Efficient Robust Adaptive Beamforming Algorithms for Sensor Arrays

Hang Ruan

Doctor of Philosophy

University of York

Electronics

March 2016

Abstract

Sensor array processing techniques have been an important research area in recent years. By using a sensor array of a certain configuration, we can improve the parameter estimation accuracy from the observation data in the presence of interference and noise. In this thesis, we focus on sensor array processing techniques that use antenna arrays for beamforming, which is the key task in wireless communications, radar and sonar systems.

Firstly, we propose a low-complexity robust adaptive beamforming (RAB) technique which estimates the steering vector using a Low-Complexity Shrinkage-Based Mismatch Estimation (LOCSME) algorithm. The proposed LOCSME algorithm estimates the covariance matrix of the input data and the interference-plus-noise covariance (INC) matrix by using the Oracle Approximating Shrinkage (OAS) method. Secondly, we present cost-effective low-rank techniques for designing robust adaptive beamforming (RAB) algorithms. The proposed algorithms are based on the exploitation of the cross-correlation between the array observation data and the output of the beamformer. Thirdly, we propose distributed beamforming techniques that are based on wireless relay systems. Algorithms that combine relay selections and SINR maximization or Minimum Mean-Square-Error (MMSE) consensus are developed, assuming the relay systems are under total relay transmit power constraint. Lastly, we look into the research area of robust distributed beamforming (RDB) and develop a novel RDB approach based on the exploitation of the cross-correlation between the received data at the relays and the destination and a subspace projection method to estimate the channel errors, namely, the cross-correlation and subspace projection (CCSP) RDB technique, which efficiently maximizes the output SINR and minimizes the channel errors. Simulation results show that the proposed techniques outperform existing techniques in various performance metrics.

Contents

Glossary   **135**

Bibliography   **137**

List of Figures

List of Tables

Acknowledgements

I would like to show great gratitude to my supervisors, Prof. Rodrigo C. de Lamare and Dr. Yury Zakharov for their support and guidance during my research, without which I would not be able to complete this work.

I'm also grateful to my family, who have been giving me both tremendous moral and financial support continuously and unwaveringly throughout my education, without which I would not achieve this far.

Finally, I would like to thank all my friends and colleagues in York and beyond, whose advice, friendship and goodwill has helped me immensely.

Declaration

This work has not previously been presented for an award at this, or any other, University. All work presented in this thesis is original to the best knowledge of the author. Some of the research presented in this thesis has resulted in some publications, which are listed in section 1.6, the Publications section, which is the at the end of Chapter 1. References and acknowledgements to work by other researchers have been given as appropriate.

Chapter 1

Introduction

Contents

1.1 Problem Statements

In important applications such as wireless communications, radar, sonar and biomedical processing, sensor array processing is indispensably and commonly used to filter signals in the space-time field by exploiting their spatial characteristics [1–3]. The advantages of sensor array processing include the following: the signal-to-interference-plus-noise ratio (SINR) is enhanced compared to using a single sensor, the directions of arrivals (DoAs) and waveforms of the emitted signal sources can be determined and parameters can be measured precisely in high dimensional spaces [1]. The main goal of sensor array signal processing is the estimation of parameters and extraction of information by fusing temporal and spatial information, captured via sampling a wavefield with a set of judiciously

1

placed sensors [1].

The topic we are particularly interested in sensor array processing is beamforming technique, which can be categorized as traditional or centralized beamforming and distributed beamforming techniques. In traditional beamforming, we aim to combine the measurements from an antenna array to maximize its gain in a specific direction [1]. In distributed beamforming, we have a relay system which can be also treated as an array composed by a set of antenna elements with distributed locations. The relay nodes (or antennas) are also independent processing units if there is no cooperation among themselves [88]. The advantages of distributed beamforming include an increase in the range of communications and a reduction in the network power consumptions, to overcome obstacles like poor channel and relay processing. However, estimation procedures of some crucial parameters like steering vectors, channel statistics and data covariance matrices can be challenging, especially if the implementations are considered under dynamic and unstable environments, which leads to the development of robust beamforming techniques. There has been an intensive research on robust beamforming methods, but still computational complexity and estimation precision are some unavoidable challenges.

