(i+N)$. Update the desired beampattern phase $\mathbf{f}_\text{dP}$ using Eq. (18). Go back to the start of the Inner Iteration.

        End Inner Iteration.
    2) Update the weight vector $\mathbf{w}$ according to Eq. (23) - Eq. (25); Go back to Outer Iteration.

    End Outer Iteration.

In the above procedure, the inner iterations are stopped when two consective solutions are close enough to each other. This is controlled by setting ν to the desired tolerance.

F. Fast Iterative Soft-Thresholding Algorithm

In this method, we relax the regularizer $\|\tilde{\mathbf{x}}\|_{2,1}$ to $\|\tilde{\mathbf{x}}\|_1$. Intuitively, this relaxation decouples the real and imaginary parts from each other and one would expect it not to be able to synthesize very sparse arrays with respect to the coupled case. In the extreme, the array weights can be either purely real or purely imaginary, which leads to small $\|\tilde{\mathbf{x}}\|_0$ norm (half the original value) but an entirely non-sparse array [29]. However, as we will show mathematically, the soft-thresholding operator prevents such problems from occurring. We will in fact elucidate the relationship between the two.

Let us rewrite the relaxed version of the optimisation problem in Eq. (12) as follows,

$$\min_{\tilde{\mathbf{x}}} \qquad \|\tilde{\mathbf{x}}\|_1 + \frac{\beta_2}{2} \parallel \tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{f}}_{\rm d} \parallel_{\tilde{\mathbf{W}}}^2 . \tag{37}$$

Using the function $G(\tilde{\mathbf{x}})$ to represent the second part of Eq. (37), the gradient of $G(\tilde{\mathbf{x}})$ denoted by $\triangledown G(\tilde{\mathbf{x}})$ is

$$\triangledown G(\tilde{\mathbf{x}}) = \tilde{\mathbf{A}}^H \tilde{\mathbf{W}}(\tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{f}}_{\rm d}). \tag{38}$$

It is well known that the minimizer of Eq. (37) can be obtained by an iterative soft-thresholding algorithm [21], [22]. That is

$$\tilde{\mathbf{x}}^{(k+1)} = \operatorname*{argmin}_{\tilde{\mathbf{x}}} \|\tilde{\mathbf{x}}\|_1 + \frac{\tau\beta_2}{2}\|\tilde{\mathbf{x}} - (\tilde{\mathbf{x}}^{(k)} - \frac{1}{\tau}\triangledown G(\tilde{\mathbf{x}}^{(k)}))\|_2^2, \tag{39}$$

with $\tau \geq \lambda_{\max}(\tilde{\mathbf{A}}^H\tilde{\mathbf{W}}\tilde{\mathbf{A}})$. Then the unique minimizer of Eq. (39) is

$$\tilde{\mathbf{x}}^{(k+1)} = \text{Shrink}\left(\tilde{\mathbf{x}}^{(k)} - \frac{1}{\tau}\triangledown G(\tilde{\mathbf{x}}^{(k)}), \frac{1}{\tau\beta_2}\right). \tag{40}$$

Similarly to the vector-form soft-thresholding operator in Eq. (34), the scalar soft-thresholding operator is defined as,

$$\text{Shrink}\left(\cdot, \frac{1}{\tau\beta_2}\right) = \text{sign}(\cdot)\max\{|\cdot| - \frac{1}{\tau\beta_2}, 0\}. \tag{41}$$

It is well-known that the array aperture length is a crucial factor for the mainlobe width. In some applications, there are stringent requirements on the array resolution and mainlobe width. Here we propose a modification that allows us to preserve the maximum array aperture length, namely we replace the constant scalar β_2, by a constant vector $\boldsymbol{\beta_2} \in \mathbb{R}^N$. Thus a larger entry of $\boldsymbol{\beta_2}$ implies lower threshold, and a higher likelihood of selecting the corresponding antenna. So for example, if we hope to preserve the full array aperture length, we can assign large values to the entries of $\boldsymbol{\beta_2}$ corresponding to the boundary antennas.

It has been shown in [22] that the convergence rate of the traditional soft-thresholding algorithm of Eq. (40) is linear, which is not fast enough for reconfigurable array applications. In order to accelerate the convergence speed, a fast iterative soft-thresholding algorithm (FISTA) that uses a linear combination of the previous two points was proposed in [21]. This method can achieve a quadratic convergence rate. Starting with $\mathbf{y}^{(1)} = \tilde{\mathbf{x}}^{(0)}$ and $t_1 = 1$, we make at the k^{th} iteration the following calculations

$$\tilde{\mathbf{x}}^{(k)} = \text{Shrink}\left(\mathbf{y}^{(k)} - \frac{1}{\tau}\triangledown G(\mathbf{y}^{(k)}), \frac{1}{\tau\beta_2}\right), \tag{42}$$

$$t^{(k+1)} = \frac{1 + \sqrt{1 + 4(t^{(k)})^2}}{2}, \tag{43}$$

$$\mathbf{y}^{(k+1)} = \tilde{\mathbf{x}}^{(k)} + \left(\frac{t^{(k)} - 1}{t^{(k+1)}}\right)\left(\tilde{\mathbf{x}}^{(k)} - \tilde{\mathbf{x}}^{(k-1)}\right). \tag{44}$$

It is desirable to design thinned arrays having no more elements than necessary. For this reason, the continuation strategy proposed in [22] is adopted here to adjust the value of β_2 for automatic determination of the required number of antennas during the design process. Now we give the detailed implementation procedure of the FISTA as follows:

- **Initialisation:** Initialise the desired beampattern phase to zero, i.e. $\mathbf{f}_{\rm dP} = \mathbf{1}$. Set $\tilde{\mathbf{x}}^{(0)} = \mathbf{0}$. Initialise the trade-off parameter β_2 according to the sparsity requirement, and set the values for k_m, k_s, k_n.
- **Outer Iteration:** *while* the desired beampattern has not been achieved *do*
  1) **Inner Iteration:** *while* $\|\tilde{\mathbf{x}}^{(k+1)} - \tilde{\mathbf{x}}^{(k)}\|_2 > \nu$ *do*
     a) Transform the problem from the complex to the real domain according to Eqs. (7)-(11);
     b) Obtain $\tilde{\mathbf{x}}^{(k)}$ using Eqs. (42)-(44);
     c) Transform the real solution $\tilde{\mathbf{x}}^{(k+1)}$ back to a complex solution $\mathbf{x}^{(k+1)}$ by $\mathbf{x}^{(k+1)}(i) = \tilde{\mathbf{x}}^{(k+1)}(i) + j\tilde{\mathbf{x}}^{(k+1)}(i+N)$. Update the desired beampattern phase $\mathbf{f}_{\rm dP}$ using Eq. (18). Go back to the start of the Inner Iteration;

     End Inner Iteration.
  2) Update the weight vector $\mathbf{w}$ according to Eqs. (23)-(25);
  3) Adjust the value of β_2 utilising the continuation strategy outlined below; Go back to the Outer Iteration.

