Progress In Electromagnetics Research B, Vol. 29, 157–174, 2011

NEAR OPTIMAL ROBUST ADAPTIVE BEAMFORMING APPROACH BASED ON EVOLUTIONARY ALGORITHM

R. Mallipeddi

School of Electrical and Electronic Engineering
Nanyang Technological University
50 Nanyang Ave. Block S2, 639798, Singapore

J. P. Lie

Temasek Laboratories
Nanyang Technological University
9th Storey, BorderX Block, Research Techno Plaza
50 Nanyang Drive, 637553, Singapore

P. N. Suganthan

School of Electrical and Electronic Engineering
Nanyang Technological University
50 Nanyang Ave. Block S2, 639798, Singapore

S. G. Razul

Temasek Laboratories
Nanyang Technological University
9th Storey, BorderX Block, Research Techno Plaza, 50 Nanyang Drive, 637553, Singapore

C. M. S. See †

DSO National Laboratories
20 Science Park Drive, 118230, Singapore

Received 8 November 2010, Accepted 23 March 2011, Scheduled 29 March 2011

Corresponding author: Joni Polili Lie (jonilie@ntu.edu.sg).

† Also with Temasek Laboratories, Nanyang Technological University, 9th Storey, BorderX Block, Research Techno Plaza
50 Nanyang Drive, 637553, Singapore.

Abstract—The presence of desired signal in the training data for sample covariance matrix calculation is known to lead to a substantial performance degradation, especially when the desired signal is the dominant signal in the training data. Together with the uncertainty in the look direction, most of the adaptive beamforming solutions are unable to approach the optimal performance. In this paper, we propose an evolutionary algorithm (EA) based robust adaptive beamforming that is able to achieve near optimal performance. The essence of the idea is to shape the array beam response such that it has maximum response in the desired signal's angular range and minimum response in the interferences' angular range. In addition, the approach introduces null-response constraints deduced from the array observation to achieve better interference cancelation performance. As a whole, the proposed optimization is solvable using an improved variant of the differential evolution (DE) algorithm. Numerical simulations are also presented to demonstrate the efficacy of the proposed algorithm.

1. INTRODUCTION

Traditional adaptive beamforming techniques work effectively *only* under the assumption that precise knowledge of the desired signal steering vector is known [1–3]. Any violation or mismatch between the assumed (nominal) and the actual knowledge will cause a substantial degradation in their performance [4]. Several approaches to improve their robustness were proposed over the past decade [5–7]. However, many of the proposed methods are limited to certain type of mismatches.

These mismatches are the look-direction, gain-phase, array geometry as well as other mismatches due to incorrect assumption of the signal model, e.g., point-source or scattered-source (either coherent or incoherent scattering) signal model. Besides these mismatches, the performance of the beamformers is also known to degrade when the number of snapshots is small or any other effects introduced by the propagation environment.

In [4, 8], an approach based on loading of the diagonal of the sample array covariance matrix is proposed to improve the robustness against more general mismatches. While having the advantage of being invariant to the type of mismatches, the choice of the optimal loading factor is not obvious.

Recently, the authors in [9–13] uncertainty setting of the steering vector which allows the norm of the mismatch vector to be bounded. These approaches are modeled based on arbitrary mismatch and can be interpreted as Capon beamformer with optimal diagonal loading when the level of uncertainty due the mismatches is precisely known. It is also worth mentioning that other approaches reported in [14, 15] does not even need to specify the mismatch as the diagonal loading value will be calculated in the method.

Although most of the existing approaches are able to recover the performance degradation due to imprecise knowledge of the desired signal steering vector, the performance degradation due to the use of the sample covariance matrix received less attention. This is because the performance degradation is observed only when the input signal-to-noise ratio (SNR) is relatively high. In this paper, we propose an evolutionary algorithm (EA) based robust adaptive beamforming that is able to achieve near optimal performance at high SNR case. The principal motivation behind the proposed approach is to avoid the use of interference-plus-noise covariance because its formulation strictly requires the array observation to be free of the desired signal component. Given the capability of differential evolution (DE) method to solve non-convex optimization and the simplicity of its implementation, we propose to use DE to solve for a novel objective function formulated as the ratio between the interference-plus-noise and the desired signal beam pattern response, subjected to the constraint that nullifies the response at interferences' direction-of-arrival (DOA).

At high SNR case, the proposed approach performs consistently better than the state-of-the-art robust adaptive beam forming approaches based on our empirical study. However, the proposed method is a computationally expensive as compared to existing beam forming techniques. Evolutionary algorithms are inherently parallel and the computational complexity of the proposed approached can be overcome by using Graphical Processing Units (GPUs) in general and General-purpose GPUs (GPGPUs) [16–18] in particular as inexpensive arithmetic processing units. In [16], the authors show that DE can be executed 10-100 times faster on GPUs compared to CPUs depending upon the complexity of the problem. Such a speed-up would make the proposed robust adaptive beam forming realizable in real-time.