1.2   Motivations

Both environmental effects and internal factors can affect the overall system performance. In traditional beamforming, the steering vector may suffer mismatch due to environmental uncertainties like look direction and pointing errors, source wavefront distortion, near-far field problem, signal fading and scattering, as well as non-environmental factors like imperfect array calibration and distorted antenna shape [7]. In distributed beamforming, the channel state information (CSI) is normally unknown (mismatched) in practical scenarios, which may be caused by limited channel feedback or outdate channel states [80]. In order to mitigate the effects of mismatch and preserve the precision of parameter estimation, we have developed novel methods and algorithms that aim to maximize the system performance and keep low computational complexities. Those methods and algorithms have been shown to obtain excellent performance in both simulations and analysis.

1.3 Contributions

- A cross-correlation and subspace projection method for estimating the desired signal steering vector mismatch is developed. The approach first computes the cross-correction vector of the system output and array observation data. The subspace is constructed as an eigensubspace. We show that projecting the cross-correlation vector onto the subspace gives superior estimation precision especially at medium to high input signal-to-noise ratios (SNRs). An iterative shrinkage method that approximates the cross-correlation vector and shrinkage coefficient is devised to improve the estimation accuracy of the steering vector mismatch. The above approaches are combined together and named as low-complexity shrinkage-based mismatch estimation (LOCSME) robust adaptive beamforming (RAB) algorithm.

- Adaptive algorithms that are based on stochastic gradient (SG) and conjugate gradient (CG) approaches for the batch LOCSME algorithm have been devised and named as LOCSME-SG, LOCSME-CCG and LOCSME-MCG, where CCG stands for conventional conjugate gradient and MCG stands for modified conjugate gradient. LOCSME-SG does not require matrix inversions or costly recursions to update the beamforming weights adaptively. In particular, the sample covariance matrix (SCM) is estimated only once using a knowledge-aided (KA) linear shrinkage algorithm along with the computation of the beamforming weights based on the estimated steering vector through SG recursions. LOCSME-CCG and LOCSME-MCG algorithms not only update the beamforming weights, but can also estimate the mismatched steering vector sequentially in every snapshot, to further improve estimation precision.

- Novel RAB algorithms that are based on low-rank and cross-correlation techniques is proposed. Firstly, a linear system (considered in high dimension) involving the mismatched steering vector and the statistics of the sampled data is constructed. Then we iteratively compute an orthogonal Krylov subspace whose model order is determined by both the minimum sufficient rank, which ensures no information loss when capturing the signal of interest (SoI) with interferers, and an execute-and-stop criterion, which automatically avoids overestimating the number of bases of the computed subspace. The estimated vector that contains the cross-correlation between the array observation data and the beamformer output is projected onto the

Krylov subspace, in order to update the steering vector mismatch, resulting in the proposed orthogonal Krylov subspace projection mismatch estimation (OKSPME) method.

- Based on the OKSPME method, we have also devised adaptive stochastic gradient (SG), CCG and MCG algorithms derived from the proposed optimization problems to reduce the cost for computing the beamforming weights, resulting in the proposed OKSPME-SG, OKSPME-CCG and OKSPME-MCG RAB algorithms. We remark that the steering vector is also estimated and updated using the CG-based recursions to produce an even more precise estimate. Derivations of the proposed algorithms are presented and discussed along with an analysis of their computational complexity. Moreover, we develop an analysis of the mean squared error (MSE) between the estimated and the actual steering vectors for the general approach of using a presumed angular sector associated with subspace projections. This analysis mathematically describes how precise the steering vector mismatch can be estimated. Upper and lower bounds are derived and compared with the existing approaches in the literature. Another analysis on the computational complexity of the proposed and existing algorithms is also provided.

- A joint maximum SINR (MSINR) distributed beamforming and restricted greedy search relay selection (RGSRS) algorithm with a total relay transmit power constraint is proposed, which iteratively performs relay selection and optimizes the beamforming weights at the relay nodes and maximizing the output SINR at the destination, provided that the second-order statistics of the CSI is perfectly known. Specifically, we devise a relay selection scheme based on a greedy search and compare it to other schemes like restricted random relay selection (RRRS) and restricted exhaustive search relay selection (RESRS). The RRRS scheme selects a fixed number of relays randomly from all relays. The RESRS scheme employs the exhaustive search method that runs every single possible combination among all relays aiming to obtain the set with the best SINR performance. The proposed RGSRS scheme is developed from a greedy search method with a specific optimization problem that works in iterations and requires SINR feedback from the destination.

