STUDY OF ROBUST ADAPTIVE BEAMFORMING ALGORITHMS BASED ON CONJUGATE GRADIENT TECHNIQUES

Hang Ruan * and Rodrigo C. de Lamare**

*Department of Electronics, The University of York, England, YO10 5BB
**CETUC, Pontifical Catholic University of Rio de Janeiro, Brazil
Emails: hr648@york.ac.uk, delamare@cetuc.puc-rio.br

ABSTRACT
This paper proposes low-complexity robust adaptive beamforming (RAB) techniques based on shrinkage methods. We firstly briefly review a Low-Complexity Shrinkage-Based Mismatch Estimation (LOCSME) batch algorithm to estimate the desired signal steering vector mismatch, in which the interference-plus-noise covariance (INC) matrix is also estimated with a recursive matrix shrinkage method. Then we develop low complexity adaptive robust version of the conjugate gradient (CG) algorithm to both estimate the steering vector mismatch and update the beamforming weights. A computational complexity study of the proposed and existing algorithms is carried out. Simulations are conducted in local scattering scenarios and comparisons to existing RAB techniques are provided.

Index Terms— robust adaptive beamforming, shrinkage methods, low complexity methods.

1. INTRODUCTION
Sensor array signal processing techniques and their applications to wireless communications, sensor networks and radar have been widely investigated in recent years. Adaptive beamforming is one of the most important topics in sensor array signal processing which has applications in many fields. However, adaptive beamformers may suffer performance degradation due to small sample data size or the presence of the desired signal in the training data. In practical environments, desired signal steering vector mismatch problems like signal pointing errors [13], imprecise knowledge of the antenna array, look-direction mismatch or local scattering may even lead to more significant performance loss [2].

1.1. Prior and Related Work
In order to address these problems, robust adaptive beamforming (RAB) techniques have been developed in recent years. Popular approaches include worst-case optimization [2], diagonal loading [3, 4], and eigen-decomposition [12, 13]. However, general RAB designs have some limitations such as their ad hoc nature, high probability of subspace swap at low SNR and high computational cost [5].

Further recent works have looked at approaches based on combined estimation procedures for both the steering vector mismatch and interference-plus-noise covariance (INC) matrix to improve RAB performance. The worst-case optimization methods in [2] solve an online semi-definite programming (SDP) while using a matrix inversion to estimate the INC matrix. The method in [8] estimates the steering vector mismatch by solving an online Sequential Quadratic Program (SQP) [6], while estimating the INC matrix using a shrinkage method [8]. Another similar method which jointly estimates the steering vector using SQP and the INC matrix using a covariance reconstruction method [9] has outstanding performance compared to other RAB techniques. However, their main disadvantages include the high computational cost associated with online optimization programming, the matrix inversion or reconstruction process, and slow convergence. In [11] and [15] we have introduced a Low-Complexity Shrinkage-Based Mismatch Estimation (LOCSME) method and a reduced-cost adaptive version [15] for robust beamforming, which estimate the steering vector mismatch by exploiting the cross-correlation vector between the sensor array data and the beamformer output.

1.2. Contributions
In this work, we develop an adaptive version of the LOCSME technique in [11] based on a conjugate gradient (CG) adaptive algorithm, resulting in the proposed LOCSME-CG algorithm. Different from the approach of LOCSME-SG, the LOCSME-CG algorithm not only updates the beamforming weights using a subspace approach [16]-[67], but can also estimates the mismatched steering vector, which sequentially performs the estimation of the mismatched vector by LOCSME in every snapshot for large systems [68]-[93],[94]-[110]. An analysis shows that LOCSME-CG requires lower complexity than the original LOCSME and has comparable complexity to LOCSME-SG. Simulations also show an excellent performance which benefits from the precise estimation of the steering vector.
The paper is organized as follows. The system model and problem statement are described in Section II. A review of the LOCSME method is provided in Section III whereas Section IV presents the proposed LOCSME-CG algorithm. Section V presents the simulation results. Section VI gives the conclusion.

