Robust Adaptive LCMV Beamformer Based on an Iterative Suboptimal Solution

Xiansheng GUO, Lei CHU, Baocang LI

Dept. of Electronic Engineering, University of Electronics Science and Technology of China,
Chengdu, Sichuan, 611731, China

xsguo@uestc.edu.cn, uestcahcl@gmail.com

Abstract. *The main drawback of closed-form solution of linearly constrained minimum variance (CF-LCMV) beamformer is the dilemma of acquiring long observation time for stable covariance matrix estimates and short observation time to track dynamic behavior of targets, leading to poor performance including low signal-noise-ratio (SNR), low jammer-to-noise ratios (JNRs) and small number of snapshots. Additionally, CF-LCMV suffers from heavy computational burden which mainly comes from two matrix inverse operations for computing the optimal weight vector. In this paper, we derive a low-complexity Robust Adaptive LCMV beamformer based on an Iterative Suboptimal solution (RAIS-LCMV) using conjugate gradient (CG) optimization method. The steepest descent weight updated strategy is adopted to obtain a simple iteration process. The merit of our proposed method is threefold. Firstly, RAIS-LCMV beamformer can reduce the complexity of CF-LCMV remarkably. Secondly, RAIS-LCMV beamformer can adjust output adaptively based on measurement and its convergence speed is comparable. Finally, RAIS-LCMV algorithm has robust performance against low SNR, JNRs, and small number of snapshots. Simulation results demonstrate the superiority of our proposed algorithms.*

Keywords

Robust adaptive beamformer, LCMV, iterative suboptimal solution, computational analysis

1. Introduction

Linearly constrained minimum variance (LCMV) beamformer is a conventional and powerful tool for signal enhancement in multiple sources cases. The conventional LCMV beamformer is a kind of closed form LCMV (CF-LCMV) beamformer which is obtained by directly using Lagrange multipliers and is discussed extensively in many works, such as [1, 2, 4]. As a generalized minimum variance distortion-less response (MVDR) beamformer, the conventional LCMV beamformer minimizes the array output power while maintaining a constant response in the direction of the signal of interest (SOI). However, CF-LCMV beamformer requires inversions of the input data covariance matrix $\mathbf{R}_{xx}$ and $\mathbf{C}^H\mathbf{C}$ ($\mathbf{C}$ is a constraint matrix), causing the main computation burden of CF-LCMV algorithm. Additionally, the robustness of CF-LCMV is poor when small number of snapshots, low SNR and JNRs, are present [16].

Numerous adaptive versions of LCMV were reported in the last decades to overcome the above listed drawbacks of CF-LCMV, including LMCV with stochastic gradient (LCMV-SG) algorithm [13], LCMV with recursive least squares (LCMV-RLS) algorithm [6], and so on. Among these algorithms, the LCMV-SG algorithm is a low complexity method but converges slowly with correlated data inputs [9]. The LCMV-RLS algorithm has a fast convergence speed but suffers from high complexity numerical instability. To reduce the computational complexity and improve the stability of CF-LCMV algorithm, a low-complexity constrained affine-projection (CAP) algorithm is proposed to update the weight vector of linearly constrained adaptive filters [12] recently. The performance of CAP beamformer is robust but the complexities of CAP and its derivations are still high.

Comparatively speaking, conjugate gradient (CG) technique shows an attractive trade-off between performance and complexity and has been adopted in many related works, such as [10, 9, 11]. Among them, [10] is a convex optimization framework. In each iteration, the algorithm in [10] needs to solve a constrained least square problem, which leads to a more heavy computational burden. Based on the conventional MVDR criterion, [11] proposed a stronger constraint set to constrain the magnitude of array output and the so-called set-membership)(SM) technique is adopted to specify a bound on the magnitude of the estimation error or the array output, which can reduce the computational complexity because of data selective updates. While [9] is based on a constrained constant modulus (CCM) criterion and a modified conjugate gradient (MCG) and a conventional conjugate gradient (CCG) methods are used to drive the optimal weight vector. The weight update strategies of the two approaches are recursive least squares (RLS), which is more complicated than steepest descent technique.The two approaches can improve the performance to some extent with a more complicated iteration process. Additionally, their performances are affected by too many parameters, which degenerates their performance if these parame-

DOI: 10.13164/re.2015.0572

ters are not given accurately. In this paper, by using CG technique and steepest descent technique, we derive robust adaptive LCMV beamformer based on iterative suboptimal solution (RAIS-LCMV) to improve the performance of CF-LCMV beamformer. Compared with the aforementioned approaches, our approach is only based on the conventional LCMV framework, we needn't other constraints to our cost function. Hence, it only needs several simple iteration steps. Firstly, we derive the updating procedures of RAIS-LCMV by using CG technique. The way to determine the parameters to be used in RAIS-LCMV is addressed subsequently. Furthermore, we discuss the convergence speed and computational complexity of RAIS-LCMV algorithm. Compared with some existing LCMV beamformers, our proposed algorithm has a low complexity and achieves better performance with small number of snapshots and low SNR and JNRs cases.

2. Problem Formulation

Consider D point source signals (including signals and interferences) impinging on an array comprising M sensors with an arbitrary geometry from directions $\theta_1, \theta_2, \cdots, \theta_D$. The output $\mathbf{y}(k)$ of array beamformer can be expressed as

$$y(k) = \mathbf{w}^H \mathbf{x}(k), \tag{1}$$

where $k = 1, 2, \cdots, K$ is the time index and K is the number of snapshot, $\mathbf{w}$ is a $M \times 1$ complex vector of beamformer weight to be estimated, and $(\cdot)^H$ stands for Hermitian transpose.

For a minimum-variance distortionless-response (MVDR) beamformer, the objective is to minimize the array output energy, subject to a linear constraint on the desired direction-of-arrival (DOA), i.e.,

$$\begin{aligned} &\text{minimize} \quad J(\mathbf{w}) = \tfrac{1}{2}\mathbf{w}^H \mathbf{R}_{xx}\mathbf{w} \\ &\text{subject to} \quad \mathbf{w}^H \mathbf{a}(\theta_1) = 1, \end{aligned} \tag{2}$$

where $\mathbf{R}_{xx} = E\left\{\mathbf{x}(k)\mathbf{x}^H(k)\right\}$ is the covariance matrix of the received signal $\mathbf{x}$ with $E\{\cdot\}$ being the expectation operator.

For LCMV-type beamformers, $\mathbf{w}$ is the solution to the following multiple linearly constrained optimization problem:

$$\begin{aligned} &\text{minimize} \quad J(\mathbf{w}) = \tfrac{1}{2}\mathbf{w}^H \mathbf{R}_{xx}\mathbf{w} \\ &\text{subject to} \quad \mathbf{C}^H \mathbf{w} = \mathbf{f}, \end{aligned} \tag{3}$$

where $\mathbf{C}$ is the aforementioned constraint matrix and $\mathbf{f}$ is a constraint vector. For example, if we need to generate unit gains in L DOAs θ_i $(i = 1, 2, \cdots, L)$ while forming nulling in other DOAs, the constraint matrix and constraint vector can be expressed as

$$\mathbf{C} = [\mathbf{a}(\theta_1), \cdots, \mathbf{a}(\theta_L), \mathbf{a}(\theta_{L+1}), \cdots, \mathbf{a}(\theta_D)] \tag{4}$$

and

$$\mathbf{f} = \Bigg[\underbrace{1, \cdots, 1}_{L}, \underbrace{0, \cdots, 0}_{D-L}\Bigg]^T, \tag{5}$$

respectively. Based on (3), (4), and (5), by using Lagrange multipliers technique, one can obtain a closed-form solution of the optimal weight vector $\mathbf{w}$ as follows [13]