- A novel robust distributed beamforming (RDB) technique is proposed. In this situation, the system CSI is imperfectly known at the relays, where the channel errors are modeled using an additive matrix perturbation method. We also assume that

4

there is no direct link between the signal sources and the destination. With a total relay transmit power constraint and an objective of maximizing the output SINR, we exploit the cross-correlation between the received data at the relays and the system output, a subspace projection method to estimate the channel errors and develop the cross-correlation and subspace projection (CCSP) RDB technique. A performance analysis regarding of the channel estimation MSE is provided for the proposed technique and simulations show an excellent performance as compared to previously reported algorithms.

1.4   Thesis Outline

Chapter 2 introduces the background theory relevant to the work presented in this thesis, which includes the topics of sensor array processing, traditional beamforming and related conventional adaptive beamforming algorithms, robust beamforming, steering vector mismatches and related RAB algorithms, and distributed beamforming (relay networking), cooperative relay systems, SINR maximization, relay selection and robust distributed beamforming.

Chapter 3 introduces a novel low complexity RAB algorithm named LOCSME and its system model, a cross-correlation and eigen-subspace projection approach as well as an iterative shrinkage method used for estimating the steering vector mismatch, novel SG and CG based adaptive algorithms that avoid costly matrix inversions and iteratively estimate the beamforming weight vector.

Chapter 4 introduces a novel low-rank RAB method named OKSPME based on dimensionality reduction techniques, which is based on the idea of constructing an orthogonal Krylov subspace and solving for the steering vector mismatch recursively where the model order is also determined automatically with constraints. SG and CG based adaptive algorithms based on the batch OKSPME method are devised to further reduce the computational complexity.

Chapter 5 presents the system model for distributed beamforming and novel relay selection algorithms combined with a maximum output SINR driven algorithm named as

MSINR.

Chapter 6 details a RDB technique that exploits the cross-correlation between the received data at the relays and the system output, a subspace projection method to estimate the channel errors and develop the CCSP RDB technique, which is superior in minimizing the channel mismatch and maximizing the system output SINR.

Chapter 7 gives a summary of this thesis and discuss potential future work.

1.5 Notations

In all expressions and equations of this thesis, lowercase non-bold letters represent scalar values whereas bold lowercase and upper case letters represent vectors and matrices, respectively. $(.)^*$, $(.)^T$, $(.)^{-1}$ and $(.)^H$ denote the complex conjugate operator, the transpose operator, matrix inversion operator and the Hermitian transpose operator, respectively. $|.|$, $||.||$, and $||.||_F$ denote the absolutely value of a scalar, the Euclidean norm of a vector or matrix and the Frobenius norm of a vector or matrix, respectively. $\odot$ represents the Schur-Hadamard product. $E[.]$ denotes the expectations. $.!$ denotes factorial operator. $tr(.)$ and $diag(.)$ denote the trace and the diagonal entry of a matrix, respectively. $sup.$ and $inf.$ denote the supreme and infimum bounds of a certain set. An identity matrix of size M is represented by $\mathbf{I}_M$.

1.6 Publications

Journals

H. Ruan, R. C. de Lamare, "Robust Adaptive Beamforming Using a Low-Complexity Shrinkage-Based Mismatch Estimation Algorithm," *IEEE Sig. Proc. Letters.*, Vol. 21, pp 60-64, 2013.

H. Ruan and R. C. de Lamare, "Robust Adaptive Beamforming Based on Low-Rank

and Cross-Correlation Techniques," *IEEE Trans. Signal Process.*, Vol. 64, Issue. 15, pp. 3919-3932, April 2016.

H. Ruan and R. C. de Lamare, "Low-Complexity Robust Adaptive Beamforming Algorithms Exploiting Shrinkage for Mismatch Estimation," *IET Signal Process.*, Vol. 10, Issue. 5, pp. 429-438, June 2016.