  End Outer Iteration.

Continuation Strategy:

Let $N_a^{(i)}$, $\beta_2^{(i)}$ and $\Delta\beta_2^{(i)}$ denote respectively the number of selected antennas, the value of β_2 and the step size at the i^{th} iteration. While the desired beampattern has not been reached, we adjust the value of β_2 in every iteration as follows:

1) If $N_a^{(i-1)} < K$ then $\beta_2^{(i)} = \beta_2^{(i-1)} + \Delta\beta_2^{(i)}$, where
   a) if $\beta_2^{(i-1)} > \beta_2^{(i-2)}$ (that is β_2 was increased in the previous iteration), then $\Delta\beta_2^{(i)} = \Delta\beta_2^{(i-1)}$;
   b) if $\beta_2^{(i-1)} < \beta_2^{(i-2)}$ (that is β_2 was decreased in the previous iteration), then $\Delta\beta_2^{(i)} = 0.5\Delta\beta_2^{(i-1)}$;
2) If $N_a^{(i-1)} > K$ then $\beta_2^{(i)} = \beta_2^{(i-1)} - \Delta\beta_2^{(i)}$ where
   a) if $\beta_2^{(i-1)} > \beta_2^{(i-2)}$ (that is β_2 was increased in the previous iteration), then $\Delta\beta_2^{(i)} = 0.5\Delta\beta_2^{(i-1)}$;
   b) if $\beta_2^{(i-1)} < \beta_2^{(i-2)}$ (that is β_2 was decreased in the previous iteration), then $\Delta\beta_2^{(i)} = \Delta\beta_2^{(i-1)}$.

Here K is the number of front-ends installed in the receiver. In our work, we have found that setting the initial step size, $\Delta\beta_2$ equal to the initial value of β_2, works quite well.

G. Sparsity Performance of the FISTA

As mentioned above, the decoupling of the real and imaginary parts from each other might be expected to produce a less sparse solution by suppressing only either the real or imaginary parts to zero. In this section, we give a simple proof of the optimality of Eq. (40) that gives insight into the comparable sparsity that is achieved by the FISTA and GSO approaches. Let us denote the optimum solutions obtained from the FISTA and the GSO to be $\tilde{\mathbf{x}}_{oF}$ and $\tilde{\mathbf{x}}_{oG}$ respectively. We define a mapping $\mathcal{M} : \mathbb{R}^{2N} \rightarrow \mathbb{R}_+^N$ as

$$\mathcal{M}(\tilde{\mathbf{x}}_{oT})(i) = \max\{|\tilde{\mathbf{x}}_{oT}(i)|, |\tilde{\mathbf{x}}_{oT}(i+N)|\}, \; T = F, G. \tag{45}$$

That is the mapping which selects the maximum absolute value between the real and imaginary components of the corresponding excitation vector element for each antenna. Then the number of selected antennas for each algorithm can be expressed as the l_0-norm of $\mathcal{M}(\tilde{\mathbf{x}}_{oT})$.

Theorem: The number of selected antennae of the FISTA is always less than or equal to that of the GSO for the same threshold value, i.e.

$$\|\mathcal{M}(\tilde{\mathbf{x}}_{oF})\|_0 \leq \|\mathcal{M}(\tilde{\mathbf{x}}_{oG})\|_0. \tag{46}$$

Proof: For simplicity, we rewrite the objective function of Eq. (39) as the following form,

$$\|\tilde{\mathbf{x}}\|_1 + \frac{\tau\beta_2}{2}\|\tilde{\mathbf{x}} - \mathbf{y}\|_2^2. \tag{47}$$

It is well known that the condition for $\tilde{\mathbf{x}}_{oT}$ being one of the optimal solutions of Eq. (47) is

$$\mathbf{0} \in \mathrm{SIGN}(\tilde{\mathbf{x}}_{oT}) + \tau\beta_2(\tilde{\mathbf{x}}_{oT} - \mathbf{y}), \tag{48}$$

where $\mathbf{0}$ is the zero vector in $\mathbb{R}^{2N}$. The function $\mathrm{SIGN}(\tilde{\mathbf{x}}_{oT})$ is the subgradient function of the l_1-norm $\|\tilde{\mathbf{x}}_{oT}\|_1$, where $\tilde{\mathbf{x}}_{oT} \in \mathbb{R}^{2N}$. It is defined elementwise with respect to each entry of the vector $\tilde{\mathbf{x}}_{oT}$:

$$\mathrm{SIGN}(\tilde{\mathbf{x}}_{oT}(i)) \begin{cases} = 1 & \tilde{\mathbf{x}}_{oT}(i) > 0, \\ \in [-1, 1] & \tilde{\mathbf{x}}_{oT}(i) = 0, \\ = -1 & \tilde{\mathbf{x}}_{oT}(i) < 0. \end{cases} \tag{49}$$

Substituting Eq. (49) into Eq. (48) yields

$$\tau\beta_2(\tilde{\mathbf{x}}_{oT}(i) - \mathbf{y}(i)) \begin{cases} = -1 & \tilde{\mathbf{x}}_{oT}(i) > 0, \\ \in [-1, 1] & \tilde{\mathbf{x}}_{oT}(i) = 0, \\ = 1 & \tilde{\mathbf{x}}_{oT}(i) < 0. \end{cases} \tag{50}$$