The rest of this paper is organized as follows. In Section 2, we describe the array signal model and the existing robust adaptive beamformers as well as some background on EA algorithms. Next in Section 3, we explain the proposed approach that includes the optimization formulation up to the implementation of the proposed DE algorithm in addition to estimating the interferences' DOA as a pre-processing step required for the optimization. Section 4 presents the simulation results that demonstrate the efficacy of the proposed solution. And finally, Section 5 concludes the paper.

2. BACKGROUND

2.1. Signal Model

Consider a narrowband beamforming model in which K narrowband plane wave signals, modeled as statistically independent zero-mean random sequence, impinge on an array of M sensors ($K < M$) from directions θ_s and θ_i ($i = 1, 2, \ldots, K-1$). The received signal at the array is given by

$$\mathbf{x}(t) = \mathbf{a}(\theta_s)s_d(t) + \sum_{i=1}^{K-1} \mathbf{a}(\theta_i)s_i(t) + \mathbf{n}(t) \tag{1}$$

where $s_d(t)$, $s_i(t)$ and $n(t)$ are the desired signal, i-th interference and noise, respectively. $\mathbf{a}(\theta)$ is the steering vector of the plane wave from direction θ.

Generally, the adaptive beamformer applies a weight vector to the array received signal in order to achieve the noise-reduced desired signal and cancel the interferences at the same time. Conventional adaptive beamforming calculates the optimal weight vector that minimizes the interference-plus-noise output power subjected to a unity response of the desired signal

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{in} \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a} = 1 \tag{2}$$

where $\mathbf{w} = [w_1, \ldots, w_M]^T$ is the complex vector of beamformer weights, M is the number of sensors, $\mathbf{R}_{in}$ is the interference-plus-noise covariance matrix, $\mathbf{a}$ is the desired signal steering vector and $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and Hermitian transpose, respectively. The optimal beamformer weight obtained by solving (2) can be expressed as

$$\mathbf{w}_{mvdr} = (\mathbf{a}^H \mathbf{R}_{in}^{-1} \mathbf{a})^{-1} \mathbf{R}_{in}^{-1} \mathbf{a} \tag{3}$$

In practice, the matrix $\mathbf{R}_{in}$ is estimated from the discrete-sampled array received signal according to

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}(n)\mathbf{x}(n)^H \tag{4}$$

where N is the number of snapshots. Because of the presence of the desired signal components $\mathbf{a}(\theta_s)s_d(t)$ in the array received signal for computing $\hat{\mathbf{R}}$, it will lead to a substantial performance degradation as measured by the output signal-to-interference-and-noise ratio (SINR). That is, the output SINR performance of the existing beamformers is unable to match the increasing rate of an ideal beamformer and eventually the output SINR flattens as signal-to-noise ratio (SNR) increases. This problem is further complicated when there is a mismatch between the actual and the presumed ASV (denoted as $\bar{\mathbf{a}}$).

2.2. Existing Robust Adaptive Beamformers

A simple-yet-effective approach for robust adaptive beamforming is the diagonal loading approach. As the name implies, the approach offers robustness by adding a positive value to the diagonal terms of the sample covariance matrix. The beamformer's weight is then formulated based on the loaded sampled covariance matrix $\mathbf{R}_{dl}$ according to

$$\mathbf{w}_{lsmi} = (\bar{\mathbf{a}}^H \mathbf{R}_{dl}^{-1} \bar{\mathbf{a}})^{-1} \mathbf{R}_{dl}^{-1} \bar{\mathbf{a}} \tag{5}$$

where $\bar{\mathbf{a}}$ denotes the *presumed* steering vector and $\mathbf{R}_{dl} \triangleq \gamma \mathbf{I} + \hat{\mathbf{R}}$ is the diagonally loaded sample covariance matrix, γ denotes the loading factor and $\mathbf{I}$ is the identity matrix. Such an approach is termed as the loaded sample matrix inverse (LSMI) beamformer.

Although it has been shown to improve the performance, it is not clear how much loading factor or what is the suitable value for γ is required. To explicitly relate the amount of loading factor to the uncertainties in the desired signal steering vector, the authors in [12] proposed a different optimization formulation for solving the beamformer's weight. The formulation is based on the following quadratic optimization problem with a multi-dimensional spherical constraint that models the uncertainty as the square-norm of the mismatch vector:

$$\min_{\mathbf{a}} \quad \mathbf{a}^H \hat{\mathbf{R}}^{-1} \mathbf{a} \qquad \text{subject to} \quad \|\mathbf{a} - \bar{\mathbf{a}}\|^2 \leq \varepsilon. \tag{6}$$

where ε quantifies the level of the uncertainty between the nominal and actual steering vector. By imposing a quadratic equality constraint on (1) and using the Lagrange multiplier method, the estimated steering vector is given by:

$$\hat{\mathbf{a}} = \bar{\mathbf{a}} - (\mathbf{I} + \lambda \hat{\mathbf{R}})^{-1} \bar{\mathbf{a}} \tag{7}$$

and the Lagrange multiplier λ is obtained by solving the following constraint equation:

$$g(\lambda) \triangleq \left\| (\mathbf{I} + \lambda \hat{\mathbf{R}})^{-1} \bar{\mathbf{a}} \right\|^2 = \varepsilon. \tag{8}$$

The estimated steering vector is later used to formulate the weight vector

$$\mathbf{w}_{rcb} = (\hat{\mathbf{a}}^H \hat{\mathbf{R}}^{-1} \hat{\mathbf{a}})^{-1} \hat{\mathbf{R}}^{-1} \hat{\mathbf{a}} \tag{9}$$

In the analysis presented in [12], this approach belongs to the diagonal loading method where the loading factor is a function of the Lagrange multiplier λ.