H. Ruan and R. C. de Lamare, "Robust Distributed Beamforming Based on Cross-Correlation and Subspace Projection Techniques," manuscript submitted to IEEE Transactions on Signal Processing.

Conferences

H. Ruan and R. C. de Lamare, "Low-Complexity Robust Adaptive Beamforming Based on Shrinkage and Cross-Correlation," *19th International ITG Workshop on Smart Antennas (WSA)*, pp 1-5, March 2015.

H. Ruan and R. C. de Lamare, "Low-Complexity Robust Adaptive Beamforming Algorithms Exploiting Shrinkage for Mismatch Estimation," *Sensor Signal Processing for Defence (SSPD)*, pp 1-5, 2015.

H. Ruan and R. C. de Lamare, "Robust adaptive beamforming based on low-rank and cross-correlation techniques," *Signal Processing Conference (EUSIPCO)*, pp 854-858, 2015.

H. Ruan and R. C. de Lamare, "Joint MSINR and Relay Selection Algorithms for Distributed Beamforming," *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, July 2016.

H. Ruan and R. C. de Lamare, "Joint MMSE Consensus and Relay Selection Algorithms for Distributed Beamforming," *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, July 2016.

H. Ruan and R. C. de Lamare, "Cross-Correlation and Subspace Projection Based Robust Distributed Beamforming Techniques," manuscript submitted to Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), 2017.

Chapter 2

Literature Review

Contents

2.1 Introduction

This Chapter briefly reviews the background knowledge in terms of sensor array processing on both centralized and distributed beamforming techniques, as well as some typical robust adaptive beamforming (RAB) techniques. Firstly, sensor array configurations are discussed and then the optimum minimum variance distortionless response (MVDR) bramformer is reviewed. Secondly, we describe the research area of RAB techniques, where the details of steering vector mismatch are introduced and some of the most important existing RAB methods and algorithms are reviewed and discussed. Lastly, we introduce the research topic of distributed beamforming, where the fundamentals of relay

networking, relay selection, centralized and cooperative relay systems, as well as robust distributed beamforming are introduced.

2.2 Sensor Array Processing

Sensor array processing aims to process data collected at sensor elements in order to extract useful information, suppress interference and estimate parameters. In order to describe a discrete-time sensor array model using linear algebra, two commonly used array geometries, namely, the uniform linear array (ULA) and the Uniform Circular Array (UCA), are briefly introduced.

2.2.1 Uniform Linear Array

The ULA is the simplest and the most commonly used sensor array structure. As shown in Fig. 2.1, M antenna elements are located in an axis with uniform spacing equal to d. A single source signal has DoA θ thus the corresponding steering vector is represented by $\mathbf{a}(\theta) = [a_1(\theta), \cdots, a_M(\theta)]^T$. The sensors take samples from the source signal at time instant i as $\mathbf{x}(i) = [x_1(i), \cdots, x_M(i)]^T$. The phase delay τ between two adjacent sensors is equal to $e^{-\frac{j2\pi d \sin \theta}{\lambda}}$, where λ is the wavelength of the wavefront. If we select the sensor at the edge which firstly receives the coming signal as the reference sensor, then the steering vector can be represented in terms of time delays as $\mathbf{a}(\theta) = [1, e^{-\frac{j2\pi d \sin \theta}{\lambda}}, \cdots, e^{-\frac{j(M-1)2\pi d \sin \theta}{\lambda}}]^T$. Therefore, the discrete-time signal model for a scenario with K source signals is given by

$$\mathbf{x}(i) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(i) + \mathbf{n}(i),$$

where $\mathbf{s}(i) \in \mathbb{C}^{K \times 1}$ are source signals, $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_K]^T \in \mathbb{R}^K$ is a vector containing the directions of arrivals (DoAs), $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \cdots, \mathbf{a}(\theta_K)] \in \mathbb{C}^{M \times K}$ is the matrix which contains the steering vector for each DoA, $\mathbf{n}(i) \in \mathbb{C}^{M \times 1}$ is assumed to be complex Gaussian noise with zero mean and variance σ_n^2.