2. SYSTEM MODEL AND PROBLEM STATEMENT

Consider a linear antenna array of $M$ sensors and $K$ narrow-band signals which impinge on the array. The data received at the $i$th snapshot can be modeled as

$$ x(i) = A(\theta)s(i) + n(i), $$

where $s(i) \in \mathbb{C}^{K \times 1}$ are uncorrelated source signals, $\theta = [\theta_1, \ldots, \theta_K]^T \in \mathbb{R}^K$ is a vector containing the directions of arrival (DoAs), $A(\theta) = [a(\theta_1) + e, \ldots, a(\theta_K)] \in \mathbb{C}^{M \times K}$ is the matrix which contains the steering vector for each DoA, and $e$ is the steering vector mismatch of the desired signal, $n(i) \in \mathbb{C}^{M \times 1}$ is assumed to be complex Gaussian noise with zero mean and variance $\sigma_n^2$. The beamformer output is

$$ y(i) = w^H x(i), $$

where $w = [w_1, \ldots, w_M]^T \in \mathbb{C}^{M \times 1}$ is the beamformer weight vector, where $(\cdot)^H$ denotes the Hermitian transpose. The optimum beamformer is computed by maximizing the SINR given by

$$ SINR = \frac{\sigma_s^2 |w^H a|^2}{w^H R_{i+n} w}. $$

3. LOCSME ROBUST BEAMFORMING ALGORITHM

The basic idea of LOCSME [11] is to obtain a precise estimate of the desired signal steering vector by exploiting cross-correlation vector between the beamformer output and the array observation data and then computing the beamforming weights.

3.1. Steering Vector Estimation

The cross-correlation between the array observation data and the beamformer output can be expressed as $d = E\{xy^*\}$. With assumptions that $|a_m w| \ll |a_i w|$ for $m = 2, \ldots, K$ and that the signal sources and that the system noise have zero mean while the desired signal is independent from the interferers and the noise, $d$ can be rewritten as $d = E\{\sigma_s^2 a w_1 + n^H w\}$. By projecting $d$ onto a predefined subspace [7], which collects all possible information from the desired signal, the unwanted part of $d$ can be eliminated. LOCSME also exploits prior knowledge which amounts to choosing an angular sector in which the desired signal is located, say $[\theta_1 - \theta_c, \theta_1 + \theta_c]$. The subspace projection matrix $P$ is given by

$$ P = [c_1, c_2, \ldots, c_p][c_1, c_2, \ldots, c_p]^H, $$

where $c_1, \ldots, c_p$ are the $p$ principal eigenvectors of the matrix $C$, which is defined by [6]

$$ C = \int_{\theta_1 - \theta_c}^{\theta_1 + \theta_c} a(\theta)a^H(\theta)d\theta. $$

In order to achieve a better estimation of the steering vector, an extension of the oracle approximating shrinkage (OAS) ([10]) technique is employed to obtain a more accurate estimate of the vector $d$. Let us define the sample correlation vector (SCV) in snapshot $i$ as

$$ \hat{l}(i) = \frac{1}{i} \sum_{k=1}^{i} x(k)y^*(k), $$

and its mean value as

$$ \hat{\nu}(i) = \sum_{i} \hat{l}(i)/M. $$

Then we aim to shrink the SCV towards its mean value $\hat{\nu}(i)$, which yields

$$ \hat{d}(i) = \hat{\rho}(i)\hat{\nu}(i) + (1 - \hat{\rho}(i))\hat{l}(i), $$

where $\hat{\rho}(i)$ represents the shrinkage coefficient $(\hat{\rho}(i) \in (0, 1))$. To find out the optimum $\hat{\rho}(i)$, we minimize the mean square
error (MSE) of $E[\|\hat{d}(i) - \hat{d}(i - 1)\|^2_2]$, which leads to
\[
\hat{\rho}(i) = \frac{(1 - \frac{2}{M}) \hat{d}^H(i - 1) \hat{l}(i - 1) + \sum \hat{d}(i - 1) \sum^* \hat{d}(i - 1)}{(1 - \frac{2}{M}) \hat{d}^H(i - 1) \hat{l}(i - 1) + (1 - \frac{2}{M}) \sum \hat{d}(i - 1) \sum^* \hat{d}(i - 1)}.
\]