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{C}\left(\mathbf{C}^H \mathbf{R}_{xx}^{-1}\mathbf{C}\right)^{-1}\mathbf{f}. \tag{6}$$

Equation (6) is the well-known optimal weight vector of LCMV beamformer and it is a closed-form solution to (3). The closed-form solution of (6) can constrain the interferences and keep simultaneous response of multiple expected signals. To obtain the optimal weight vector, we need to compute two matrix inversions from (6), i.e., one is the inverse operation of covariance matrix $\mathbf{R}_{xx}$, with size $M \times M$, the other is the inverse operation of matrix $\mathbf{C}^H \mathbf{R}_{xx}^{-1}\mathbf{C}$, with size $D \times D$. For large arrays or adaptive phase array radar (APAR) [18], the number of array sensors is very large, so the computational complexity of covariance matrix inversion is very high. So how to decrease the computational burden of CF-LCMV is a serious problem. Additionally, the main drawback of the aforementioned two beamformers is the dilemma of acquiring long observation time for stable covariance matrix estimates and short observation time to track dynamic behavior of targets, leading to poor performance including low SNR, low JNRs, and small number of snapshots.

In [1], Breed and Strauss derived an equivalent generalized sidelobe canceller (GSC) structure of LCMV (GSC-LCMV). GSC-LCMV is obtained by splitting the applied filters into two components, i.e., $\mathbf{w} = \mathbf{w}_q - \mathbf{w}_s$. The components of $\mathbf{w}_q$ and $\mathbf{w}_s$ lie in the column-subspace of the constraint matrix $\mathbf{C}$ and its component null-subspace, respectively. The GSC-LCMV formulation of the problem can be expressed as following

$$\begin{cases} \mathbf{w} = \mathbf{w}_q - \mathbf{w}_s, & \text{(7a)} \\ \mathbf{w}_q = \mathbf{C}\mathbf{Q}^{-1}\mathbf{w}_a, & \text{(7b)} \\ \mathbf{w}_s = \mathbf{C}_a \mathbf{w}_a, & \text{(7c)} \\ \mathbf{w}_a = \left(\mathbf{C}_a^H \mathbf{R}_{xx}\mathbf{C}_a\right)^{-1}\mathbf{C}_a^H \mathbf{R}_{xx}\mathbf{w}_q, & \text{(7d)} \\ \mathbf{Q} = \mathbf{C}^H \mathbf{C}, & \text{(7e)} \end{cases}$$

where the matrix $\mathbf{C}_a \in \mathcal{C}^{D\times(D-L)}$ is composed of the ortho-complement vectors of constraint matrix $\mathbf{C}$. It should be noted that GSC-LCMV also requires two matrix inversions and its computational complexity will be slightly decreased when number of signals is far less than that of array elements. Overall speaking, the computational burden of GSC-LCMV is still heavy and its performance is still poor with small number of snapshots, low SNRs and JNRs.

In order to ease the computational burdens of CF-LCMV and GSC-LCMV and obtain a better performance under small number of snapshots, low SNRs and JNRs, we derive a low-complexity RAIS-LCMV algorithm using conjugate gradient (CG) optimization technique. The proposed RIAS-LCMV beamformer does not require matrix inversion in each iteration and the suboptimal weight vector is computed iteratively given small number of iterations. Compared to CF-LCMV, GSC-LCMV beamformers, and some other existing iterative methods, RAIS-LCMV beamformer has lower computational complexity and shows great robustness to SNR, JNRs, and number of snapshots.

3. Robust Adaptive LCMV Beamformer Based on an Iterative Suboptimal Solution (RAIS-LCMV)

Consider the objective optimization function of (3) which is based on the typical least mean square (LMS) criterion. One can obtain a solution of weight vector by using conjugate gradient (CG) optimization algorithm [7, 8, 14]. In order to derive an iterative suboptimal solution of weight vector $\mathbf{w}$, the proposed RIAS-LCMV beamformer is designed using the following optimization problem in real valued form:

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^H\mathbf{R}_{\text{xx}}\mathbf{w} + \text{Real}\left\{\lambda^H\left(\mathbf{C}^H\mathbf{w}-\mathbf{f}\right)\right\}, \tag{8}$$

where Real$\{\cdot\}$ is an operator of taking real part and $\lambda \in \mathcal{C}^{D\times 1}$ is a Lagrange multipliers vector. Calculating the conjugate gradient of $J(\mathbf{w})$ with respect to $\mathbf{w}$, we have

$$\nabla_{\mathbf{w}}J(\mathbf{w}) = \mathbf{R}_{xx}\mathbf{w} + \frac{\partial\left(\text{Real}\left\{\lambda^H\left(\mathbf{C}^H\mathbf{w}-\mathbf{f}\right)\right\}\right)}{\partial\mathbf{w}^*}, \tag{9}$$

in which

$$\frac{\partial\left(\text{Real}\left\{\lambda^H\left(\mathbf{C}^H\mathbf{w}-\mathbf{f}\right)\right\}\right)}{\partial\mathbf{w}^*} = \frac{\partial\left(\text{Real}\left\{\lambda^H\mathbf{C}^H\mathbf{w}\right\}\right)}{\partial\mathbf{w}^*}. \tag{10}$$

It is noted that $\lambda^H\mathbf{C}^H\mathbf{w}$ is a complex scalar, thus we have

$$\text{Real}\left\{\lambda^H\mathbf{C}^H\mathbf{w}\right\} = \frac{1}{2}\left(\lambda^H\mathbf{C}^H\mathbf{w}+\mathbf{w}^H\mathbf{C}\lambda\right). \tag{11}$$

Based on (11), (10) can be expanded to

$$\begin{aligned}\frac{\partial\left(\text{Real}\left\{\lambda^H\left(\mathbf{C}^H\mathbf{w}-\mathbf{f}\right)\right\}\right)}{\partial\mathbf{w}^*} &= \frac{1}{2}\frac{\partial\left(\left(\mathbf{C}\lambda\right)^H\mathbf{w}\right)}{\partial\mathbf{w}^*} \\ &\quad+\frac{1}{2}\frac{\partial\left(\mathbf{w}^H\mathbf{C}\lambda\right)}{\partial\mathbf{w}^*}.\end{aligned} \tag{12}$$

Since $\lambda^H\mathbf{C}^H = (\mathbf{C}\lambda)^H$ and $\mathbf{C}\lambda \in \mathcal{C}^{M\times 1}$, using the rule of gradient derivation, the first item of right side of (12) is zero, and the second item of right side of (12) is $\mathbf{C}\lambda$, so we have

$$\frac{\partial\left(\text{Real}\left\{\lambda^H\left(\mathbf{C}^H\mathbf{w}-\mathbf{f}\right)\right\}\right)}{\partial\mathbf{w}^*} = \mathbf{C}\lambda. \tag{13}$$

Based on the results above, we can calculate the derivation of $\nabla_{\mathbf{w}}J(\mathbf{w})$ as

$$\nabla_{\mathbf{w}}J(\mathbf{w}) = \mathbf{R}_{xx}\mathbf{w} - \mathbf{C}\lambda. \tag{14}$$

The following updating formulation is used to compute the weight vector $\mathbf{w}(n+1)$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\left[-\nabla_{\mathbf{w}}J(\mathbf{w})\right], \tag{15}$$

where μ is a non-negative step size parameter, which decides the convergence of our method. Substituting (14) into (15) and using $\lambda(n)$ instead of λ as the n index, we have