2.2.2 Uniform Circular Array

The structure of UCA is shown in Fig. 2.2. The total number of M antenna sensors are uniformly located on a planar circle. The sensor elements are usually assumed isotropic. Therefore the element spacing can be obtained by the Sampling Theorem as $d \leqslant \frac{\lambda}{2}$. The discrete-time signal model for a UCA is given by $x_m(i) = e^{\frac{j2\pi d \sin\theta}{\lambda} \cos(\theta - \gamma_m) \sin(\phi)} s(i) + n_m(i)$, where $s(i)$ is the zero-mean and complex narrowband signal source with power σ_s^2, each of $n_m(i)$ is assumed to be zero-mean, spatially and temporally white Gaussian process and independent of $s(i)$. $\gamma_m = 2\pi(m-1)/N$ is the angle of the kth sensor measured counterclockwise from the x axis. The azimuth angle $\theta \in [0, 2\pi)$ is measured counterclockwise from the x axis and the elevation angle $\phi \in [0, \pi)$ is measured down from the z axis, which is perpendicular to the x-y plane [25].
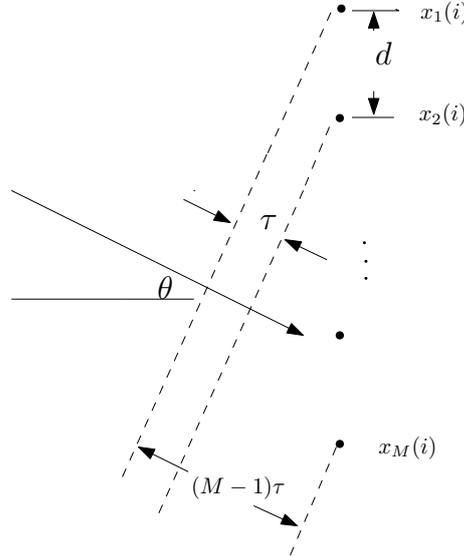
Figure 2.1: Uniform Linear Array

2.3 Beamforming

When referring to beamforming, we usually consider the traditional or centralized beamforming techniques, which are essentially signal processing techniques specified for using sensor arrays for directional signal transmission and reception. In traditional beamforming, we aim to combine the measurements from a uniformly configured antenna array to
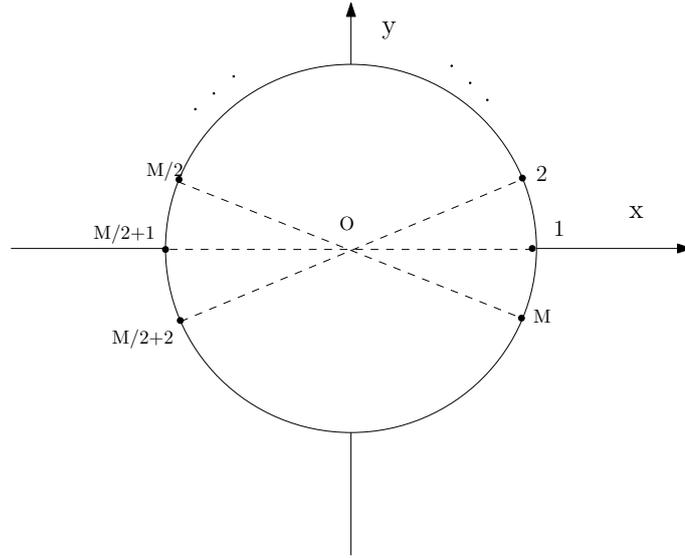
Figure 2.2: Uniform Circular Array

maximize its gain in a specific direction, by phasing the array in a certain angle so that the desired signal(s) are enhanced and the undesired signal(s) or interferer(s) are attenuated or rejected. The most popular optimum beamformer is known as the MVDR (or Capon) beamformer [1], which is introduced in the following subsection.