In another approach that attempts to avoid the need to know the loading factor, the authors in [15] showed that the robust adaptive beamforming can be formulated as a ridge regression problem. As a result, the design of the weight vector does not require any preset parameter even though the steering vector used is inaccurate. At the end of the formulation, the authors in [15] also show that this parameter-free robust beamformer is in fact a diagonal loading approach with the loading parameter

$$\begin{aligned} \rho &= (M-1)\hat{\sigma}_{LS}^2/\|\eta_{LS}\|^2 \\ \hat{\sigma}_{LS}^2 &= \|\hat{\mathbf{R}}^{1/2}\mathbf{B}\eta_{LS} - \hat{\mathbf{R}}^{1/2}\bar{\mathbf{a}}/M\|^2 \\ \eta_{LS} &= (\mathbf{B}^H\hat{\mathbf{B}}\mathbf{B})^{-1}\mathbf{B}^H\hat{\mathbf{R}}\bar{\mathbf{a}}/M \end{aligned} \tag{10}$$

where $\mathbf{B}$ is an $M \times (M-1)$ semi-unitary matrix orthogonal to the nominal steering vector $\bar{\mathbf{a}}$. The beamformer weight is the same as that in LSMI beamformer with the diagonally-loaded covariance matrix $\mathbf{R}_{rr} \triangleq \rho\mathbf{I} + \hat{\mathbf{R}}$:

$$\mathbf{w}_{rr} = (\bar{\mathbf{a}}^H\mathbf{R}_{rr}^{-1}\bar{\mathbf{a}})^{-1}\mathbf{R}_{rr}^{-1}\bar{\mathbf{a}} \tag{11}$$

2.3. Evolutionary Algorithms Background

Recently, population-based stochastic algorithms such as evolutionary algorithms (EAs), inspired by Darwinian Theory of evolution, are gaining importance compared to Classical Optimization techniques due to their ability to handle real-world optimization problems which are non continuous and/or non-differentiable and characterized by chaotic disturbances, randomness and complex non-linear dynamics. EAs start with a population of individuals, each encoding a potential solution to a given problem in a predefined search space. The individuals communicate and exchange information through natural processes like mutation, recombination and selection, to evolve increasingly fitter new individuals to a particular environment.

More recently, differential evolution (DE) method proposed by Storn and Price [19–21], a simple and powerful global optimization algorithm, has attracted much attention due to its simplicity and less number of parameters to tune [22–24]. DE perturbs the current-generation population members with a scaled difference of randomly selected and distinct population members.

The performance [25] of the DE algorithm is sensitive to the mutation strategy, crossover strategy and control parameters such as the population size (NP), crossover rate (CR) and the scale factor (F). The best settings for the control parameters can be different for different optimization problems and the same functions with different requirements for consumption time and accuracy. Therefore, to successfully solve a specific optimization problem, it is generally necessary to perform a time-consuming trial-and-error search for the most appropriate combination of strategies and their associated parameter values. However, such a trial-and-error search process suffers from high computational costs. The population of DE may evolve through different regions in the search space, within which different strategies [28] with different parameter settings may be more effective than others. Different partial adaptation schemes have been proposed [26, 27] to overcome the time consuming trial-and-error procedure. However, in [31] the authors propose a DE algorithm with ensemble approach (EPSDE) and demonstrated its superior performance.

2.4. Ensemble of Mutation Strategies and Parameters in DE (EPSDE)

The effectiveness of conventional DE in solving a numerical optimization problem depends on the selected mutation and crossover strategy and its associated parameter values. However, different optimization problems require different mutation strategies with different parameter values depending on the nature of problem (uni-modal and multi-modal) and available computation resources. In addition, to solve a specific problem, different mutation strategies with different parameter settings may be better during different stages of the evolution than a single mutation strategy with unique parameter settings as in the conventional DE. Motivated by these observations, we propose an ensemble of mutation and crossover strategies and parameter values for DE (EPSDE) in which a pool of mutation strategies, along with a pool of values corresponding to each associated parameter competes to produce successful offspring population. The candidate pool of mutation and mutation strategies and parameters should be restrictive to avoid the unfavorable influences of less effective mutation strategies and parameters [28]. The mutation strategies or the parameters present in a pool should have diverse characteristics, so that they can exhibit distinct performance characteristics during different stages of the evolution, when dealing with a particular problem.