Therefore we have that

$$\tilde{\mathbf{x}}_{oT}(i) = \begin{cases} \mathbf{y}(i) - \frac{1}{\tau\beta_2} & \mathbf{y}(i) > \frac{1}{\tau\beta_2}, \\ 0 & \mathbf{y}(i) \in [-\frac{1}{\tau\beta_2}, \frac{1}{\tau\beta_2}], \\ \mathbf{y}(i) + \frac{1}{\tau\beta_2} & \mathbf{y}(i) < -\frac{1}{\tau\beta_2}. \end{cases} \tag{51}$$

which is the same as Eq. (40). We can observe that the soft-thresholding operator works by suppressing the entries within the region bounded by the positive and negative thresholds, $[-\frac{1}{\tau\beta_2}, \frac{1}{\tau\beta_2}]$, to zero and using a linear mapping for the other areas, i.e. the magnitude of $\mathbf{y}(i)$ is reduced by an amount equal to $\frac{1}{\tau\beta_2}$. Thus, as shown in Fig. 2, the two-dimensional nulling region for the FISTA is a square with sides of length $\frac{2}{\tau\beta_2}$. The GSO algorithm, on the other hand, compares each pair of $\mathbf{y}$, i.e. $\sqrt{\mathbf{y}(i)^2 + \mathbf{y}(i+N)^2}$ to the threshold. Consequently, its two-dimensional nulling region is a circle with radius $\frac{1}{\tau\beta_2}$. Clearly, $\sqrt{\mathbf{y}(i)^2 + \mathbf{y}(i+N)^2} \in [0, \frac{1}{\tau\beta_2}]$ implies $|\mathbf{y}(i)| \in [0, \frac{1}{\tau\beta_2}]$ and $|\mathbf{y}(i+N)| \in [0, \frac{1}{\tau\beta_2}]$, whereas the converse does not hold (put another way, the nulling region of the GSO is circumscribed by that of the FISTA). This means that an antenna that is nulled by the GSO will also be discarded FISTA, but an antenna discarded by FISTA may be kept by the GSO algorithm. Therefore, for the same threshold value $\frac{1}{\tau\beta_2}$, $\|\mathcal{M}(\tilde{\mathbf{x}}_{oF})\|_0 \leq \|\mathcal{M}(\tilde{\mathbf{x}}_{oG})\|_0$. This shows that the FISTA can generate a sparser solution than the GSO and prevent the purely real or imaginary solution from happening, in contrast to the Baysian Inference Solver of [29] that needs to utilise the multi-task strategy to address this problem.

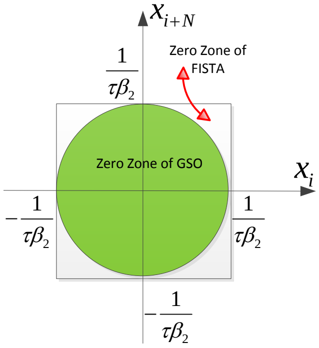


Fig. 2. The suppression to zero (nulling) regions for the FISTA and the GSO under the same threshold. The FISTA has a square nulling region whereas that of the GSO is a circle internally tangent to the square.

H. Amplitude Sparse Optimization

In this section, we propose another method, called Amplitude Sparse Optimization (ASO), which imposes the sparsity constraint on the amplitude of the excitation vector $\mathbf{x}$. We decompose the complex excitation $\mathbf{x}$ into two parts: the amplitude and the phase, i.e. $\mathbf{x} = \mathbf{X}_{\mathrm{a}}\mathbf{x}_{\mathrm{p}} = \mathbf{X}_{\mathrm{p}}\mathbf{x}_{\mathrm{a}}$, where $\mathbf{X}_{\mathrm{a}} = \mathrm{diag}(\mathbf{x}_{\mathrm{a}})$ and $\mathbf{X}_{\mathrm{p}} = \mathrm{diag}(\mathbf{x}_{\mathrm{p}})$. Now the problem is formulated as

$$\min_{\mathbf{x}_{\mathrm{a}},\mathbf{x}_{\mathrm{p}}} \mathbf{1}^T\mathbf{x}_{\mathrm{a}} + \frac{\beta_2}{2}\|\mathbf{A}\mathbf{X}_{\mathrm{a}}\mathbf{x}_{\mathrm{p}} - \mathbf{f}_{\mathrm{d}}\|_{\mathbf{W}}^2, \tag{52}$$

where $\mathbf{1}$ is the vector with all entries 1. In this method, we assume the desired beampattern $\mathbf{f}_{\mathrm{d}}$ is real. Firstly, we minimize the objective function in Eq. (52) with respect to $\mathbf{x}_{\mathrm{p}}$. Setting the first derivative with respect to $\mathbf{x}_{\mathrm{p}}^H$ to zero yields

$$\mathbf{x}_{\mathrm{p}} = \mathbf{X}_{\mathrm{a}}^{-1}(\mathbf{A}^H\mathbf{W}\mathbf{A})^{-1}(\mathbf{A}^H\mathbf{W})\mathbf{f}_{\mathrm{d}}. \tag{53}$$

After normalisation, we obtain

$$\mathbf{x}_{\mathrm{p}} = [(\mathbf{A}^H\mathbf{W}\mathbf{A})^{-1}(\mathbf{A}^H\mathbf{W})\mathbf{f}_{\mathrm{d}}] \oslash |(\mathbf{A}^H\mathbf{W}\mathbf{A})^{-1}(\mathbf{A}^H\mathbf{W})\mathbf{f}_{\mathrm{d}}|. \tag{54}$$

Next we minimize the objective function in Eq. (52) with respect to $\mathbf{x}_{\mathrm{a}}$, which can be rewritten as,

$$\begin{aligned} & \mathbf{1}^T\mathbf{x}_{\mathrm{a}} + \frac{\beta_2}{2}\|\mathbf{A}\mathbf{X}_{\mathrm{p}}\mathbf{x}_{\mathrm{a}} - \mathbf{f}_{\mathrm{d}}\|_{\mathbf{W}}^2 \\ = \; & \mathbf{1}^T\mathbf{x}_{\mathrm{a}} + \frac{\beta_2}{2}(\mathbf{A}\mathbf{X}_{\mathrm{p}}\mathbf{x}_{\mathrm{a}} - \mathbf{f}_{\mathrm{d}})^H\mathbf{W}(\mathbf{A}\mathbf{X}_{\mathrm{p}}\mathbf{x}_{\mathrm{a}} - \mathbf{f}_{\mathrm{d}}) \\ = \; & \mathbf{1}^T\mathbf{x}_{\mathrm{a}} + \frac{\beta_2}{2}\mathbf{x}_{\mathrm{a}}^T\mathcal{R}\{\mathbf{X}_{\mathrm{p}}^H\mathbf{A}^H\mathbf{W}\mathbf{A}\mathbf{X}_{\mathrm{p}}\}\mathbf{x}_{\mathrm{a}} \\ & -\beta_2\mathbf{x}_{\mathrm{a}}^T\mathcal{R}\{\mathbf{X}_{\mathrm{p}}^H\mathbf{A}^H\mathbf{W}\mathbf{f}_{\mathrm{d}}\} + \frac{\beta_2}{2}\mathbf{f}_{\mathrm{d}}^T\mathbf{W}\mathbf{f}_{\mathrm{d}}. \end{aligned} \tag{55}$$