2.3.1   MVDR Optimum Beamformer

The MVDR optimum beamformer aims to retrieve or extract a desired signal (signal of interest (SoI)) in a given direction and frequency with unit gain, while the weights are chosen to minimize the output power with a single linear constraint, which preserves the SoI and attenuates the interferences and noise [4]. In this case, the desired signal is not distorted [5] and the beamforming weight vector $\mathbf{w} = [w_1, \cdots, w_M]^T$ is determined by

$$\mathbf{w}_{MVDR} = \underset{\mathbf{w}}{\operatorname{argmin}} E[|y|^2] \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a} = 1, \tag{2.1}$$

where $y = \mathbf{w}^H \mathbf{x}$ is the beamformer output $\mathbf{a}$ corresponds to the steering vector of the SoI. By employing the Lagrange multiplier method, we need to minimize the Lagrangian function as described by

$$\mathcal{L}(\mathbf{w}, \lambda) = E[|y|^2] + \lambda(\mathbf{w}^H \mathbf{a} - 1) + \lambda^*(\mathbf{a}^H \mathbf{w} - 1)$$
$$= E[|\mathbf{w}^H \mathbf{x}\mathbf{x}^H \mathbf{w}] + \mathcal{R}e\{\lambda\}(\mathbf{w}^H \mathbf{a} - 1) = E[|\mathbf{w}^H \mathbf{R}\mathbf{w}] + \mathcal{R}e\{\lambda\}(\mathbf{w}^H \mathbf{a} - 1), \tag{2.2}$$

where $\mathbf{x}$ is the array observation data and λ here is the Lagrange multiplier. Take the partial derivative of the above equation with respect to $\mathbf{w}$ and equal it to zero, the weight vector is obtained as

$$\mathbf{w}_{MVDR} = -\lambda \mathbf{R}^{-1}\mathbf{a}. \tag{2.3}$$

Substituting (2.3) into the linear constraint, λ is obtained as

$$\lambda = -(\mathbf{a}^H \mathbf{R}^{-1}\mathbf{a})^{-1}. \tag{2.4}$$

Combining (2.3) and (2.4) then we have the optimum MVDR weight as the following:

$$\mathbf{w}_{MVDR} = \frac{\mathbf{R}^{-1}\mathbf{a}}{\mathbf{a}^H \mathbf{R}^{-1}\mathbf{a}}. \tag{2.5}$$

2.4 Adaptive Beamforming Algorithms

In adaptive beamforming, the statistics (e.g. the covariance matrix) are usually unknown and may change over time and need to be estimated from the available data [1, 4]. There are several approaches to learning the unknown statistics. One approach is to estimate the covariance matrix of the antenna observation data (e.g. implemented with Sampled Matrix Inversion (SMI) [1], which results in a SMI beamformer) [1]. Another approach is based on an optimization problem and employs conventional adaptive algorithms (e.g. Stochastic Gradient (SG) and Conjugate Gradient (CG) [21, 23, 24]) to realize the adaptation of beamforming weights, which usually requires a low computational complexity but converges slower than the SMI beamformer [1].

Fig. 2.3 describes the systematic diagram of an adaptive beamformer. Different from the optimum beamformer, the covariance matrix $\hat{\mathbf{R}}$ is unknown and needs to be estimated in order to obtain the beamformer weights. One typical approach is to use the SMI method. In SMI, the covariance matrix is computed from the array observation data and referred to the Sampled Covariance Matrix (SCM) described by

$$\hat{\mathbf{R}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}(k)\mathbf{x}^H(k), \tag{2.6}$$

and its weight vector is computed as

$$\mathbf{w}_{SMI} = \frac{\hat{\mathbf{R}}^{-1}\mathbf{a}}{\mathbf{a}^H \hat{\mathbf{R}}^{-1}\mathbf{a}}. \tag{2.7}$$
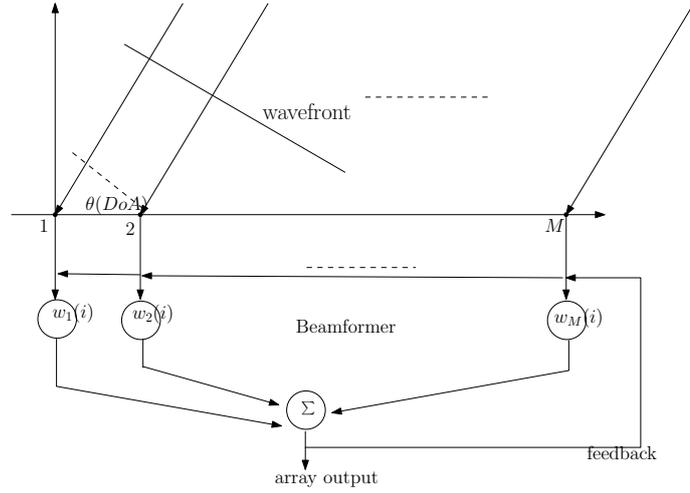
Figure 2.3: Adaptive Beamformer