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\left[\mathbf{R}_{xx}\mathbf{w}(n) - \mathbf{C}\lambda(n)\right], \tag{16}$$

where $\lambda(n)$ varies with index n. We constrain that the weight vector $\mathbf{w}(n+1)$ must satisfy the following constraint condition in each iteration

$$\mathbf{C}^H\mathbf{w}(n+1) = \mathbf{f}. \tag{17}$$

Premultiplying both sides of (15) by $\mathbf{C}^H$ yields

$$\mathbf{C}^H\mathbf{w}(n+1) = \mathbf{C}^H\mathbf{w}(n) - \mu\mathbf{C}^H\left[\mathbf{R}_{xx}\mathbf{w}(n) - \mathbf{C}\lambda(n)\right]. \tag{18}$$

Combining (17) and (18), we can obtain the following equivalent form

$$\mathbf{f} = \mathbf{C}^H\mathbf{w}(n) - \mu\mathbf{C}^H\left[\mathbf{R}_{xx}\mathbf{w}(n) - \mathbf{C}\lambda(n)\right]. \tag{19}$$

Consequently, the expression of $\lambda(n)$ is given as follows

$$\begin{aligned}\lambda(n) &= \frac{1}{\mu}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{f} - \frac{1}{\mu}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H\mathbf{w}(n) \\ &\quad+\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H\mathbf{R}_{xx}\mathbf{w}(n).\end{aligned} \tag{20}$$

Substituting (20) into (16), we can obtain the iterative equation of weight vector $\mathbf{w}(n+1)$ as follows

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) - \mu\mathbf{R}_{xx}\mathbf{w}(n) + \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{f} \\ &\quad-\mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H\mathbf{w}(n) \\ &\quad+\mu\mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H\mathbf{R}_{xx}\mathbf{w}(n).\end{aligned} \tag{21}$$

Simplifying (21) further yields

$$\begin{aligned}\mathbf{w}(n+1) &= \left[\mathbf{I} - \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H\right]\left[\mathbf{I} - \mu\mathbf{R}_{xx}\right]\mathbf{w}(n) \\ &\quad+\mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{f}.\end{aligned} \tag{22}$$

Denoting

$$\mathbf{P} = \left[\mathbf{I} - \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H\right], \tag{23}$$

and

$$\mathbf{G} = \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{f}, \tag{24}$$

we can rewrite the iterative process of weight vector $\mathbf{w}(n+1)$ as the following compact form

$$\mathbf{w}(n+1) = \mathbf{P}\left(\mathbf{I} - \mu\mathbf{R}_{xx}\right)\mathbf{w}(n) + \mathbf{G}. \tag{25}$$

Equation (25) is the updating formulation of estimated suboptimal weight vector. Because the covariance matrix is not an iterative form, the beamformer constructed by (25) is not an adaptive one. For simplicity, we denote the updating procedure of (25) as a simple iterative suboptimal solution of LCMV (SIS-LCMV) beamformer. Note that the covariance matrix of $\mathbf{R}_{xx}$ is unavailable in real time, so SIS-LCMV beamformer is still torn between the need of long observation time for stable covariance matrix estimates and the need of short observation time to track dynamic behavior of targets.

Generally speaking, an available approximation of $\mathbf{R}_{xx}$ is

$$\hat{\mathbf{R}}_{xx} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}(k)\,\mathbf{x}^{H}(k), \tag{26}$$

in which K is the total number of snapshots. Hence, the weight updating formula of SIS-LCMV should be substituted by

$$\mathbf{w}(n+1) = \mathbf{P}\left(\mathbf{I} - \mu\hat{\mathbf{R}}_{xx}\right)\mathbf{w}(n) + \mathbf{G}. \tag{27}$$

The drawback of SIS-LCMV is that it needs long observation time for stable covariance matrix estimate. In order to solve the problem of long observation time and to improve the dynamic behavior of beamformer, the approximation of covariance matrix (26) can be substituted by the instantaneous estimate of covariance matrix as follows

$$\hat{\mathbf{R}}_{xx} = \mathbf{x}(k)\,\mathbf{x}^{H}(k)\,. \tag{28}$$

In this case, the suboptimal weight updating process of our proposed iterative adaptive LCMV beamformer can be rewritten as

$$y(k) = \mathbf{w}^{H}(k)\,\mathbf{x}(k)\,, \tag{29}$$

and

$$\mathbf{w}(k+1) = \mathbf{P}\{\mathbf{w}(k) - \mu y(k)\,\mathbf{x}(k)\} + \mathbf{G}, \tag{30}$$

in which the iteration index $k = 1, 2, \cdots, K$ with k being the index of measurement number. The scalar $y(k)$ is the instantaneous output of beamformer in kth observation time. Using the updating formulae (29) and (30), we can obtain an adaptive iterative suboptimal solution of LCMV (AIS-LCMV) beamformer.

It should be noted that the updating process of AIS-LCMV beamformer can improve the dynamic behavior of SIS-LCMV beamformer remarkably. However, one can imagine that the error will be large due to the poor approximation of covariance matrix. In order to improve the performance of AIS-LCMV, we adopt the recursive strategy of covariance matrix as follows

$$\tilde{\mathbf{R}}_{xx}(k) = \beta\tilde{\mathbf{R}}_{xx}(k-1) + (1-\beta)\,\mathbf{x}(k)\,\mathbf{x}^{H}(k)\,, \tag{31}$$

where β is a forgetting factor, and $\tilde{\mathbf{R}}_{xx}(k)$ and $\tilde{\mathbf{R}}_{xx}(k-1)$ are the estimates of covariance matrix in the kth and $(k-1)$th observation time, respectively. Based on the recursive estimate of covariance matrix of (31), we obtain RAIS-LCMV algorithm as follows

$$\mathbf{w}(k+1) = \mathbf{P}\left(\mathbf{I} - \mu\tilde{\mathbf{R}}_{xx}(k)\right)\mathbf{w}(k) + \mathbf{G}. \tag{32}$$

Obviously, the smooth technique used here can reduce the estimating error of covariance matrix effectively.

It should be noted that AIS-LCMV and RAIS-LCMV are the special cases of SIS-LCMV algorithm. Among them, AIS-LCMV is just the case of $K = 1$ in the estimation of covariance matrix in (26), while RAIS-LCMV adopts a data reusing strategy to improve the estimation precision of covariance matrix. We believe that RAIS-LCMV has a better performance in both tracking speed and estimation of stable covariance matrix than AIS-LCMV beamformer. Additionally, the computational complexity of RAIS-LCMV is lower than that of SIS-LCMV because the latter needs more observation time to compute the approximation of covariance matrix and the robustness of RAIS-LCMV will be superior to that of SIS-LCMV and AIS-LCMV with small number of snapshots, low SNRs and JNRs cases.

Now we summarize the three listed algorithms: SIS-LCMV, AIS-LCMV, and RAIS-LCMV in Tab. 1, Tab. 2 and Tab. 3, respectively. In Tab. 1, N is the total number of iteration in SI-LCMV and ε is the weight vector difference between two adjacent estimated weight vectors. $\bigtriangledown\delta$ is a small threshold we set. In Tab. 2 and Tab. 3, the termination condition of iterative process is determined only by the number of snapshots K.

	SIS-LCMV Algorithm
Step 1)	**Initialization**
	(a) $\mathbf{w}(0), \mu, N\,(\bigtriangledown\delta)$;
	(b) Calculate $\mathbf{P}$ using (23);
	(c) Calculate $\mathbf{G}$ using (24);
	(d) Calculate $\hat{\mathbf{R}}_{xx}$ using (26);
Step 2)	**For** $n = 1, 2, \ldots, N$
	(a) Update weight vector using (27);
Step 3)	$n \leftarrow n+1$;
Step 4)	**If** $n > N$ **or iterative error** $\varepsilon \leq \bigtriangledown\delta$**, return; else, go to (a) in Step 2)**.