Setting the derivative with respect to $\mathbf{x}_{\mathrm{a}}$ to zero and noting that $\mathbf{x}_{\mathrm{a}} \geq 0$, we find that

$$\begin{aligned} \mathbf{x}_{\mathrm{a}} \; = \; & \max\Big\{\mathcal{R}\{\mathbf{X}_{\mathrm{p}}^H\mathbf{A}^H\mathbf{W}\mathbf{A}\mathbf{X}_{\mathrm{p}}\}^{-1} \\ & \Big[\mathcal{R}\{\mathbf{X}_{\mathrm{p}}^H\mathbf{A}^H\mathbf{W}\mathbf{f}_{\mathrm{d}}\} - \frac{1}{\beta_2}\mathbf{1}\Big], \mathbf{0}\Big\}. \end{aligned} \tag{56}$$

Now we summarise the implementation procedure of the proposed ASO method as follows:

- **Initialisation:** Initialise $\mathbf{w}(i) = 1/\mathbf{f}_{\mathrm{d}}(i)$, the trade-off parameter β_2 as well as k_m, k_s, k_n.

- **Loop:** *while* the desired beampattern has not been achieved *do*
    1) Calculate the phase of the excitation vector $\mathbf{x}$ according to Eq. (54);
    2) Calculate the amplitude of the excitation vector $\mathbf{x}$ according to Eq. (56);
    3) Obtain the complex excitation vector by $\mathbf{x} = \mathbf{X}_{\mathrm{a}}\mathbf{x}_{\mathrm{p}}$;
    4) Update the weight $\mathbf{w}$ according to Eqs. (23) - (25);
    5) According to the sparsity of selected subarray, adjust β_2 utilising the continuation strategy;
    6) go back to Step 1) if stopping criterion is not met, otherwise terminate.

  End Loop.

In this method, the preservation of the maximum aperture length can be realised by changing $\mathbf{1}^T\mathbf{x}_{\mathrm{a}}$ to $\mathbf{t}^T\mathbf{x}_{\mathrm{a}}$, where smaller entries of $\mathbf{t}$ imply that the corresponding antennas have a higher likelihood of being selected. It should be noted that the assumption of real beampattern in the ASO algorithm results in a slightly worse performance compared to the other two methods, a fact that is borne by the simulation results.

Now we will summarise the merits and disadvantages of the proposed three methods for beampattern synthesis: (1) It is hard for the ASO algorithm to synthesize large antenna arrays due to the matrix inversion involved in the formulas. Moreover, since $\mathbf{A}^H\mathbf{W}\mathbf{A}$ may become singular due to the high coherency of the steering matrix $\mathbf{A}$, the ASO algorithm may not work under some scenarios. However, the ASO algorithm converges very fast when synthesizing small antenna arrays due to its simple iteration structure. (2) Because of the transformation from the group sparsity promoting variable $\mathbf{z}$ to $\tilde{\mathbf{x}}$ in Eq. (28), the GSO algorithm focuses on the least squared error more than the solution sparsity. Thus it is good at controlling the mainlobe ripple but produces less sparse arrays especially when the least squared solution does not have an approximate sparse structure. (3) The FISTA method can be used to synthesize large arrays with sparse solutions, but with more computational time compared to the GSO and ASO methods.

III. Simulation Results

In this section, we present extensive numerical results to validate the effectiveness and reliability of the proposed methods. We give a number of examples based on representative reference patterns, including both focused and shaped beampatterns, as well as arbitrary initial array layouts. We also compare their performance to the reweighted l_1-norm method described in [8] and the iterative FFT algorithm in [15]. The computational time, number of selected antennas, Dynamic Range Ratio (DRR) and performance of these examples are summarised in Table II. Since we are aiming at synthesizing receiver array beampattern, we do not place any explicit constraints on the excitation weights. The computer used for simulation has an Intel-i5 CPU and 8GB RAM.

A. Focused Beampattern Synthesis

1) Example 1: In the first example, we synthesize a nonsymmetric focused beampattern as shown in [12]. The mainlobe width is $|u| = 0.12$ with the left and right sidelobe level being -20dB and -30dB respectively. The synthesized beampattern is shown in Fig. 3. The 22-antenna positions are same as those in [12]. The synthesis speeds of both the FISTA and the GSO algorithms are an order of 10 faster than the l_1-norm method, while the ASO is an order of 100 faster. Both the FISTA and the GSO algorithms can maintain a -24dB left sidelobe level with preserved array directivity. The complex antenna excitations for the FISTA are shown in Table I.

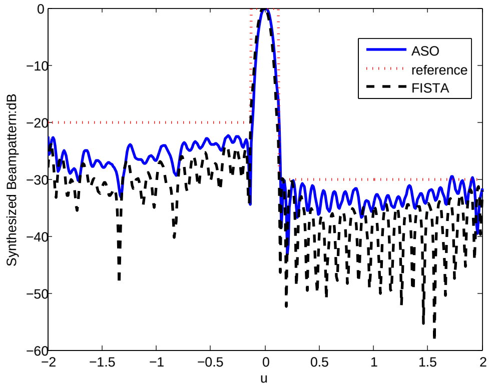


Fig. 3. Focused beampattern of Example 1 with mainlobe width $|u| = 0.12$, sidelobel level -20dB and -30dB. The beampattern of the GSO is similar to the FISTA and is omitted here for clarity.

TABLE I
Complex Excitation Weights Obtained by the FISTA method for Example 1.