However, in real applications, the SMI approach will usually include a diagonal loading (DL) term (i.e. $\sigma^2 \mathbf{I}_M$, where σ^2 is a constant and $\mathbf{I}_M$ is an identity matrix of size M) [1]. Therefore, (2.6) is reformulated as

$$\hat{\mathbf{R}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}(k)\mathbf{x}^H(k) + \sigma^2 \mathbf{I}_M. \tag{2.8}$$

The DL technique is an attractive modification to SMI beamformers because of its simplicity and potential performance improvement especially in a strong interference level situation. Even for a fixed DL, the loading level σ^2 needs to be appropriately selected, which can be done by evaluating the signal and interference levels [1].

2.4.1 MVDR-LMS Adaptive Algorithm

The least mean squares (LMS) adaptive algorithm belongs to the class of stochastic gradient (SG) methods. In this case, we consider deriving the LMS algorithm directly from the MVDR beamformer using the SG approach. To satisfy the MVDR beamforming criterion we have the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w}(i)}{\text{minimize}} \quad & \mathbf{w}^H(i)\mathbf{R}(i)\mathbf{w}(i) \\ \text{subject to} \quad & \mathbf{w}^H(i)\mathbf{a} = 1, \end{aligned} \tag{2.9}$$

14

where i denotes the time instant. Applying the SG recursion on its Lagrangian function [1], we have

$$\mathbf{w}(i + 1) = \mathbf{w}(i) - \mu \nabla \mathcal{L}(\mathbf{w}(i), \lambda) = \mathbf{w}(i) - \mu(\mathbf{x}(i)y^*(i) + \lambda \mathbf{a}), \tag{2.10}$$

where μ is the step size, λ is the Lagrange multiplier and ∇ denotes the gradient operator. Substituting (2.10) into the constraint of (2.9), we have

$$(\mathbf{w}^H(i) - \mu(\mathbf{x}(i)y^*(i) + \lambda \mathbf{a}))^H \mathbf{a} = 1. \tag{2.11}$$

Solving equation (2.11), we can obtain the Lagrange multiplier λ:

$$\lambda = \frac{-\mathbf{a}^H \mathbf{x}(i)y^*(i)}{\mathbf{a}^H \mathbf{a}}. \tag{2.12}$$

Substituting λ back into (2.10) and simplifying the result, the beamforming weight adaptation of MVDR-LMS algorithm is derived as

$$\mathbf{w}(i + 1) = \mathbf{w}(i) - \mu y^*(i)(\mathbf{I}_M + \frac{\mathbf{a}\mathbf{a}^H}{\mathbf{a}^H \mathbf{a}})\mathbf{x}(i). \tag{2.13}$$

2.4.2 MVDR-RLS Adaptive Algorithm

For the recursive least squares (RLS) algorithm, the optimization problem is defined as

$$\begin{aligned} \text{minimize} \quad & \epsilon_y(i) = \sum_{l=1}^{i} \mu^{i-l} |y(l)|^2 \\ \text{subject to} \quad & \mathbf{w}^H(i)\mathbf{a} = 1, \end{aligned} \tag{2.14}$$

where i is the current time index, μ is the forgetting factor and $y(l) = \mathbf{w}^H(i)\mathbf{x}(l)$. Again by using the method of Lagrange multipliers, a Lagrangian cost function $\mathcal{L}$ is introduced

$$\mathcal{L} = \sum_{l=1}^{i} \mu^{i-l} \mathbf{w}^H(i)\mathbf{x}(l)\mathbf{x}^H(l)\mathbf{w}(i) + \lambda(\mathbf{w}^H(i)\mathbf{a} - 1) + \lambda^*(\mathbf{a}^H \mathbf{w}(i) - 1). \tag{2.15}$$