Tab. 1. The proposed SIS-LCMV algorithm.

```
AIS-LCMV Algorithm
Step 1)   Initialization
              (a) $\mathbf{w}(0)$, $\mu$, $K$;
              (b) Calculate $\mathbf{P}$ using (23);
              (c) Calculate $\mathbf{G}$ using (24);
Step 2)   For $k = 1,2,\ldots,K$
              (a) Update the instantaneous output of beamformer using (29);
              (b) Update weight vector using (30);
Step 3)   $k \leftarrow k+1$;
Step 4)   If $k > K$, return; else, go to (a) and (b) in Step 2).
```

Tab. 2. The proposed AIS-LCMV algorithm.

```
RAIS-LCMV Algorithm
Step 1)   Initialization
              (a) $\mathbf{w}(0)$, $\mu$, $K$, $\tilde{\mathbf{R}}_{xx}(0)$, and $\beta$;
              (b) Calculate $\mathbf{P}$ using (23);
              (c) Calculate $\mathbf{G}$ using (24);
Step 2)   For $k = 1,2,\ldots,K$
              (a) Recursive estimation of covariance matrix using (31);
              (b) Update weight vector using (32);
Step 3)   $k \leftarrow k+1$;
Step 4)   If $k > K$, return; else, go to (a) and (b) in Step 2).
```

Tab. 3. The proposed RAIS-LCMV algorithm.

4. Parameters Selection

The parameters to be considered here are the step size parameter μ, the initial weight vector $\mathbf{w}(0)$, the total iterative number N or equivalently speaking, the iterative error $\bigtriangledown\delta$, the initial covariance matrix $\tilde{\mathbf{R}}_{xx}(0)$, and the forget factor β. Among them, the initial covariance matrix $\tilde{\mathbf{R}}_{xx}(0)$ is usually set to be identity matrix with size of $M \times M$ and the forget factor β is always selected as a constant close to one, e.g., $\beta = 0.998$. Note that AIS-LCMV and RAIS-LCMV are just the special cases of SIS-LCMV, for the sake of simplicity, we only take SIS-LCMV algorithm as an example to demonstrate the selection of some parameters. Such conclusions can be applied to AIS-LCMV and RAIS-LCMV algorithms directly.

4.1 The Step Size Parameter μ

The first parameter to be determined is the step size parameter μ. In order to determine the range bound of μ, we substitute (17) and (24) into (25), which yields

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{P}(\mathbf{I}-\mu\mathbf{R}_{xx})\mathbf{w}(n) \\ &\quad +\mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H\mathbf{w}(n+1). \qquad (33)\end{aligned}$$

Further simplifying (33), we obtain a compact form about $\mathbf{w}$ as follows

$$\mathbf{P}\mathbf{w}(n+1) = \mathbf{P}(\mathbf{I}-\mu\mathbf{R}_{xx})\mathbf{w}(n). \qquad (34)$$

Note that $\mathbf{P}$ is a full-rank matrix if all signals and interferences are not coherent.

Premultiplying both sides of (34) by $\mathbf{P}^{-1}$ yields

$$\mathbf{w}(n+1) = (\mathbf{I}-\mu\mathbf{R}_{xx})\mathbf{w}(n). \qquad (35)$$

Recall that the eigen-decomposition of $\mathbf{R}_{xx}$ is given as

$$\mathbf{R}_{xx} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H = \sum_{i=1}^{M}\lambda_i\mathbf{q}_i\mathbf{q}_i^H, \qquad (36)$$

where $\lambda_1,\lambda_2,\cdots,\lambda_M$ is the eigenvalue of $\mathbf{R}_{xx}$, $\mathbf{q}_1,\mathbf{q}_2,\cdots,\mathbf{q}_M$ is the normalized eigenvector, $\mathbf{Q}$ is a unitary matrix whose columns are composed of all eigenvectors.

$$\mathbf{Q} = [\mathbf{q}_1,\mathbf{q}_2,\cdots,\mathbf{q}_M], \qquad (37)$$

and the diagonal matrix $\mathbf{\Lambda}$ is composed of its eigenvalues

$$\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1,\lambda_2,\cdots,\lambda_M\}. \qquad (38)$$

Substituting (36) into (35), we have

$$\mathbf{w}(n+1) = \left(\mathbf{I}-\mu\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H\right)\mathbf{w}(n). \qquad (39)$$

Because of $\mathbf{Q}^H\mathbf{Q} = \mathbf{Q}\mathbf{Q}^H = \mathbf{I}$, (39) can be rewritten as

$$\mathbf{w}(n+1) = \mathbf{Q}(\mathbf{I}-\mu\mathbf{\Lambda})\mathbf{Q}^H\mathbf{w}(n). \qquad (40)$$

Premultiplying both sides of (40) by $\mathbf{Q}^H$ and denoting $\mathbf{b}(n) = \mathbf{Q}^H\mathbf{w}(n)$, we have

$$\mathbf{b}(n+1) = (\mathbf{I}-\mu\mathbf{\Lambda})\mathbf{b}(n). \qquad (41)$$

Assume that $\mathbf{b}(n) = [b_1(n), b_2(n), \cdots, b_M(n)]^T$, we can expand (41) as

$$\begin{bmatrix} b_1(n+1) \\ b_2(n+1) \\ \vdots \\ b_M(n+1) \end{bmatrix} = \begin{bmatrix} 1-\mu\lambda_1 & & \\ & \ddots & \\ & & 1-\mu\lambda_M \end{bmatrix} \times \begin{bmatrix} b_1(n) \\ b_2(n) \\ \vdots \\ b_M(n) \end{bmatrix}. \qquad (42)$$

Equivalently, each row of (42) has the following recursion form

$$b_i(n+1) = (1-\mu\lambda_i)b_i(n). \qquad (43)$$

If the initialized value of $b_i(n)$ is $b_i(0)$, then the following relationship holds

$$b_i(n) = (1-\mu\lambda_i)^n b_i(0). \tag{44}$$

Obviously, if $|1-\mu\lambda_i| < 1$, i.e., $-1 < 1-\mu\lambda_i < 1$. Or equivalently, if the following condition is satisfied,

$$0 < \mu < \frac{2}{\lambda_i}, \tag{45}$$

we can always have $\lim_{n\to\infty} b_i(n) = 0, i = 1,2,\cdots,M$.

To ensure the step size parameter μ satisfies (45) for all $\lambda_i, i = 1,2,\cdots,M$, the step size parameter μ should satisfy

$$0 < \mu < \frac{2}{\lambda_{\max}}, \tag{46}$$

where $\lambda_{\max}$ is the biggest eigenvalue of $\mathbf{R}_{xx}$.

4.2 The Initialized Weight Vector $\mathbf{w}(0)$

The second parameter to be determined in our proposed algorithm listed in Tab. 1, Tab. 2, and Tab. 3 is $\mathbf{w}(0)$. Recall that $\mathbf{G}$ in (25) is a fixed part in each iteration and note that the expression of $\mathbf{G}$ is just a min-norm solution to (3), if we choose the initialized condition of $\mathbf{w}(0) = \mathbf{0}$, then

$$\mathbf{w}(1) = \mathbf{G}, \tag{47}$$

that is to say, $\mathbf{w}(1)$ is the min-norm solution of LCMV. In other words, the expected weight vector will be the min-norm solution after only one iteration. Meanwhile, setting $\mathbf{w}(0) = \mathbf{0}$ can ensure the constrain condition in each iteration due to

$$\mathbf{C}^H\mathbf{G} = \mathbf{C}^H\mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{f} = \mathbf{f}. \tag{48}$$

Based on $\mathbf{w}(0) = \mathbf{0}$, we can calculate $\mathbf{w}(n+1)$ by

$$\mathbf{w}(n+1) = \left(\mathbf{P}^n(\mathbf{I}-\mu\mathbf{R}_{xx})^n + \cdots + \mathbf{I}\right)\mathbf{G}. \tag{49}$$

Because $\mathbf{P}$ is the orthocomplement matrix of $\mathbf{C}$, we have

$$\mathbf{C}^H\mathbf{P}^n = \mathbf{0}, n = 1,2,\cdots,N. \tag{50}$$

Premultiplying both sides of (49) by $\mathbf{C}^H$, we find that the weight vector in each iteration satisfies the following constrain condition

$$\mathbf{C}^H\mathbf{w}(n+1) = \mathbf{f}. \tag{51}$$

From the analysis above, we find that the choice of $\mathbf{w}(0) = \mathbf{0}$ is reasonable and it can give a good suboptimal solution of weight vector.