Once the correlation vector $\hat{d}$ is obtained, the steering vector is estimated by
\[
\hat{a}_1(i) = \frac{\mathbf{P} \hat{d}(i)}{\|\mathbf{P} \hat{d}(i)\|_2}. \quad (12)
\]

### 3.2. Signal Power Estimation and Beamforming Weights

Following the description in [11], the desired signal power $\sigma_2^2$ is estimated by
\[
\hat{\sigma}_2^2(i) = \frac{|\hat{a}_1^H(i) x(i)|^2 - |\hat{a}_1^H(i) \hat{a}_1(i)| \sigma_n^2}{|\hat{a}_1^H(i) \hat{a}_1(i)|^2}, \quad (13)
\]
which has a linear complexity $O(M)$.

Once the steering vector and power of the desired signal are obtained, the INC matrix is also estimated by a matrix shrinkage method [11] and the weight vector is computed by
\[
\hat{w}(i) = \frac{\hat{R}_{i+n}^{-1}(i) \hat{a}_1(i)}{\hat{a}_1^H(i) \hat{R}_{i+n}^{-1}(i) \hat{a}_1(i)}, \quad (14)
\]
which has a computationally costly matrix inversion $\hat{R}_{i+n}^{-1}(i)$.

### 4. PROPOSED LOCSME-CG ALGORITHM

In this section, we develop a CG adaptive strategy based on LOCSME. We employ the same recursions as in LOCSME to estimate the steering vector and the desired signal power, whereas the estimation procedure of the beamforming weights is different. In order to avoid costly inner recursions, we let only one iteration be performed per snapshot[14]. Here we denote the CG-based weights and steering vector updated by snapshots as
\[
\hat{a}_1(i) = \hat{a}_1(i - 1) + \alpha_{a_1}(i) \mathbf{p}_{a_1}(i), \quad (15)
\]
\[
\mathbf{v}(i) = \mathbf{v}(i - 1) + \alpha_v(\mathbf{v}) \mathbf{p}_v(i). \quad (16)
\]
As can be seen, the subscripts of all the quantities for inner iterations are eliminated. Then, we employ the degenerated scheme to ensure $\alpha_{a_1}(i)$ and $\alpha_v(\mathbf{v})$ satisfy the convergence bound [14] given by
\[
0 \leq \mathbf{p}_{a_1}^H(i) \mathbf{g}_{a_1}(i) \leq 0.5 \mathbf{p}_{a_1}^H(i) \mathbf{g}_{a_1}(i - 1), \quad (17)
\]
\[
0 \leq \mathbf{p}_v^H(i) \mathbf{g}_v(i) \leq 0.5 \mathbf{p}_v^H(i) \mathbf{g}_v(i - 1). \quad (18)
\]