EPSDE consists of a pool of mutation and crossover strategies along with a pool of values for each of the associated control parameters. Each member in the initial population is randomly assigned with a mutation strategy and associated parameter values taken from the respective pools. The population members (target vectors) produce offspring (trial vectors) using the assigned mutation strategy and parameter values. If the generated trial vector produced is better than the target vector, the mutation strategy and parameter values are retained with trial vector which becomes the parent (target vector) in the next generation. The combination of the mutation strategy and the parameter values that produced a better offspring than the parent are stored. If the target vector is better than the trial vector, then the target vector is randomly re-initialized with a new mutation strategy and associated parameter values from the respective pools or from the successful combinations stored with equal probability. This leads to an increased probability of production of offspring by the better combination of mutation strategy and the associated control parameters in the future generations.

The implementation of the EPSDE algorithm is presented in [31]. The outline of the algorithm is presented in Section 3.2.

3. PROPOSED APPROACH

In this section, we propose a different approach to solve for the robust adaptive beamforming problem. The idea pursued here is to design the robust beamformer's weight vector by minimizing the ratio of the noise-plus-interference to the desired signal response. In other words, the array beam response is designed in such a way that it has maximum response in the desired signal's angular range bounded by the inequality: $\theta_L < \theta_s < \theta_U$, where θ_L and θ_U are the lower and upper bounds respectively; and it has minimum response in the interferences' angular range. In mathematical expression, this ratio can be formulated as follows

$$f(\mathbf{w}) = \frac{\mathbf{w}^H \bar{\mathbf{A}}_{in} \bar{\mathbf{A}}_{in}^H \mathbf{w}}{\mathbf{w}^H \bar{\mathbf{A}}_s \bar{\mathbf{A}}_s^H \mathbf{w}} \tag{12}$$

where $\bar{\mathbf{A}}_s = [\bar{\mathbf{a}}(\theta_{s,1}), \bar{\mathbf{a}}(\theta_{s,1}), \ldots, \bar{\mathbf{a}}(\theta_{s,L_s})]$ and $\{\theta_{s,q}\}_{q=1}^{L_s}$ are the possible look-directions derived from the desired signal's angular range. Likewise, $\bar{\mathbf{A}}_{in} = [\bar{\mathbf{a}}(\theta_{in,1}), \bar{\mathbf{a}}(\theta_{in,1}), \ldots, \bar{\mathbf{a}}(\theta_{in,L_{in}})]$, where $\{\theta_{in,q}\}_{q=1}^{L_{in}}$ are derived from the interference-plus-noise angular range. Note that the directions defined in $\theta_{in,q}$ do not overlap with those in $\theta_{s,q}$.

Besides minimizing the function defined in (12), the proposed optimization design includes the null-response constraints deduced from the array observation $\mathbf{x}(t)$ to achieve better interference cancelation performance. These null-response constraints can be expressed mathematically as $h_i(\mathbf{w}) = 0$ for $i = \{1, 2, \ldots, K-1\}$ where $h_i(\mathbf{w})$ is defined as

$$h_i(\mathbf{w}) = \mathbf{w}^H \bar{\mathbf{a}}(\hat{\theta}_i) \tag{13}$$

3.1. Estimating Interferences' DOA

From the null-response constraints expression, we need to estimate the interferences' DOA. The key idea pursued here is to utilize the fact that the beamformer's response still forms nulls at the interferences' DOA regardless of the look direction mismatch. This is because the sensitivity of the Capon beamformer to the data model errors due to mismatches only affects the desired signal DOA and this phenomenon is widely known as signal self-nulling. As for the rest of the nulls formed in the beampattern, they are corresponding to the interferences' DOAs. Therefore, this property can be utilized for estimating the interferences' DOA from the rest of the nulls. Unless the desired signal's DOA and the interferences' ones are very closely separated, the estimation errors will not influence the proposed algorithm.

We know that the presumed steering vector $\bar{\mathbf{a}}(\theta_s)$ corresponding to the desired signal is known only approximately. Also, the presumed look direction θ_s contains mismatch but the exact DOA of the desired signal can be found within the range defined by $\theta_L < \theta_s < \theta_U$. Notice that in order to form $\bar{\mathbf{a}}(\theta_s)$ for the evaluation of the beamformer's response or beampattern, we also require the knowledge of the presumed array geometry. We start off with the Standard Capon Beamformer solution given as

$$\mathbf{w}_{\text{SCB}} = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta_s)}{\mathbf{a}^H(\theta_s)\mathbf{R}^{-1}\mathbf{a}(\theta_s)}. \tag{14}$$

Based on this solution, we can evaluate the beam pattern using the expression

$$B(\theta) = \mathbf{w}_{\text{SCB}}^H \mathbf{a}(\theta) \quad \text{for} \quad -90\,\text{deg} \leq \theta \leq 90\,\text{deg}\,. \tag{15}$$

By setting to a suitable scanning resolution ϵ, we search for nulls outside the SOI region. The location of the nulls outside the SOI region would correspond to the interferences or grating nulls. We represent the set of all the angles corresponding to these nulls as $\mathcal{I} = \{\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_{K-1}\}$, where the maximum number of nulls found is $K-1$. This is done heuristically and can be written as,

$$\mathcal{I} = \{\mathcal{I} : \mathcal{I} = \theta \,|\, \{\theta < \theta_L\} \cup \{\theta > \theta_U\}, B(\theta - \epsilon) > B(\theta) < B(\theta + \epsilon)\} \tag{16}$$

where ϵ is the scanning resolution used. Hence, from the interferences' DOA estimates given by $\mathcal{I}$ we can then form the null-response constraints in (13).