4.3 The Iteration Number N and $\bigtriangledown\delta$

The third parameter to be determined is the iteration number N. The iteration number N can be set preliminarily and the value of N must be big enough for obtaining the optimal weight vector $\mathbf{w}_{\text{opt}}$. Based on our experiences of many simulations, the iteration number N must be big enough to obtain a reasonable suboptimal weight vector. Empirically, any $N \geq 20$ is appropriate because a suboptimal weight vector can be acheived after several iterations in most cases. Such selection of N is justified subsequently by convergence analysis in Section 5 and simulation results in Section 6.

As a candidate parameter, the iterative error of weight vector $\bigtriangledown\delta$ is also an efficient way to control the termination condition of our proposed SIS-LCMV, AIS-LCMV, and RIAS-LCMV algorithms. The iterative error $\bigtriangledown\delta$ can be computed in the following way

$$\bigtriangledown\delta = \|\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)\|_2^2, \tag{52}$$

where $\hat{\mathbf{w}}(n)$ is the estimate of the nth iteration and $\|\cdot\|_2^2$ is the ℓ_2-norm. We set $\bigtriangledown\delta = 10^{-3}$ in our simulation unless stated otherwise. A smaller $\bigtriangledown\delta$ will take much time to get stable point.

5. Discussion

In this section, we evaluate the performance of our proposed algorithms from two aspects, i.e., convergence performance and computational complexity.

5.1 Convergence Analysis

It is obvious that the orthocomplement matrix $\mathbf{P}$ of $\mathbf{C}$ has the following properties

$$\mathbf{P}^n = \mathbf{P}, n = 1,2,\cdots. \tag{53}$$

Based on (49) and (53), with $\mathbf{w}(0) = \mathbf{0}$, the relationship between weight vectors $\mathbf{w}(n+1)$ and $\mathbf{w}(n)$ can be further rewritten as

$$\begin{aligned}\mathbf{w}(n+1) \;=\;& \mathbf{P}(\mathbf{I}-\mu\mathbf{R}_{xx})^{n+1}\mathbf{w}(0)\\ &+\mathbf{P}(\mathbf{I}-\mu\mathbf{R}_{xx})^n\mathbf{G} + \cdots + \mathbf{G}.\end{aligned} \tag{54}$$

Hence, (54) can be rewritten as

$$\begin{aligned}&\mathbf{w}(n+1) =\\ &\mathbf{P}\left((\mathbf{I}-\mu\mathbf{R}_{xx})^n + (\mathbf{I}-\mu\mathbf{R}_{xx})^{n-1} + \cdots + \mathbf{I}\right)\mathbf{G} + \mathbf{G}.\end{aligned} \tag{55}$$

Note that the formula of finite sum of matrix can be expressed as

$$\sum_{i=1}^{n}\mathbf{A}^i = \left(\mathbf{I}-\mathbf{A}^{n+1}\right)(\mathbf{I}-\mathbf{A})^{-1}. \tag{56}$$

Let $\mathbf{A} = \mathbf{I}-\mu\mathbf{R}_{xx}$ in (56), and (55) can be equivalently expressed as

$$\mathbf{w}(n+1) = \frac{1}{\mu}\mathbf{P}\left(\mathbf{I}-(\mathbf{I}-\mu\mathbf{R}_{xx})^n\right)\mathbf{R}_{xx}^{-1}\mathbf{G} + \mathbf{G}. \tag{57}$$

The first term on the right side of (57) is the *transient* component of the vector $\mathbf{x}(n)$, and the second term is the *steady-state* component. Equation (55) shows that the estimated suboptimal weight vector of our proposed algorithm includes two parts: fixed part and adaptive part. The fixed part is known as the aforementioned min-norm solution of constraint condition $\mathbf{C}^H\mathbf{w} = \mathbf{f}$. The adaptive part is $\frac{1}{\mu}\mathbf{P}\left(\mathbf{I}-\left(\mathbf{I}-\mu\mathbf{R}_{xx}\right)^n\right)\mathbf{R}_{xx}^{-1}\mathbf{G}$, which is related to the covariance matrix of the receiving data. To prove the convergence performance of our proposed methods, we only require to consider the extreme value of $\frac{1}{\mu}\mathbf{P}\left(\mathbf{I}-\left(\mathbf{I}-\mu\mathbf{R}_{xx}\right)^n\right)\mathbf{R}_{xx}^{-1}\mathbf{G}$ when μ lies in $0 < \mu < \frac{2}{\lambda_{\max}}$.

From the step size selection of (46), we know that if $n \to \infty$, item $\left(\mathbf{I}-\mu\mathbf{R}_{xx}\right)^n \to 0$ for any μ lies in $0 < \mu < \frac{2}{\lambda_{\max}}$. Therefore, (57) can be rewritten as

$$\mathbf{w}(n+1) \simeq \frac{1}{\mu}\mathbf{P}\mathbf{R}_{xx}^{-1}\mathbf{G}+\mathbf{G}. \tag{58}$$

Equation (58) tells us that the stable point of suboptimal weight vector is $\frac{1}{\mu}\mathbf{P}\mathbf{R}_{xx}^{-1}\mathbf{G}+\mathbf{G}$ when the step size lies in the range of $0 < \mu < \frac{2}{\lambda_{\max}}$. That is to say, our proposed algorithms are stable when finding the suboptimal weight vector.

5.2 Complexity Evaluation

In this section, we compare the computational complexity of our proposed algorithms, i.e., SIS-LCMV, AIS-LCMV, and RIAS-LCMV, with the aforementioned CF-LCMV [19], GSC-LCMV [1], and two other iterative algorithms, i.e., constrained affine-projection (CAP) algorithm [12], and low-complexity addition or removal of sensors/constraints in LCMV beamformers [17]. The computational analysis is based on the complexity of basic operations defined in Tab. 4. For the convenience of comparison, we summarize the computational complexities of them in Tab. 5. Among them, the computational complexities of CF-LCMV and GSC-LCMV can be derived directly from the closed-form solution of optimal weight. It should be noted that here we considered is only $L = 1$ case in CAP algorithm because it has the lowest computational complexity among all the cases of $L \geq 1$ (L is the data reuse number, refer [12] for details) to justify the fairness of the complexity comparison. Despite of multiple different updating algorithms proposed in, we only consider the typical sensor update incremental LCMV (SUI-LCMV) algorithm for the sake of simplicity. For the listed iterative algorithms, we only consider the computation in each iteration. Note that for our proposed algorithms, we do not need compute $\mathbf{P}$ and $\mathbf{G}$ in each iteration because they are computed in initializing step. Hence, the computation number of our proposed algorithms have nothing to do with D. Compared with other algorithms in Tab. 5, it is obvious that our proposed algorithms have lower computational complexity compared with other listed algorithms. Among them, AIS-LCMV has the lowest computational complexity.