Instead of updating the negative gradient vectors $\mathbf{g}_{a_1}(i)$ and $\mathbf{g}_v(i)$ in iterations, now we utilize the forgetting factor to re-express them in one snapshot as
\[
\mathbf{g}_{a_1}(i) = (1 - \lambda) \mathbf{v}(i) + \lambda \mathbf{g}_{a_1}(i - 1) + \sigma_2^2(i) \alpha_{a_1}(i) \mathbf{v}(i) \mathbf{v}(i)^H \mathbf{p}_{a_1}(i) - \mathbf{x}(i) \mathbf{x}(i)^H \mathbf{a}_1(i), \quad (19)
\]
\[
\mathbf{g}_v(i) = (1 - \lambda) \mathbf{a}_1(i) + \lambda \mathbf{g}_v(i - 1) - \alpha_v(\mathbf{v})(\mathbf{R}(i) - \sigma_2^2(i) \mathbf{a}_1(i) \mathbf{a}_1(i)^H \mathbf{p}_{a_1}(i) - \mathbf{x}(i) \mathbf{x}(i)^H \mathbf{v}(i - 1). \quad (20)
\]
Pre-multiplying (19) and (20) by $\mathbf{p}_{a_1}^H(i)$ and $\mathbf{p}_v^H(i)$, respectively, and taking expectations we obtain
\[
E[\mathbf{p}_{a_1}^H(i) \mathbf{g}_{a_1}(i)] = E[\mathbf{p}_{a_1}^H(i) \mathbf{v}(i) - \mathbf{x}(i) \mathbf{x}(i)^H \mathbf{a}_1(i)]
+ \lambda E[\mathbf{p}_{a_1}^H(i) \mathbf{g}_{a_1}(i - 1)] - \lambda E[\mathbf{p}_{a_1}^H(i) \mathbf{v}(i)]
+ E[\alpha_{a_1}(i) \mathbf{p}_{a_1}^H(i) \sigma_2^2(i) \mathbf{v}(i) \mathbf{v}(i)^H \mathbf{p}_{a_1}(i)], \quad (21)
\]
\[
E[\mathbf{p}_v^H(i) \mathbf{g}_v(i)] = \lambda E[\mathbf{p}_v^H(i) \mathbf{g}_v(i - 1)] - \lambda E[\mathbf{p}_v^H(i) \mathbf{a}_1(i)]
- E[\alpha_v(\mathbf{v}) \mathbf{p}_v(i)(\mathbf{R}(i) - \sigma_2^2(i) \mathbf{a}_1(i) \mathbf{a}_1(i)^H \mathbf{p}_{a_1}(i))] \mathbf{p}_v(i), \quad (22)
\]
where in (22) we have $E[\mathbf{R}(i) \mathbf{v}(i - 1)] = E[\mathbf{a}_1(i)]$. After substituting (22) back into (18) we obtain the bounds for $\alpha_v(\mathbf{v})$ as follows
\[
\frac{(\lambda - 0.5) E[\mathbf{p}_v^H(i) \mathbf{g}_v(i - 1)] - \lambda E[\mathbf{p}_v^H(i) \mathbf{a}_1(i)]}{E[\mathbf{p}_v^H(i)(\mathbf{R}(i) - \sigma_2^2(i) \mathbf{a}_1(i) \mathbf{a}_1(i)^H \mathbf{p}_{a_1}(i))] \mathbf{p}_v(i)} \leq E[\alpha_v(\mathbf{v})],
\]
\[
\frac{(\lambda - 0.5) E[\mathbf{p}_v^H(i) \mathbf{g}_v(i - 1)] - \lambda E[\mathbf{p}_v^H(i) \mathbf{a}_1(i)]}{E[\mathbf{p}_v^H(i)(\mathbf{R}(i) - \sigma_2^2(i) \mathbf{a}_1(i) \mathbf{a}_1(i)^H \mathbf{p}_{a_1}(i))] \mathbf{p}_v(i)} \leq \lambda E[\mathbf{p}_v^H(i) \mathbf{g}_v(i - 1)] - \lambda E[\mathbf{p}_v^H(i) \mathbf{a}_1(i)]. \quad (23)
\]
Then we can introduce a constant parameter $\eta_v \in [0, 0.5]$ to restrict $\alpha_v(\mathbf{v})$ within the bounds in (23) as
\[
\alpha_v(\mathbf{v}) = \frac{\lambda E[\mathbf{p}_v^H(i) \mathbf{g}_v(i - 1)] - \lambda E[\mathbf{p}_v^H(i) \mathbf{a}_1(i)]}{\lambda E[\mathbf{p}_v^H(i)(\mathbf{R}(i) - \sigma_2^2(i) \mathbf{a}_1(i) \mathbf{a}_1(i)^H \mathbf{p}_{a_1}(i))] \mathbf{p}_v(i)} . \quad (24)
\]
Similarly, we can also obtain the bounds for $\alpha_{a_1}(i)$. For simplicity let us define $E[\mathbf{p}_{a_1}^H(i) \mathbf{g}_{a_1}(i - 1)] = A$, $E[\mathbf{p}_{a_1}^H(i) \mathbf{v}(i)] = B$, $E[\mathbf{p}_{a_1}^H(i) \mathbf{x}(i) \mathbf{x}(i)^H \mathbf{a}_1(i)] = C$ and $E[\mathbf{p}_{a_1}^H(i) \sigma_2^2(i) \mathbf{v}(i) \mathbf{v}(i)^H \mathbf{p}_{a_1}(i)] = D$. Substituting equation (21) into (17) gives
\[
\frac{\lambda(B - A) - B + C}{D} \leq E[\alpha_{a_1}(i)] \leq \frac{\lambda(B - A) - B + C + 0.5A}{D}, \quad (25)
\]
in which we can introduce another constant parameter $\eta_{a_1} \in [0, 0.5]$ to restrict $\alpha_{a_1}(i)$ within the bounds in (25) as
\[
E[\alpha_{a_1}(i)] = \frac{\lambda(B - A) - B + C + \eta_{a_1}A}{D}, \quad (26)
\]
or
\[
\alpha_a(i) = \frac{[p_a(i)^H(v(i) - p_a(i)g_a(i - 1)) - \beta_{a(i)} p_a(i) v(i)]}{\eta^2(i)p_a(i)^Hv(i) p_a(i)}. \tag{27}
\]