3.2. Solving for Beamformer's Weight

Recall that the objective function and the constraints are defined in (12) and (13). Also, the estimates of interferences' DOA $\hat{\theta}_i$ can be obtained from the approach described in Section 3.1. Hence, the proposed adaptive beamformer's weight can be solved from the following optimization formulation

$$\begin{array}{ll} \text{minimize} & f(\mathbf{w}) \\ \text{subject to} & h_i(\mathbf{w}) = 0 \quad i = 1, \ldots, K-1 \end{array} \tag{17}$$

By using a tolerance value of δ, the equality constraints can be written as

$$g_i(\mathbf{w}) = \max\{|h_i(\mathbf{w})| - \delta, 0\} \quad i = 1, \ldots, K-1$$

With the above definitions, the objective is to minimize the fitness function $f(\mathbf{w})$ such that the optimal solution obtained satisfies all the equality constraints $g_i(\mathbf{w})$. Let $v(\mathbf{w})$ denote the overall constraint violation for an individual, formulated as a weighted mean of all the equality constraints according to

$$v(\mathbf{w}) = \frac{\sum_{i=1}^{K-1} \sigma_i g_i(\mathbf{w})}{\sum_{i=1}^{K-1} \sigma_i}$$

where $\sigma_i = \frac{1}{g_{\max_i}}$ is a weight parameter, $g_{\max_i}$ is the maximum violation of the $g_i(\mathbf{w})$ constraint obtained so far. Here, σ_i is set as $\frac{1}{g_{\max_i}}$ which varies during the evolution in order to balance the contribution of every constraint in the problem irrespective of their differing numerical ranges.

As the formulation for the above robust adaptive beamforming involves equality constraints, high quality solutions can be obtained by constraint handling methods like ϵ-constraint with proper control of the ϵ parameter [29, 30]. In ϵ-constraint handling method the relaxation of the constraints is controlled by using the ϵ parameter. The ϵ level is updated until the generation counter G reaches the control generation T_c. After the generation counter exceeds T_c, the ϵ level is set to zero to obtain solutions with no constraint violation. That is,

$$\begin{array}{ll} \text{Initialize:} & \epsilon(0) = v(\mathbf{w}_\theta) \\ \text{Update:} & \epsilon(G) = \begin{cases} \epsilon(0)\left(1 - \frac{G}{T_c}\right)^{cp}, & 0 < G < T_c \\ 0, & G \geq T_c \end{cases} \end{array} \tag{18}$$

where $\mathbf{w}_\theta$ is the top ϑ-th individual and $\vartheta = 0.5NP$ where NP denotes the number of individuals in [29, 30]: $T_c \in [0.1T_{\max}, 0.8T_{\max}]$ and $cp \in [2, 10]$. The algorithmic description of the DE algorithm can be summarized as

STEP 1: Set the generation count $G = 0$, and randomly initialize a population of NP individuals $P_G = \{\mathbf{w}_{1,G}, \ldots, \mathbf{w}_{NP,G}\}$ with $\mathbf{w}_{i,G} = \{\mathbf{w}^1_{i,G}, \ldots, \mathbf{w}^D_{i,G}\}$, $i = 1, \ldots, NP$ uniformly distributed in the range $[\mathbf{w}_{\min}, \mathbf{w}_{\max}]$.

STEP 2: Select a pool of mutation strategies and a pool of values for each associated parameters corresponding to each mutation strategy.

STEP 3: Each population member is randomly assigned with one of the mutation strategy from the pool and the associated parameter values are chosen randomly from the corresponding pool of values.

STEP 4: WHILE stopping criterion is not satisfied, DO FOR $i = 1$ to NP.

Mutation Step

Generate a mutated vector $V_{i,G} = \{V^1_{i,G}, \ldots, V^D_{i,G}\}$, $i = 1, \ldots, NP$ corresponding to the target vector $\mathbf{w}_{i,G}$

$$V_{i,G} = \mathbf{w}_{r^i_1,G} + F(\mathbf{w}_{r^i_2,G} - \mathbf{w}_{r^i_3,G})$$

The indices r_1, r_2, r_3 are mutually exclusive integers randomly generated anew for each mutant vector within the range $[1, NP]$, which are also different from the index i.

Crossover Step

Generate a trial vector $U_{i,G} = U^1_{i,G}, \ldots, U^D_{i,G}$, $i = 1, \ldots, NP$ for each target vector $\mathbf{w}_{i,G}$

$$u^j_{i,G} = \begin{cases} V^j_{i,G}, & rand^j(0,1) \le CR \quad \text{or} \quad j = j_{rand} \\ \mathbf{w}^j_{i,G}, & \text{otherwise} \end{cases}$$

where $j = 1, \ldots, D$.