Element No.	Position (λ)	Amplitude	Phase (rad)
1	0	0.3479	0.6421
2	0.34	0.4089	1.4605
3	0.80	0.4486	1.4876
4	1.27	0.5137	1.454
5	1.74	0.6071	1.4385
6	2.20	0.7003	1.4961
7	2.67	0.7817	1.4876
8	3.14	0.8717	1.4774
9	3.60	0.9415	1.5365
10	4.07	0.9814	1.5291
11	4.53	1	1.5824
12	5.00	0.9796	1.5777
13	5.47	0.9617	1.5708
14	5.93	0.9126	1.6332
15	6.40	0.8382	1.6277
16	6.86	0.7461	1.6925
17	7.33	0.6376	1.6884
18	7.80	0.5429	1.6918
19	8.26	0.4457	1.7492
20	8.73	0.35	1.729
21	9.20	0.264	1.6907
22	9.66	0.2204	1.9198

2) Example 2: In the second example, We compare our method with iterative FFT based algorithm [15] for large array synthesis problems that CVX cannot handle. The array aperture length is 180λ and [15] has 326 randomly placed antennas with inter-element spacing greater than $\lambda/4$. Since arbitrarily placed antennas are not practical in real applications, we apply array thinning to a 721-antenna uniform linear array with inter-element spacing $\lambda/4$. The 3dB mainlobe width and sidelobe level given in [15] are $[-0.36°, 0.36°]$ and -24.6dB respectively. Since there is a stringent requirement on the

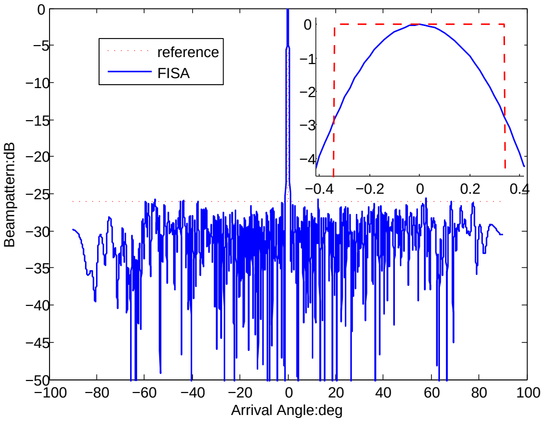


Fig. 4. Focused Beampattern of Example 2: the sidelobe level is -26dB, the mainlobe width is $[-0.34, 0.34]$ degrees.

mainlobe width, the maximum aperture length should be used. Therefore, instead of using the scalar trade-off parameter β_2 in Eq. (6), we employ a vector $\boldsymbol{\beta_2}$ with the first and last entries set much larger than the others (with values 100 versus 0.05). The beampattern that is obtained from the proposed FISTA method is shown in Fig. 4. Neither the ASO nor the GSO method work well in this example, where the least squared solution does not have an approximate sparse structure. We can see that the sidelobe level of the FISTA method is -26dB and the mainlobe width $[-0.34°, 0.34°]$. Both of these are better than the results obtained by the FFT method. The cost of this, however, is that the computational time of our method is 16.18 seconds, which is slower than the 6 seconds reported in [15] with an Intel i5-Core and 4GB RAM. The computational time of the FFT method may be less than 6 seconds if run on our computer, but it is important to point out that their method is fast only when best values of the parameters are chosen. The search for right parameters requires many trials which will dramatically increase the computational load. The number of selected antennas is 325 for the FISTA, one antenna fewer than the arbitrarily placed antennas in [15].

B. *Shaped Beampattern Synthesis*

1) Example 3: In the third example, we synthesise a non-uniform 41-antenna linear array as was done in [8]. This is a symmetrical array with an aperture length of 20 wavelengths. The antenna positions are shown in [30]. The mainlobe width and ripple are 40° and -0.4455dB respectively and the sidelobe level is -30dB in [8] with 31 selected antennas. The ASO gives the worst performance with -0.5dB mainlobe ripple and 35 selected antennas. The GSO presents the lowest sidelobe level -31.23dB and -0.4dB mainlobe ripple with 30 selected antennas. The proposed FISTA method has the smallest mainlobe ripple -0.3845dB with only 24 selected antennas. Thus in this example, the FISTA produces a much sparser array than other methods while still fulfilling the requirements. For the computational time, the ASO method is the fastest. The FISTA is slower than the ASO and a little faster than the GSO, but much faster than the reweighted l_1-norm method. The selected antenna positions and corresponding normalized excitation amplitude are shown in Fig. 6.

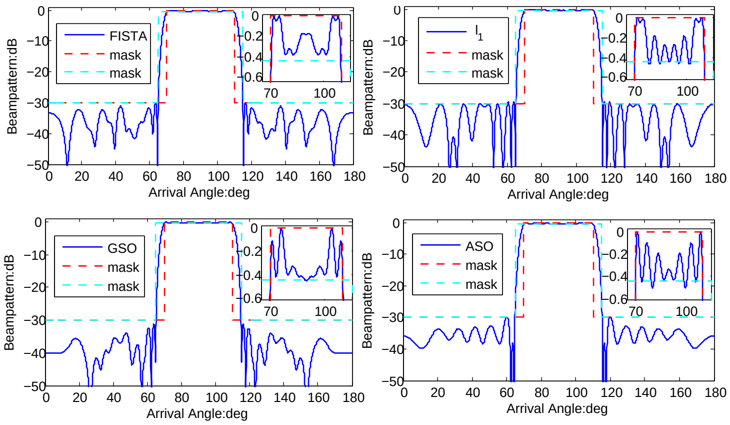


Fig. 5. Flap-top beampattern of Example 3.

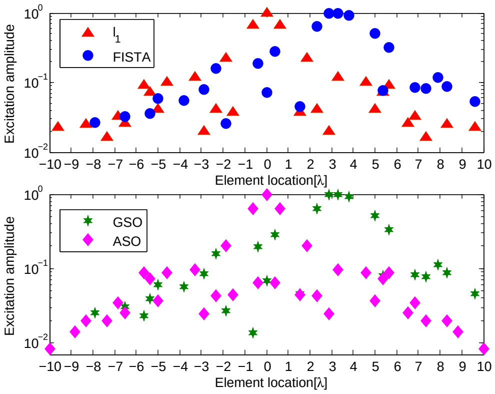


Fig. 6. Antenna positions and corresponding normalized excitation amplitude of Example 3.

2) Example 4: In order to show the proposed method can deal with arbitrary shaped beampatterns, we synthesise in the fourth example a flat-top pattern with a notch, again using the same non-uniform 41-antenna linear array. The synthesised beampattern is shown in Fig. 7. The sidelobe level is -40dB and the notch depth is -60dB. We can see that all four methods used select 37 antennas. Note that the GSO gives the smallest mainlobe ripple of -0.145dB and very fast convergence rate, while the FISTA method has the deepest null depth of -63.32dB with -0.309dB mainlobe ripple. The computational time of the FISTA is still half of the reweighted l_1-norm. The ASO method is again the fastest, but with unsatisfactory null depth. The reweighted l_1-norm exhibits the highest mainlobe ripple and the slowest convergence rate.