Then we can update the direction vectors \(p_a(i)\) and \(p_v(i)\) by
\[
p_a(i + 1) = g_a(i) + \beta_a(i)p_a(i), \tag{28}
\]
\[
p_v(i + 1) = g_v(i) + \beta_v(i)p_v(i), \tag{29}
\]
where \(\beta_a(i)\) and \(\beta_v(i)\) are updated by
\[
\beta_a(i) = \frac{[g_a(i) - g_a(i - 1)]^Hg_a(i)}{g_a(i)^Hg_a(i - 1)}, \tag{30}
\]
\[
\beta_v(i) = \frac{[g_v(i) - g_v(i - 1)]^Hg_v(i)}{g_v(i - 1)^Hg_v(i)}. \tag{31}
\]

Finally we can update the beamforming weights by
\[
w(i) = \frac{v(i)}{\alpha_a(i)v(i)}. \tag{32}
\]

To reproduce the proposed LOCSME-CG algorithm, equations (9)-(12),(13),(15),(16),(19),(20),(24),(27)-(32) are required. The forgetting factor \(\lambda\) and constant \(\eta\) for estimating \(\alpha(i)\) need to be adjusted to give its best performance.

A complexity analysis in terms of flops (total number of additions and multiplications) required by the proposed LOCSME-CG algorithm and the existing ones is compared in Table 1. The proposed LOCSME-CG algorithm avoids costly matrix inversion and multiplication procedures, which are unavoidable in the existing RAB algorithms. The LCWC algorithm of [12] requires \(N\) inner iterations per snapshot, which significantly varies in different snapshots (here we take \(N = 50\) as an averagely evaluated value). It is clear that LOCSME-CG has lower complexity in terms of the number of sensors \(M\), dominated by \(O(M^2)\), resulting in great advantages when \(M\) is large.

5. Simulation Results

The simulations are carried out under both coherent and incoherent local scattering mismatch [4] scenarios. A uniform linear array (ULA) of \(M = 12\) omnidirectional sensors with half wavelength spacing is considered. 100 repetitions are executed to obtain each point of the curves and a maximum of \(i = 300\) snapshots are unavoidable in the existing RAB algorithms. The LCWC algorithm of [12] requires \(N = 30M^3 + 3M^2 + 20M\) costly matrix inversion and multiplication procedures, which are unavoidable in the existing RAB algorithms. The LOCSME-CG outperforms the other algorithms and is very advantageous when \(\eta\) need to be adjusted to give its best performance.

A complexity analysis in terms of flops (total number of additions and multiplications) required by the proposed LOCSME-CG algorithm and the existing ones is compared in Table 1. The LCWC algorithm of [12] requires \(N\) inner iterations per snapshot, which significantly varies in different snapshots (here we take \(N = 50\) as an averagely evaluated value). It is clear that LOCSME-CG has lower complexity in terms of the number of sensors \(M\), dominated by \(O(M^2)\), resulting in great advantages when \(M\) is large.