Selection Step

Evaluate the trial vector $U_{i,G}$

$$\mathbf{w}_{i,G+1} = \begin{cases} U_{i,G}, & v(U_{i,G}) < v(\mathbf{w}_{i,G}) \\ & \text{or } \ v(U_{i,G}) = v(\mathbf{w}_{i,G}) = 0 \\ & \text{and } f(U_{i,G}) < f(\mathbf{w}_{i,G}) \\ \mathbf{w}_{i,G}, & \text{otherwise} \end{cases}$$

Increment the generation count $G = G + 1$.

STEP 5: END WHILE.

4. SIMULATION RESULTS

Consider 10-element ULA with half-wavelength spacing receiving three Gaussian signals: the SOI from $\theta_s = 1^\circ$ and two interferences from

$\theta_i = \{20^\circ, 30^\circ\}$. The two interferences are of equal power, i.e., 20 dB and 0 dB white Gaussian distributed random variable is considered as the additive noise. Here, we only consider the mismatch due to the look-direction error although at later simulations we will demonstrate the efficacy of the proposed algorithm in the presence of both look-direction and array geometry mismatch. Also, $\hat{\mathbf{R}}$ calculated from 100 snapshots is used to implement all the beamformers discussed here. In EPSDE algorithm the only parameter that has to be tuned is the population size (NP). In our experimentation we tried different values for NP (for example 10, 20, 30, 40 and 50). The population size $NP = 20$ gives the best values for maximum function evaluations of 50000 and maximum generations of 2500. The other parameter values used are: Tolerance for equality constraints $= 10^{-8}$, $Tc = 0.8\times$ Maximum Generations and $cp = 10$.

First, we look at the beampattern plot for the proposed robust beamforming approach as compared to some of the existing robust beamforming approaches at 10 dB SNR. Figure 1 shows the beampattern plot comparison of the proposed approach against the RCB and the IRCB [32] approaches. These are obtained from one of the realization in the simulation. The proposed approach provides deeper null in the directions of interferences (e.g., 20° and 30°) as well as lower side lobe level.

As a result of the better beamformer's response, we expect

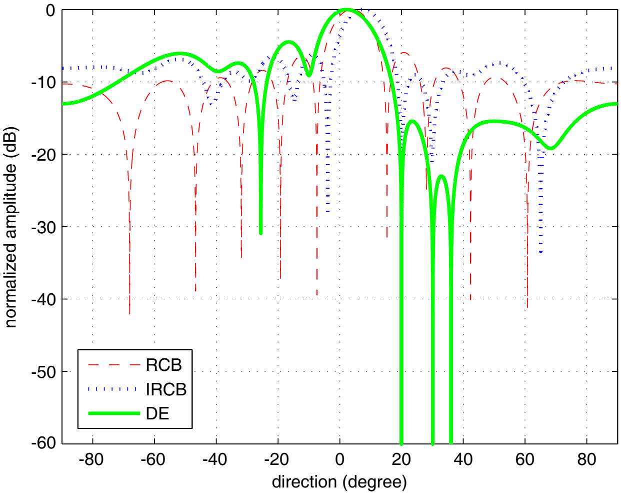

Figure 1. Beampattern comparison of the best results of RCB, IRCB and the proposed beamformer for 15% geometry error case.

to observe better SINR performance as well. Therefore in the next simulation, we evaluate the SINR performance of the proposed approach calculated from 100 realizations and compare this with the other existing approaches as well as the theoretically optimal SINR. This is repeated across various SNR ranging from −30 dB to 30 dB. Figure 2 shows the output SINR plot as a function of input SNR. For the proposed approach, Table 1 details the output SINR obtained for the proposed approach. It lists the mean, standard deviation, best and worst output SINR obtained from the 100 Monte Carlo realizations run.

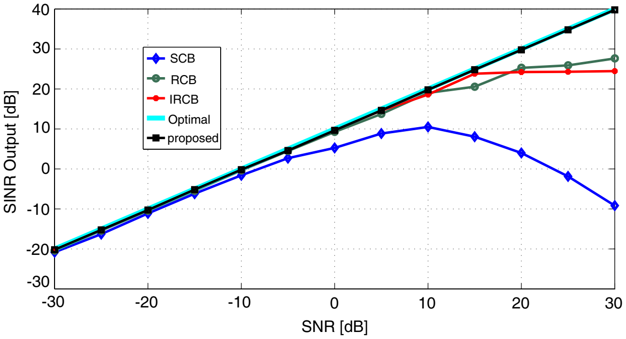

Figure 2. Best output SINR versus SNR for no geometry error case.

Table 1. For no geometry error case.

SNR [dB]	SINR [dB]			
	Mean	STD	Best	Worst
−30	−22.0428	1.4355	−20.1902	−26.1966
−25	−17.3313	2.2922	−15.2362	−27.2179
−20	−11.9424	1.9474	−10.2698	−24.6896
−15	−6.4075	1.0932	−5.1682	−12.3138
−10	−1.2708	0.9750	−0.1803	−5.7857
−5	3.6565	1.4674	4.6042	−8.8450
0	8.5391	1.3168	9.6605	−0.7481
5	13.2175	2.7708	14.6395	−7.6216
10	18.8960	0.7102	19.7761	16.1672
15	23.9794	0.6474	24.7982	21.2416
20	29.0003	0.9349	29.7846	21.7928
25	34.1476	0.5379	34.7891	31.3588
30	39.1493	0.5026	39.7790	37.2189

From the results in Table 1, it can be observed that the best results of the proposed algorithm is approximately equal to the optimal values across all the simulated input SNR. From Fig. 1, as the input SNR increases the difference in performance of the proposed algorithm and the existing robust beam forming algorithms can be clearly observed. The improved performance of the proposed algorithm is due to the parallel search procedure used in the DE algorithm.