3) Example 5: In the fifth example, we again use the same non-uniform 41-antenna linear array. The desired beampattern has two flat-top mainlobes around 90° and 125° respectively with a sidelobe level of -30dB as shown in Fig. 8. We can see that the beampattern of the ASO method has the largest mainlobe ripple, around -1.972dB. The GSO method shows the best performance, with only -0.597dB mainlobe ripple and 35 selected antennas. The proposed FISTA method also selects 35 antennas, but with -0.8938dB mainlobe ripple similar to the

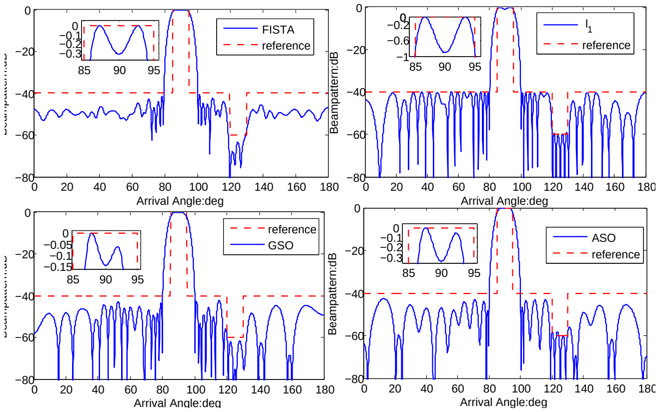


Fig. 7. Flap-top beampattern with notch of Example 4.

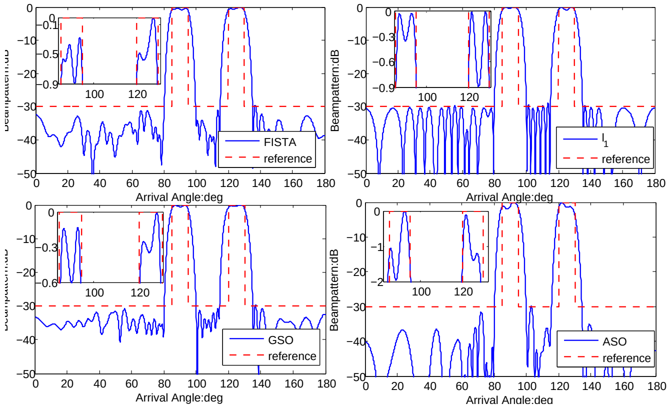


Fig. 8. Two flap-top beams of Example 5.

reweighted l_1-norm. Looking at the computational time, the ASO is always the fastest in the above three examples, while the reweighted l_1-norm method is the slowest. Except for the second example, the FISTA is slightly faster than the GSO and saves approximately half the computational time compared with the reweighted l_1-norm method.

4) Example 6: Finally, a quarter wavelength spaced rectangular planar array is considered. We employ the same 21×21 antenna array that was used in [8]. The required mainlobe region is $(u_x^2 + u_y^2) \leq 0.2^2$, and the sidelobe region is $u_x^2 + u_y^2 \geq 0.4^2$. The synthesised beampattern reported in [8] has a mainlobe ripple of -1.69dB and sidelobe level of -25.85dB as shown in Fig. 8 of that reference. The resulting beampattern using the proposed FISTA method is shown in Fig. 9 and the contour plot of the synthesized beampattern is shown in Fig. 10. Although the mainlobe ripple is slightly higher with -1.72dB, the sidelobe level is maintained by FISTA under -26.68dB. The selected subarray is shown in Fig. 11 which has similar configuration to the selected subarray in [8]. Both comprise of an inner circle and an outer region that is a square here and circle in [8]. Although the GSO method can also be utilised here and its mainlobe ripple is the smallest with -0.7751dB, its sidelobe level is only -23.32dB with 165 selected antennas. The computational time in this example is nearly same for both the FISTA and the GSO, and both are much faster than the reweighted l_1-norm method.

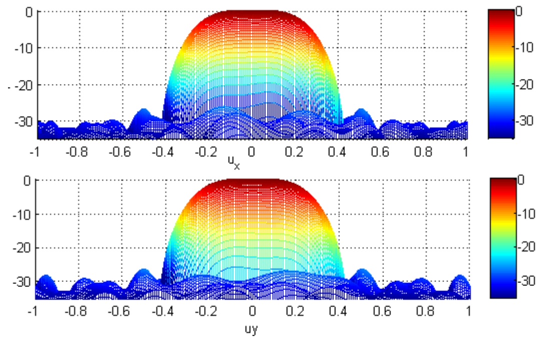


Fig. 9. Synthesized beampattern of a 21×21 planar array, the mainlobe region is $u_x^2 + u_y^s \leq 0.04$, the sidelobe level is -26.68 dB.

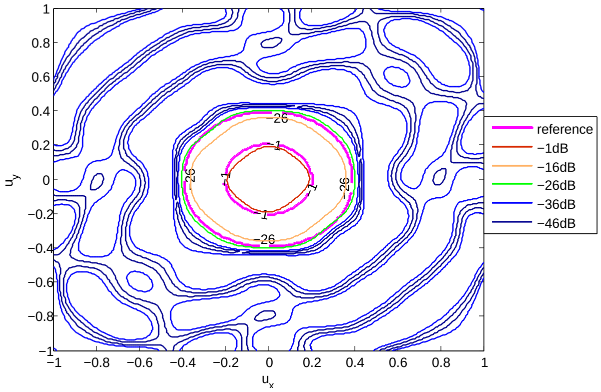


Fig. 10. Contour plot of the planar array beampattern.

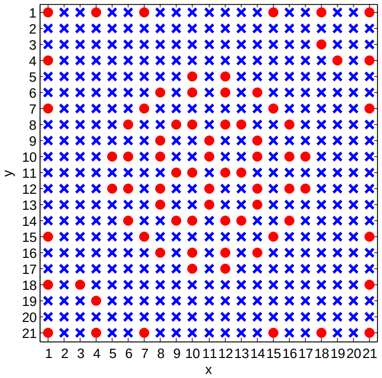


Fig. 11. 76-antenna subarray selected in a 21×21 planar array, with circle being selected and cross being abandoned.