### Table 1. Complexity Comparison

<table>
<thead>
<tr>
<th>RAB Algorithms</th>
<th>Flops</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCSME-CG</td>
<td>(13M^2 + 77M)</td>
</tr>
<tr>
<td>LOCSME-SG</td>
<td>(15M^3 + 30M)</td>
</tr>
<tr>
<td>Algorithm of [8]</td>
<td>(M^{3.0} + 7M^{2.5} + 5M^2 + 3M)</td>
</tr>
<tr>
<td>LOCSME-CG</td>
<td>(2M^2 + 4.5M^2 + 5M)</td>
</tr>
<tr>
<td>LCWC [7]</td>
<td>(100M^3 + 350M)</td>
</tr>
</tbody>
</table>

5.1. Mismatch due to Coherent Local Scattering

The steering vector of the desired signal affected by a time-invariant coherent local scattering effect is modeled as
\[
\mathbf{a}_1(i) = \mathbf{p} + \sum_{k=1}^{4} e^{j\phi_k} \mathbf{b}(\theta_k), \tag{33}
\]
where \(\mathbf{p}\) corresponds to the direct path while \(\mathbf{b}(\theta_k)(k = 1, 2, 3, 4)\) corresponds to the scattered paths. The angles \(\theta_k(k = 1, 2, 3, 4)\) are randomly and independently drawn in each simulation run from a uniform generator with mean \(10^\circ\) and standard deviation \(2^\circ\). The angles \(\phi_k(k = 1, 2, 3, 4)\) are independently and uniformly taken from the interval \([0, 2\pi]\) in each simulation run. Notice that \(\theta_k\) and \(\phi_k\) change from trials while remaining constant over snapshots.

Fig. 1 illustrate the performance comparisons of SINR versus snapshots and SINR versus SNR regarding the mentioned RAB algorithms in the last section under coherent scattering case. Specifically to obtain the SINR versus snapshots results, we select \(\lambda = 0.95, \eta = 0.2\) for LOCSME-CG. However, selection of these parameters may vary according to different input SNR as in the SINR versus SNR results. LOCSME-CG outperforms the other algorithms and is very close to the standard LOCSME.

5.2. Mismatch due to Incoherent Local Scattering

In the incoherent local scattering case, the desired signal has a time-varying signature and the steering vector is modeled by
\[
\mathbf{a}_1(i) = s_0(i)\mathbf{p} + \sum_{k=1}^{4} s_k(i)\mathbf{b}(\theta_k), \tag{34}
\]
where \(s_k(i)(k = 0, 1, 2, 3, 4)\) are i.i.d zero mean complex Gaussian random variables independently drawn from a random generator. The angles \(\theta_k(k = 0, 1, 2, 3, 4)\) are drawn independently in each simulation run from a uniform generator with mean \(10^\circ\) and standard deviation \(2^\circ\). This time, \(s_k(i)\) changes both from run to run and from snapshot to snapshot.
Different from the coherent scattering results, all the algorithms have a certain level of performance degradation due to the effect of incoherent local scattering model, in which case we have the extra system dynamics with the time variation, contributing to more environmental uncertainties in the system. However, over a wide range of input SNR values, LOCSME-CG still outperforms the other RAB algorithms. One point that needs to be emphasized is, most of the existing RAB algorithms experience significant performance degradation when the input SNR is high (i.e. around or more than 20dB), which is explained in [9] that the desired signal always presents in any kind of diagonal loading technique. However, LOCSME-CG improves the estimation accuracy, so that the high SNR degradation is successfully avoided as can be seen in Fig. 1 and Fig. 2.

6. CONCLUSION

This work proposed a low-complexity adaptive RAB algorithm, LOCSME-CG, developed from the LOCSME RAB method. We have derived recursions for the weight vector update with low complexity. We also have enabled the estimation for the mismatch steering vector inside the CG recursions to enhance the robustness. Simulation results have shown that LOCSME-CG achieves excellent output SINR performance and is suitable for operation in high input SNR.

7. REFERENCES