To show the robustness of the proposed algorithm, we introduce array geometry error which is modeled as uniform random variable according to $\mathcal{U}(-0.15\lambda, 0.15\lambda)$, where λ is the signal wavelength. Fig. 3

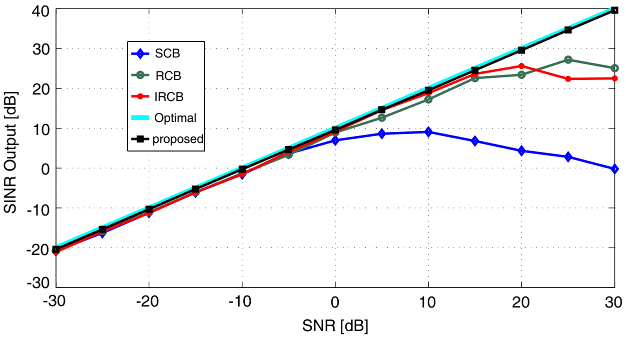

Figure 3. Best output SINR versus SNR for 15% geometry error case.

Table 2. For the case with 15% geometry error.

SNR [dB]	SINR [dB]			
	Mean	STD	Best	Worst
−30	−24.6877	3.4527	−20.3700	−38.0661
−25	−19.1883	2.9335	−15.3710	−29.2180
−20	−14.2087	3.2726	−10.3208	−23.7799
−15	−9.0150	3.1734	−5.2639	−23.0553
−10	−3.5775	2.8548	−0.2708	−13.8238
−5	2.0242	2.1475	4.6662	−6.0678
0	7.0828	2.5196	9.5768	−5.3458
5	12.0608	3.4642	14.6450	−5.1473
10	17.8096	2.6446	19.5175	0.0376
15	23.3175	1.1645	24.5569	17.5174
20	28.4938	0.9266	29.5716	24.7978
25	33.4958	1.0332	34.6542	28.0268
30	38.5034	1.2025	39.6062	29.2504

shows the performance of the proposed approach as compared to the other existing approach when the array geometry error is considered to contribute to the mismatch in the steering vector. These results show that the proposed approach provides certain degree of robustness when the presumed array geometry does not equal to the actual geometry. Table 2 lists the mean, standard deviation, best and worst output SINR for this case.

5. CONCLUSION

This paper proposes a near-optimal robust adaptive beamforming approach based on solving non-convex optimization using evolutionary algorithm. The ability to achieve near-optimal performance is attributed to the formulation of the novel optimization function for solving the beamformer's weight vector. In particular, the objective function is defined as the ratio between the interference-plus-noise response and the desired signal response. By minimizing the objective function subjected to the constraint that nullifies the response at interferences' DOA, the near-optimal performance can then be achieved. As compared to the commonly used robust beamforming formulation, our approach does not utilize the estimated covariance matrix that is calculated from the array observation that includes the desired signal. Although the objective function is no longer convex, the problem is still solvable using differential evolution (DE) algorithm. Numerical results demonstrate the efficacy of the proposed approach in comparison with other existing robust techniques. As a future work we would like to implement the proposed adaptive beam forming technique using DE on GPUs to overcome the computational complexity.

REFERENCES

1. Gu, Y.-J., Z.-G. Shi, K. S. Chen, and Y. Li, "Robust adaptive beamforming for steering vector uncertainties based on equivalent DOAs method," *Progress In Electromagnetics Research*, Vol. 79, 277–290, 2008.
2. Gu, Y.-J., Z.-G. Shi, K. S. Chen, and Y. Li, "Robust adaptive beamforming for a class of Gaussian steering vector mismatch," *Progress In Electromagnetics Research*, Vol. 81, 315–328, 2008.
3. Li, Y., Y.-J. Gu, Z.-G. Shi, and K. S. Chen, "Robust adaptive beamforming based on particle filter with noise unknown," *Progress In Electromagnetics Research*, Vol. 90, 151–169, 2009.
4. Cox, H., "Resolving power and sensitivity to mismatch of optimum

array processors," *Journal of the Acoustical Society of America*, Vol. 54, No. 3, 771–785, 1973.