Operation	Computations
Matrix mult. $(m \times n) \times (n \times p)$	mnp
Matrix inversioin $n \times n$	$\frac{2}{3}n^3$

Tab. 4. Complexity of basic operations.

Different LCMV algorithms	Computations
CF-LCMV	$\frac{2}{3}M^3 + \frac{2}{3}D^3 + 2M^2D - MD + D^2$
GSC-LCMV	$9\frac{2}{3}D^3 + 4M^2D + 9MD^2 - MD + D^2$
CAP	$\frac{2}{3}D^3 + D^2 + D + 4MD + 2M^2$
SUI-LCMV	$4M^2 + 2MD + 5D^2 - 6M + 11D + 2$
SIS-LCMV	$\frac{5}{2}M^2 - \frac{1}{2}M$
AIS-LCMV	$M^2 + 2M + 1$
RAIS-LCMV	$4M^2$

Tab. 5. Computational complexity of five listed LCMV algorithms.

6. Simulation and Results

In this section, we compare the performance of our proposed algorithms with CF-LCMV, GSC-LCMV, CAP, and SUI-LCMV algorithms from calculations, robustness and convergence.

6.1 Comparison of Computational Complexity

In this section, the computational complexity of five aforementioned algorithms is validated by simulation data. From Tab. 5, we note that CF-LCMV and GSC-LCMV are direct solutions without any iteration process and the computational complexities of them exceed the other listed algorithms by far. We do not plot the calculations of CF-LCMV and GSC-LCMV algorithms in the following simulations due to their heavy computation burden. We only compare the calculations of our proposed algorithms with other two existing iterative adaptive LCMV algorithms, named CAP, and SUI-LCMV, for the convenience of comparative.

Considered that the source number D is usually much smaller than the number of sensors M. So we only consider the case of $D \leq M$ without loss of generality. To test the relationship between the computational complexity of listed LCMV algorithms and the number of sensors, we fix the number of sources D and only change the number of sensors M. Fig. 1 gives the results of different number of sensors versus calculations of listed LCVM algorithms when the number of source D is five. Here the number of sensors varies from 20 to 100 with increment of 10. To test the relationship between and computational complexity of listed algorithms and the number of sources, we change the number of sources D from 5 to 15 and keep the number of sensors M varying as before to show the relationship between the calculations and the number of sources D. Figures 2 gives the results of calculations of listed LCMV algorithms when the number of sources D is 15.

From Fig. 1 and Fig. 2, we observe that our proposed SIS-LCMV, AIS-LCMV, and RAIS-LCMV algorithms have competitive computational complexity in both cases. For the three proposed LCMV algorithms, the calculations of AIS-LCMV are smaller than that of the other two algorithms. In the former case, the calculations of AIS-LCMV are a bit smaller than that of SUI-LCMV while the calculations of RAIS-LCMV are close to that of CAP. In the latter case, the calculations of AIS-LCMV are smaller than that of SUI-LCMV and the calculations of RAIS-LCMV are smaller than that of CAP. As the number of sources increases, the superiority of our proposed algorithms is remarkable.

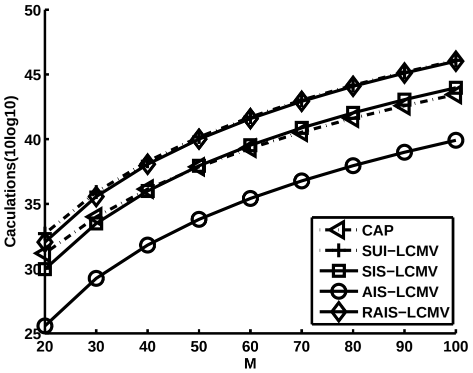

Fig. 1. Number of calculations versus different M for different LCMV algorithms with $D = 5$.

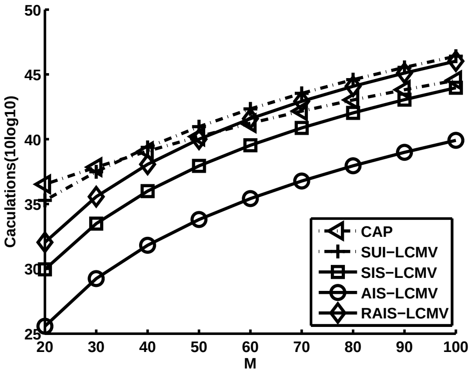

Fig. 2. Number of calculations versus different M for different LCMV algorithms with $D = 15$.

6.2 Accuracy in Beamforming and Interference Constraining

This experiment is carried out in a beamforming application. In this scenario, a uniform linear array with $M = 8$ antennas with element spacing equal to a half-wavelength was used in a system with $D = 4$ users. The signal of one user whose look-direction is 0° is of interest, and the other three signals whose incident angles are −20°, −40°, and −50°, respectively, are treated as interferences or jammers. In order to show the performance of our proposed algorithms, we also compare the performance of our algorithms with a constrained affine-projection (CAP) algorithm [12]. Because GSC-LCMV is only a different version of CF-LCVM and they have the same performance in beamforming ability, we omit GSC-LCMV in this simulation. Additionally, SUI-LCMV is a sensor update incremental algorithm and it is not appropriate for fixed sensor case, hence we do not consider its performance in this experiment. The parameters used in CAP algorithm in this experiment are, $L = 1$, $\delta = 10^{-4}$, and $\mu = 0.05$, respectively. The forget factor β used in RAIS-LCMV is 0.998 and the initial covariance matrix used in RAIS-LCMV is set to identity matrix, i.e., $\mathbf{R}_{xx}(0) = \mathbf{I}^{M\times M}$.

Firstly, we consider the performance of different algorithms against snapshot number. The signal-to-noise ratio (SNR) is set to 0 dB, and jammer-to-noise ratios (JNRs) of 10 dB are used. The beamforming performances versus the number of snapshots with fixed SNR, JNRs, and the number of sensors are depicted in Fig. 3 and Fig. 4, respectively. It is evident that the listed iterative adaptive algorithms, including our proposed algorithms and CAP algorithm, are more robust to small number of snapshots, while CF-LCMV is very sensitive to the number of snapshot and its performance is poor when fewer snapshots are available.

Secondly, we fix that the number of snapshots N is 20 and the number of sensors M is 8. Fig. 5 shows the beamforming performance of listed algorithms with SNR being 10 dB, and JNRs being 20 dB. Fig. 6 plots the beamforming performance of listed algorithms with SNR being −10 dB, and JNRs being −20 dB. It is evident that our proposed algorithms and CAP algorithm are robust to low SNR and JNRs, while CF-LCMV shows degraded performance in low SNR and JNRs case, as demonstrated in Fig. 6. From the four listed figures, we observe that our proposed algorithms have a competitive performance in both cases. Among SIS-LCMV, AIS-LCMV, and RAIS-LCMV, RAIS-LCMV has a better performance in trade-off robustness and tracking speed of LCMV beamformer. It is noted that the sidelobe level of RAIS-LCMV is lower than other algorithms, including CAP, in the four listed simulation cases. AIS-LCMV has the lowest computational complexity but is more sensitive to low SNR, JNRs, and small number of snapshots and sensors. SIS-LCMV is not an adaptive beamformer but it has a better performance than CF-LCMV in the same condition.

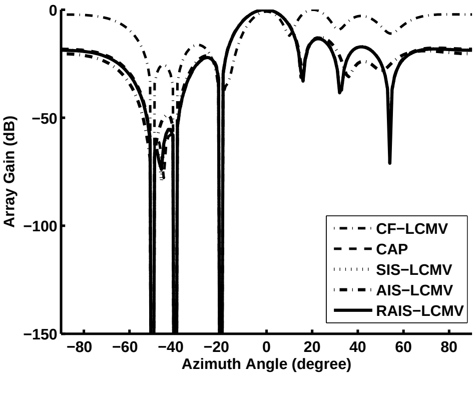

Fig. 3. Beamforming patterns of five listed algorithms with snapshot number $N = 20$, SNR = 0 dB, JNRs = 10 dB.