IV. Conclusion

Three soft-thresholding based optimization methods are proposed in this paper to synthesise thinned arrays with arbitrary shaped beampatterns including multi-beam forming and several angular regions of suppression. The proposed methods are highly flexible, easily reconfigurable and can handle arbitrarily shaped arrays. They exhibit significant improvement with respect to convex optimization in regard to the mainlobe ripple control, sidelobe level reduction and also the computational time. Moreover, the FISTA can be utilised

TABLE II
THE COMPUTATIONAL TIME, SPARSITY, DRR AND PERFORMANCE OF THE FISTA, THE GSO, THE ASO AND THE REWEIGHTED l_1-NORM METHOD.

	Focused Beampattern						
Exam	Method	Time	Antenna	Main	Side	Null	DRR
No.	Name	(sec)	Number	(u/deg)	(dB)	(dB)	1
1	l_1-norm	3.05	22	±0.12 [2]	-30,-20	-	no [1]
	FISTA	0.57	22	±0.12 [2]	-30,-24	-	4.54
	ASO	0.013	22	±0.12 [2]	-30,-22.5	-	8.64
	GSO	0.17	22	±0.12 [2]	-30,-24	-	4.63
2	l_1-norm	–	–	–	–	-	–
	FFT [3]	6.1	326	$\pm0.36°$	-24.6	-	no [1]
	FISTA	16.18	325	$\pm0.34°$	-26	-	5.43
	ASO	–	–	–	–	-	–
	GSO	–	–	–	–	-	–
	Shaped Beampattern						
Exam	Method	Time	Antenna	Main	Side	Null	DRR
No.	Name	(sec)	Number	(dB)	(dB)	(dB)	1
3	l_1-norm	2.08	31	-0.4455	-30	-	61.35
	FISTA	0.89	24	-0.3845	-30	-	39.9
	ASO	0.023	35	-0.5	-30	-	119.05
	GSO	1.32	26	-0.4	-31.23	-	81.18
4	l_1-norm	2.05	37	-0.91	-40	-60	140.85
	FISTA	1.1	37	-0.309	-40.8	-63.32	157.14
	ASO	0.01	37	-0.3448	-42.61	-58.12	184.62
	GSO	0.043	37	-0.145	-42.4	-61.67	157.14
5	l_1-norm	2.28	39	-0.8965	-30	-	68.03
	FISTA	1.33	35	-0.8938	-30	-	35.46
	ASO	0.01	37	-1.972	-30	-	50.76
	GSO	1.43	35	-0.597	-30	-	34.36
6	l_1-norm	146.72	76	-1.69	-25.85	-	34.22
	FISTA	119.05	76	-1.72	-26.68	-	11.45
	ASO	–	–	–	–	-	–
	GSO	112.48	165	-0.7751	-23.32	-	19.92

[1] Papers [15] and [12] do not report the DRR value;
[2] Paper [12] and its reference papers define the mainlobe width in terms of $u = sin\theta$, i.e. $|u| = 0.12$;
[3] Simulation results for the iterative FFT algorithm are only given for example 2.

to synthesize large scaled antenna array that the CVX cannot handle. Therefore, the proposed FISTA based array thinning method can be utilised in applications requiring a fast method to adaptively reconfigure large antenna arrays with specified beampattern requirements.

ACKNOWLEDGMENT

The authors would like to thank Dr. Yannis Kopsinis with the Department of Informatics and Telecommunications, University of Athens, Greece to give helpful suggestions for improving the paper's quality.

REFERENCES

[1] G. Toso and R. Mailloux, "Guest editorial for the special issue on innovative phased array antennas based on non-regular lattices and overlapped subarrays," *Antennas and Propagation, IEEE Transactions on*, vol. 62, pp. 1546–1548, April 2014.

[2] X.-K. Wang, Y.-C. Jiao, and Y.-Y. Tan, "Synthesis of large thinned planar arrays using a modified iterative fourier technique," *Antennas and Propagation, IEEE Transactions on*, vol. 62, pp. 1564–1571, April 2014.

[3] J. Araque Quijano, M. Righero, and G. Vecchi, "Sparse 2-d array placement for arbitrary pattern mask and with excitation constraints: A simple deterministic approach," *Antennas and Propagation, IEEE Transactions on*, vol. 62, pp. 1652–1662, April 2014.

[4] M. Amin, "Concurrent nulling and locations of multiple interferences in adaptive antenna arrays," *Signal Processing, IEEE Transactions on*, vol. 40, pp. 2658–2668, Nov 1992.

[5] B. Friedlander and B. Porat, "Performance analysis of a null-steering algorithm based on direction-of-arrival estimation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, pp. 461–466, April 1989.

[6] X. Wang, E. Aboutanios, M. Trinkle, and M. Amin, "Reconfigurable adaptive array beamforming by antenna selection," *Signal Processing, IEEE Transactions on*, vol. 62, pp. 2385–2396, May 2014.

[7] X. Wang and E. Aboutanios, "Reconfigurable adaptive linear array signal processing in GNSS applications," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 4154–4158, May 2013.

[8] S. Nai, W. Ser, Z. Yu, and H. Chen, "Beampattern synthesis for linear and planar arrays with antenna selection by convex optimization," *Antennas and Propagation, IEEE Transactions on*, vol. 58, no. 12, pp. 3923–3930, 2010.

[9] A. Massa, P. Rocca, and R. Haupt, "Interference suppression in uniform linear arrays through a dynamic thinning strategy," *Antennas and Propagation, IEEE Transactions on*, vol. 59, no. 12, pp. 4525–4533, 2011.

[10] L. Poli, P. Rocca, M. Salucci, and A. Massa, "Reconfigurable thinning for the adaptive control of linear arrays," *Antennas and Propagation, IEEE Transactions on*, vol. 61, pp. 5068–5077, Oct 2013.

[11] L. Cen, Z. Yu, W. Ser, and W. Cen, "Linear aperiodic array synthesis using an improved genetic algorithm," *Antennas and Propagation, IEEE Transactions on*, vol. 60, no. 2, pp. 895–902, 2012.

[12] B. Fuchs, "Synthesis of sparse arrays with focused or shaped beampattern via sequential convex optimizations," *Antennas and Propagation, IEEE Transactions on*, vol. 60, no. 7, pp. 3499–3503, 2012.

[13] G. Oliveri, P. Rocca, and A. Massa, "Reliable diagnosis of large linear arraysa bayesian compressive sensing approach," *Antennas and Propagation, IEEE Transactions on*, vol. 60, no. 10, pp. 4627 –4636, 2012.

[14] Y. Liu, Z. Nie, and Q. Liu, "Reducing the number of elements in a linear antenna array by the matrix pencil method," *Antennas and Propagation,*

IEEE Transactions on, vol. 56, no. 9, pp. 2955–2962, 2008.