5. Takao, K., M. Fujita, and T. Nishi, "An adaptive antenna array under directional constraint," *IEEE Transactions Antennas and Propagation*, Vol. 24, No. 5, 662–669, 1976.
6. Er, M. H. and A. Cantoni, "Derivative constraints for broad-band element space antenna array processors," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 31, No. 6, 1378–1393, 1983.
7. Krolik, J. L., "The performance of matched-field beamformers with Mediterranean vertical array data," *IEEE Transactions on Signal Processing*, Vol. 44, No. 10, 2605–2611, 1996.
8. Jablon, N., "Adaptive beamforming with the generalized sidelobe canceller in the presence of array imperfections," *IEEE Transactions on Antennas and Propagation*, Vol. 34, No. 8, 996–1012, 1986.
9. Vorobyov, S. A., A. B. Gershman, and Z.-Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem," *IEEE Transactions on Signal Process*, Vol. 51, No. 2, 313–324, 2003.
10. Shahbazpanahi, S., A. B. Gershman, Z.-Q. Luo, and K. M. Wong, "Robust adaptive beamforming for general-rank signal models using worst-case performance optimization," *Proceedings SAM Workshop*, 13–17, 2002.
11. Lorenz, R. G. and S. P. Boyd, "Robust minimum variance beamforming," *IEEE Transactions on Signal Processing*, Vol. 53, No. 5, 1684–1696, 2005.
12. Li, J., P. Stoica, and Z. Wang, "On robust capon beamforming and diagonal loading," *IEEE Transactions on Signal Processing*, Vol. 51, No. 7, 1702–1715, 2003.
13. Li, J., P. Stoica, and Z. Wang, "Doubly constrained robust capon beamformer," *IEEE Transactions on Signal Processing*, Vol. 52, No. 9, 2407–2423, 2004.
14. Du, L., J. Li, and P. Stoica, "Fully automatic computation of diagonal loading levels for robust adaptive beamforming," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 46, No. 1, 449–458, Jan. 2010.
15. Selén, Y., R. Abrahamsson, and P. Stoica, "Automatic robust adaptive beamforming via ridge regression," *Signal Processing*, Vol. 88, No. 1, 33–49, 2008.
16. De Veronese, L. P. and R. A. Krohling, "Differential evolution

algorithm on the GPU with C-CUDA," *IEEE Congress on Evolutionary Computation (CEC)*, 1–7, Barcelona, Spain, Jul. 2010.

17. De Donno, D., A. Esposito, L. Tarricone, and L. Catarinucci, "Introduction to GPU computing and CUDA programming: A case study on FDTD," *IEEE Antennas and Propagation Magazine*, Vol. 52, No. 3, 116–122, Jun. 2010.
18. Owens, J. D., D. Luebke, N. Govindaraju, M. Harris, J. Krger, A. E. Lefohn, and T. J. Purcell, "A survey of general-purpose computation on graphics hardware," *Computer Graphics Forum*, Vol. 26, No. 1, 80–113, Mar. 2007.
19. Storn, R. and K. Price, "Differential evolution — A simple and efficient adaptive scheme for global optimization over continuous spaces," *ICSI Technical Report TR-95-012*, Vol. 11, http://http.icsi.berkeley.edu/storn/litera.html, 1995.
20. Storn, R. and K. Price, "Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, Vol. 11, 341–359, 1997.
21. Storn, R., K. V. Price, and J. Lampinen, *Differential Evolution — A Practical Approach to Global Optimization*, Berlin, 2005.
22. Dib, N. I., S. K. Goudos, and H. Muhsen, "Application of Taguchi's optimization method and self-adaptive differential evolution to the synthesis of linear antenna arrays," *Progress In Electromagnetics Research*, Vol. 102, 159–180, 2010.
23. Goudos, S. K., K. Siakavara, E. Vafiadis, and J. N. Sahalos, "Pareto optimal Yagi-Uda antenna design using multi-objective differential evolution," *Progress In Electromagnetics Research*, Vol. 105, 231–251, 2010.
24. Goudos, S. K., Z. D. Zaharis, and T. Yioultsis, "Application of a differential evolution algorithm with strategy adaptation to the design of multi-band microwave filters for wireless communications," *Progress In Electromagnetics Research*, Vol. 109, 123–137, 2010.
25. Das, S. and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. on Evolutionary Computation*, Feb. 2011, DOI: 10.1109/TEVC.2010.2059031.
26. Huang, V. L., A. K. Qin, and P. N. Suganthan, "Self-adaptive differential evolution algorithm for constrained real-parameter optimization," *IEEE Congress on Evolutionary Computation, CEC 2006*, 17–24, 2006.
27. Zhang, J. and A. C. Sanderson, "Jade adaptive differential

evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 5, 945–958, 2009.

28. Qin, A. K., V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 13, 398–417, Apr. 2009.
29. Takahama, T. and S. Sakai, "Constrained optimization by the constrained differential evolution with gradient-based mutation and feasible elites," *Proc. IEEE Congress on Evolutionary Computation*, 1–8, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 2006.
30. Mallipeddi, R. and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Trans. on Evolutionary Computation*, Vol. 14, No. 4, 561–579, Aug. 2010.
31. Mallipeddi, R., P. N. Suganthan, Q. K. Pan and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, Vol. 11, No. 2, 1679–1696, 2010.
32. Lie, J. P., X. Li, W. Ser, C. M. S. See, and L. Lei, "Adaptive uncertainty based iterative robust capon beamformer," *Proc. IEEE Int Acoustics Speech and Signal Processing (ICASSP) Conf.*, 2526–2529, 2010.