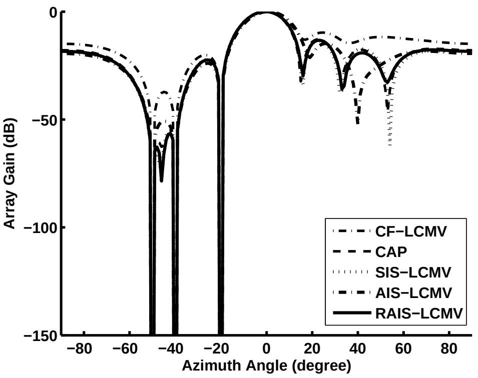

Fig. 4. Beamforming patterns of five listed algorithms with snapshot number $N = 200$, SNR = 0 dB, JNRs = 10 dB.

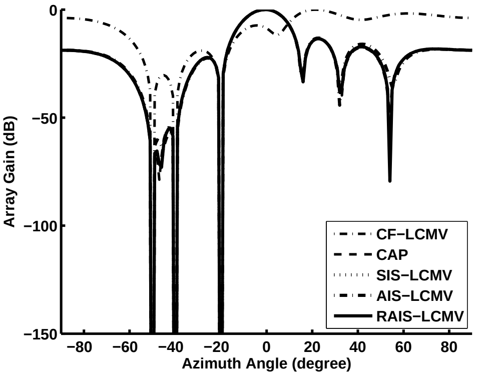

Fig. 5. Beamforming patterns of five listed algorithms with snapshot number $N = 20$, SNR = 10 dB, JNRs = 20 dB.

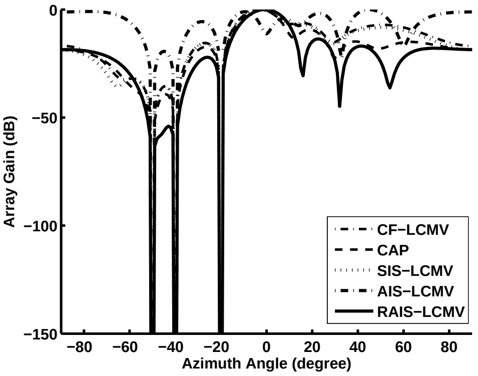

Fig. 6. Beamforming patterns of five listed algorithms with snapshot number $N = 20$, SNR = -10 dB, JNRs = -20 dB.

Finally, we also compared the performance of RAIS-LCMV with CMV-CCG (constrained minimum variance-conventional conjugate gradient) [9], CCM-MCG (constrained constant modulus-modified conjugate gradient) [9], and SM-CG (set membership-conjugate gradient) [11] algorithms. The number of snapshots, the number of sensors, SNR, JNRs and the directions of sources are all the same as the above case. The parameters of CMV-CCG, CCM-MCG, and SM-CG algorithms are listed in Tab. 6. These parameters are used in the simulations of [11] and [9]. It is worthy to point out that we just consider the fixed bound case of δ for the sake of simplicity. And the results are shown in Fig. 7. From the figure, we observe that our proposed RAIS-LCMV has a competitive performance in beamforming with respect to the other listed algorithms. Our proposed algorithm can work well with small snapshots number, while the CCM-MCG, CMV-CCG, and SM-CG need more snapshots number to obtain a good performance (more than 3000 in the simulation parts of the original literatures). Hence the speed of convergence of RAIS-LCMV is better than the above three algorithms because the steepest descent strategy is adopted.

Algorithms	Parameters
CMV-CCG	$K = M/2$, $\delta = 0.002$, $\lambda = 0.998$, $\gamma = 1$
CCM-MCG	$\tilde{\delta}_y = 0.002$, $\lambda_y = 0.998$, $\tilde{\eta} = 0.5$, $\gamma = 1$
SM-CG	$\delta = 0.8$, $\eta = 0.5$, $\gamma = 1$

Tab. 6. The simulation parameters of three algorithms.

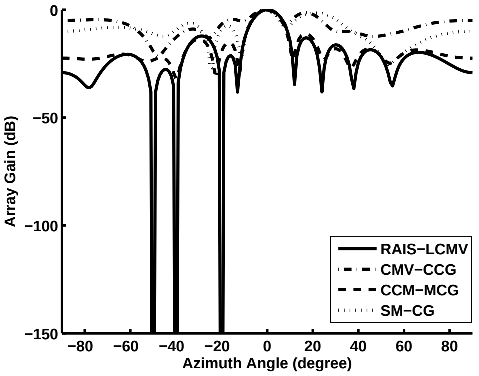

Fig. 7. Beamforming patterns of four listed algorithms with snapshot number $N = 20$, SNR = -10 dB, JNRs = -20 dB.

6.3 Convergence Speed

In this scenario, we fix the number of sensor element as eight. Four sources are the aforementioned directions-of-arrival. The number of snapshots is set to 1000 in order to test the ability of convergence of our proposed algorithms under different SNR and JNRs cases. We compare the convergence speed of our proposed algorithms with CAP algorithm. Fig. 8 and Fig. 9 show the convergence speed of our proposed algorithms and CAP algorithm under different SNR and JNRs. Note that the vertical axis in Fig. 8 and Fig. 9 is ℓ_2-norm of the difference of weight vectors between two adjacent iterations, i.e., $\|\Delta\mathbf{w}(k)\|$ is given by

$$\|\Delta\mathbf{w}(k)\| = \|\hat{\mathbf{w}}(k) - \hat{\mathbf{w}}(k-1)\|_2^2, \tag{59}$$

where $\hat{\mathbf{w}}(k)$ and $\hat{\mathbf{w}}(k-1)$ are the estimates of the kth and $(k-1)$th snapshots, respectively. We observe that, from both figures, our algorithms present a better performance in terms of convergence speed for both SNR and JNRs cases. Among them, the convergence performance of AIS-LCMV is slightly poorer than that of CAP algorithm due to adopting poor approximation of covariance matrix. The convergence performance of SIS-LCMV and RAIS-LCMV is superior to that of CAP algorithm no matter what SNR and

JNRs are in both cases. Also we see from the two figures that our proposed algorithms performs very similarly to the CAP algorithm in convergence speed: they all obtain a stable point after only several iterations, as demonstrated in Fig. 8 and Fig. 9. In other words, our algorithm needs only several snapshots to obtain the suboptimal weight vector under lower and higher SNR and JNRs cases.

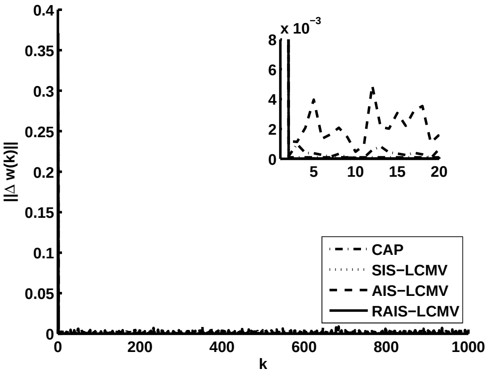

Fig. 8. Convergence performance of our proposed algorithms and CAP versus different JNRs, $N = 1000$, SNR = 10 dB, JNRs = 20 dB.