[15] K. Yang, Z. Zhao, and Q. H. Liu, "Fast pencil beam pattern synthesis of large unequally spaced antenna arrays," *Antennas and Propagation, IEEE Transactions on*, vol. 61, pp. 627 –634, feb 2013.

[16] W. Du Plessis, "Weighted thinned linear array design with the iterative fft technique," *Antennas and Propagation, IEEE Transactions on*, vol. 59, no. 9, pp. 3473–3477, 2011.

[17] W. Keizer, "Linear array thinning using iterative fft techniques," *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 8, pp. 2757–2760, 2008.

[18] T. Isernia, P. Di Iorio, and F. Soldovieri, "An effective approach for the optimal focusing of array fields subject to arbitrary upper bounds," *Antennas and Propagation, IEEE Transactions on*, vol. 48, pp. 1837–1847, Dec 2000.

[19] A. Capozzoli, C. Curcio, A. Liseno, and G. Toso, "Phase-only synthesis of flat aperiodic reflectarrays," *Progress In Electromagnetics Research*, vol. 133, 2013.

[20] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010.

[21] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[22] E. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing," *CAAM TR07-07, Rice University*, 2007.

[23] Z. Han, H. Li, and W. Yin, *Compressive Sensing for Wireless Networks*. Cambridge University Press, 2013.

[24] X. Wang and E. Aboutanios, "Adaptive beampattern synthesis with antenna selection by iterative shrinkage continuation method," in *the 21st European Signal Processing Conference (Eusipco 2013)*, EURASIP, 2013.

[25] E. Treister and I. Yavneh, "A multilevel iterated-shrinkage approach to l 1 penalized least-squares minimization," *Signal Processing, IEEE Transactions on*, vol. 60, no. 12, pp. 6319–6329, 2012.

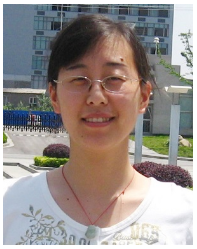
[26] Z. Shi and Z. Feng, "A new array pattern synthesis algorithm using the two-step least-squares method," *Signal Processing Letters, IEEE*, vol. 12, no. 3, pp. 250–253, 2005.

[27] P. Zhou and M. Ingram, "Pattern synthesis for arbitrary arrays using an adaptive array method," *Antennas and Propagation, IEEE Transactions on*, vol. 47, no. 5, pp. 862–869, 1999.

[28] W. Deng, W. Yin, and Y. Zhang, "Group sparse optimization by alternating direction method," *TR11-06, Department of Computational and Applied Mathematics, Rice University*, 2011.

[29] G. Oliveri, M. Carlin, and A. Massa, "Complex-weight sparse linear array synthesis by bayesian compressive sampling," *Antennas and Propagation, IEEE Transactions on*, vol. 60, no. 5, pp. 2309–2326, 2012.

[30] F. Wang, V. Balakrishnan, P. Zhou, J. Chen, R. Yang, and C. Frank, "Optimal array pattern synthesis using semidefinite programming," *Signal Processing, IEEE Transactions on*, vol. 51, no. 5, pp. 1172–1183, 2003.



Xiangrong Wang received both the B.S. degree and the M.S. degree in electrical engineering from Nanjing University of Science and Technology, China, in 2009 and 2011, respectively. She is working toward the Ph.D. degree in electrical engineering in University of New South Wales, Sydney, Australia. She was a visiting research student in the Centre for Advanced Communications, Villanova University, USA from Feb to Oct in 2014. Her research interest includes adaptive array processing, beampattern synthesis, DOA estimation and convex optimization.

Elias Aboutanios (SM'11) received a Bachelor in Engineering in 1997 from UNSW and the PhD degree in 2003 from the University of Technology, Sydney (UTS). While at UTS he was a member of the CRC for Satellite Systems, working on the development of the Ka-Band Earth station. From 2003 to 2007, he was a research fellow with the Institute for Digital Communications at the University of Edinburgh where he conducted research on Space Time Adaptive Processing for radar target detection. He is currently a senior lecturer at UNSW. In 2011, he secured over $1M in funding and led an international consortium to develop Australia's first masters in Satellite Systems Engineering. The project was completed and the masters launched in June 2013. In 2012 he set up the UNSW contribution to the QB50 project and is the project leader. Dr Aboutanios received the best oral presentation award at the 2010 CISP, and the Faculty of Engineering Teaching Excellence Award in 2011 for developing the novel Design Proficiency subject that he teaches to final year undergraduates. He is the UNSW IEEE student branch counsellor and the BLUEsat space society mentor. His research interests include parameter estimation, algorithm optimisation and analysis, adaptive and statistical signal processing and their application in the contexts of radar, GNSS, Nuclear Magnetic Resonance, and smart grids. He is the joint holder of a patent on frequency estimation.

Moeness G. Amin (F'01) received his Ph.D. degree in Electrical Engineering from University of Colorado in 1984. He has been on the Faculty of the Department of Electrical and Computer Engineering at Villanova University since 1985. In 2002, he became the Director of the Center for Advanced Communications, College of Engineering. He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE), 2001; Fellow of the International Society of Optical Engineering, 2007; and a Fellow of the Institute of Engineering and Technology (IET), 2010. Dr. Amin is a Recipient of the IEEE Third Millennium Medal, 2000; Recipient of the 2009 Individual Technical Achievement Award from the European Association of Signal Processing; Recipient of the 2010 NATO Scientific Achievement Award; Recipient of the Chief of Naval Research Challenge Award, 2010; Recipient of Villanova University Outstanding Faculty Research Award, 1997; and the Recipient of the IEEE Philadelphia Section Award, 1997. He was a Distinguished Lecturer of the IEEE Signal Processing Society, 2003-2004, and is currently the Chair of the Electrical Cluster of the Franklin Institute Committee on Science and the Arts. Dr. Amin has over 600 journal and conference publications in the areas of Wireless Communications, Time-Frequency Analysis, Sensor Array Processing, Waveform Design and Diversity, Interference Cancellation in Broadband Communication Platforms, satellite Navigations, Target Localization and Tracking, Direction Finding, Channel Diversity and Equalization, Ultrasound Imaging and Radar Signal Processing. He co-authored 18 book chapters. He is the Editor of the book "Through the Wall Radar Imagin" and "Compressive Sensing for Urban Radar," published by CRC Press in 2011 and 2014, respectively.