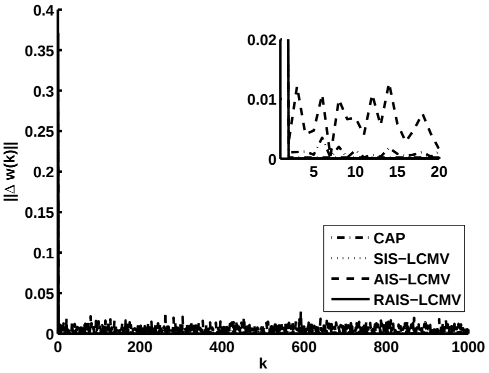

Fig. 9. Convergence performance of our proposed algorithms and CAP versus different JNRs, $N = 1000$, SNR = 0 dB, JNRs = 10 dB.

7. Conclusion

In this paper, we derive a low-complexity RAIS-LCMV algirithm using conjugate gradient (CG) optimization method. Our proposed algorithms can, in certain applications, substantially improve the robustness to low SNR, JNRs, and small number of snapshots.

Through theoretical analysis and simulation results, we show that our proposed algorithms can effectively overcome the dilemma of acquiring long observation time for stable covariance matrix estimates and short observation time to track dynamic behavior of targets. Parameters used in our algorithm are easily determined and the performance of our proposed algorithms are superior to existing iterative adaptive algorithms, such as CAP, SUI-LCMV, etc., in interference suppression, computational complexity, and robustness. Simulation results show the efficacy of our proposed algorithm.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant (61201277).

References

[1] BREED, B., STRAUSS, J. A short proof of the equivalence of LCMV and GSC beamforming. *IEEE Signal Processing Letters*, 2002, vol. 9, no. 6, p. 168–169. DOI: 10.1109/LSP.2002.800506

[2] WERNER, S., APOLINARIO JR, J. A., DE CAMPOS, M. L. R. On the equivalence of RLS implementations of LCMV and GSC processors. *IEEE Signal Processing Letters*, 2003, vol. 10, no. 12, p. 356–359. DOI: 10.1109/LSP.2003.818861

[3] DEMIRCIOGLU, E., SAZLI, M. Behavioral modeling of a C-band ring hybrid coupler using artificial neural networks. *Radioengineering*, 2010, vol. 19, no. 4, p. 645–652.

[4] BENESTY, J., CHEN, J., HUANG, Y., DMOCHOWSKI, J. On microphone-array beamforming from a MIMO acoustic signal processing perspective. *IEEE Transactions on Audio, Speech, and Language Processing*, 2007, vol. 15, no. 3, p. 1053–1065. DOI: 10.1109/TASL.2006.885251

[5] HAGER, W., ZHANG, H. The limited memory conjugate gradient method. *SIAM Journal on Optimization*, 2013, vol. 23, no. 4, p. 2150–2168. DOI: 10.1137/120898097

[6] RESENDE, L., ROMANO, J. M. T., BELLANGER, M. G. A fast least-squares algorithm for linearly constrained adaptive filtering. *IEEE Transactions on Signal Processing*, 1996, vol. 44, no. 5, p. 1168–1174. DOI: 10.1109/78.502329

[7] DAI, Y., KOU, C. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM Journal on Optimization*, 2013, vol. 23, no. 1, p. 296–320. DOI: 10.1137/100813026

[8] AL-BAYATI, A., AL-KAWAZ, R. A new hybrid WC-FR conjugate gradient-algorithm with modified secant condition for unconstrained optimization. *Journal of Mathematical and Computational Science*, 2012, vol. 2, no. 4, p. 937–966.

[9] WANG, L., DE LAMARE, R. C., Constrained adaptive filtering algorithms based on conjugate gradient techniques for beamforming. *IET Signal Processing*, 2010, vol. 4, no. 6, p. 686–697. DOI: 10.1049/iet-spr.2009.0243

[10] LANDAU, L., DE LAMARE, R. C., HAARD, M., Robust adaptive beamforming algorithms using the constrained constant modulus criterion. *IET Signal Processing*, 2014, vol. 8, no. 5, p. 447–457. DOI: 10.1049/iet-spr.2013.0166

[11] WANG, L., DE LAMARE, R. C., Set-membership constrained conjugate gradient adaptive algorithm for beamforming. *IET Signal Processing*, 2012, vol. 6, no. 8, p. 789–797. DOI: 10.1049/iet-spr.2011.0324

[12] WERNER, S., APOLINARIO, J. A., DE CAMPOS, M. L. R., DINIZ, P. S. R. Low-complexity constrained affine-projection algorithms. *IEEE Transactions on Signal Processing*, 2005, vol. 53, no. 12, p. 4545–4555. DOI: 10.1109/TSP.2005.859348

[13] FROST, O. L. III. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, 1972, vol. 60, no. 8, p. 926–935. DOI: 10.1109/PROC.1972.8817

[14] MERAD, L., BENDIMERAD, F., MERIAH, S., DJENNAS, S. Neural networks for synthesis and optimization of antenna arrays. *Radioengineering*, 2007, vol. 16, no. 1, p. 23–30.

[15] GOLUB, G., VAN LOAN, C. *Matrix Computations*, 4th ed. Baltimore (USA), JHU Press, 2012, vol. 3.

[16] GUO, X., WAN, Q., CHANG, C., LAM, E. Source localization using a sparse representation framework to achieve superresolution. *Multidimensional Systems and Signal Processing*, 2012, vol. 21, no. 4, p. 391–402. DOI: 10.1007/s11045-010-0119-y

[17] NGIAM, J., COATES, A., LAHIRI, A., PROCHNOW, B., LE, Q., NG, A. On optimization methods for deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, p. 265–272.

[18] MEIER, J., KELLEY, R., ISOM, B., YEARY, M., PALMER, R. Leveraging software-defined radio techniques in multichannel digital weather radar receiver design. *IEEE Transactions on Instrumentation and Measurement*, 2012, vol. 61, no. 6, p. 1571–1582. DOI: 10.1109/TIM.2011.2178670

[19] SCOTT, I., MULGREW, B. Sparse LCMV beamformer design for suppression of ground clutter in airborne radar. *IEEE Transactions on Signal Processing*, 1995, vol. 43, no. 12, p. 2843–2851. DOI: 10.1109/78.476428

[20] MARKOVICH-GOLAN, S., GANNOT, S., COHEN, I. Low-complexity addition or removal of sensors/constraints in LCMV beamformers. *IEEE Transactions on Signal Processing*, 2012, vol. 60, no. 3, p. 1205–1214.

[21] ZHANG, Y., ZHOU, Y., GUO, H. An iterative approach to MVDR beamforming with multi-cycle optimisation. *International Journal of Electronics*, 2013, vol. 100, no. 5, p. 616–625. DOI: 10.1080/00207217.2012.720940

About the Authors . . .

Xiansheng GUO is an associate professor in Dept. of Electronic Engineering, University of Electronic Science and Technology of China. He received his B.S. degree at Anhui Normal University in 2002, M.S. Degree at Southwest University of Science and Technology in 2005, and Ph.D. degree at University of Electronics Science and Technology of China (UESTC) in 2008. From September 2008 to August 2009, he served as a research associate in the Department Electrical and Electronic Engineering at the University of Hong Kong. From March 2012 to February 2014, he worked as a research fellow in Dept. of Electronic Engineering, Tsinghua University. His research interests include array signal processing, wireless localization, spectrum estimation, and software radio design.

Lei CHU was born in Hefei, China, in 1989. He is working towards his Master's degree in Electrical Engineering at University of Electronics Science and Technology of China, since September 2012. His research interests are in the radar signal and data processing, wireless sensor networks and indoor localization.

Baocang LI is currently pursuing the M.Sc. degree in Electrical Engineering at University of Electronics Science and Technology of China, since September 2013. His research interests include signal processing, wireless communication and indoor localization.