Taking the partial derivative of (2.15) with respect to $\mathbf{w}(i)$ and equating the term to zero, we obtain

$$\mathbf{w}(i) = \frac{\mathbf{\Phi}^{-1}(i)\mathbf{a}}{\mathbf{a}^H \mathbf{\Phi}^{-1}(i)\mathbf{a}} = \Lambda(i)\mathbf{\Phi}^{-1}(i)\mathbf{a}, \tag{2.16}$$

where $\mathbf{\Phi}(i) = \sum_{l=1}^{i} \mu^{i-l}\mathbf{x}(l)\mathbf{x}^H(l)$ is the exponentially weighted sampled covariance matrix and $\Lambda(i) = (\mathbf{a}^H \mathbf{\Phi}^{-1}(i)\mathbf{a})^{-1}$. To realize the recursion, $\mathbf{\Phi}(i)$ is expressed as

$$\mathbf{\Phi}(i) = \mu\mathbf{\Phi}(i - 1) + \mathbf{x}(i)\mathbf{x}^H(i). \tag{2.17}$$

By using the Matrix Inversion Lemma for (2.17), we have the following

$$\mathbf{\Phi}^{-1}(i) = \mathbf{\Phi}^{-1}(i-1) - \frac{\mu^2 \mathbf{\Phi}^{-1}(i-1)\mathbf{x}(i)\mathbf{x}^H(i)\mathbf{\Phi}^{-1}(i-1)}{1 + \mu^{-1}\mathbf{x}^H(i)\mathbf{\Phi}^{-1}(i-1)\mathbf{x}(i)}. \tag{2.18}$$

Let us define the following matrix quantities

$$\mathbf{P}(i) = \mathbf{\Phi}^{-1}(i) \tag{2.19}$$

and

$$\mathbf{g}(i) = \frac{\mu^{-1}\mathbf{P}(i-1)\mathbf{x}(i)}{1 + \mu^{-1}\mathbf{x}^H(i)\mathbf{P}(i-1)\mathbf{x}(i)}, \tag{2.20}$$

where $\mathbf{g}(i)$ is the gain vector, then (2.18) can be reexpressed as

$$\mathbf{P}(i) = \mu^{-1}\mathbf{P}(i-1) - \mu^{-1}\mathbf{g}(i)\mathbf{x}^H(i)\mathbf{P}(i-1). \tag{2.21}$$

Multiplying both sides by $\mathbf{x}(i)$ and simplifying the terms, we have

$$\mathbf{g}(i) = \mathbf{P}(i)\mathbf{x}(i) = \mathbf{\Phi}^{-1}(i)\mathbf{x}(i). \tag{2.22}$$

The weight vector is computed as

$$\mathbf{w}(i) = \mathbf{\Lambda}(i)\mathbf{P}(i)\mathbf{a}. \tag{2.23}$$

After substituting (2.21) into (2.23), we then have the weight update equation as

$$\mathbf{w}(i) = \frac{\mathbf{\Lambda}(i)}{\mu\mathbf{\Lambda}(i-1)}(\mathbf{I} - \mathbf{g}(i)\mathbf{x}^H(i))\mathbf{w}(i-1), \tag{2.24}$$

which completes the MVDR-RLS adaptive algorithm [1].

2.4.3 MVDR-CG Adaptive Algorithm

We have already introduced the LMS and RLS algorithms under the MVDR criterion for adaptive beamforming. In fact, LMS has the advantage of simplicity but can not achieve good convergence performance as compared to RLS; while RLS demands a higher computational cost even though it has a high performance in convergence speed. In this subsection, we review the CG adaptive algorithm which efficiently overcomes the disadvantages in LMS and RLS algorithms. Based on the linear constrained minimum variance (LCMV) criterion [1,4], we start from the following optimization problem:

$$\mathbf{v} = \underset{\mathbf{v}}{\arg\min} J(\mathbf{v}), \tag{2.25}$$

where $\mathbf{v} \in \mathbb{C}^{M \times 1}$ is the CG-based weight vector. A convex cost function $J(\mathbf{v})$ can be described by [21, 23, 24]

$$J(\mathbf{v}) = \frac{1}{2}\mathbf{v}^H \mathbf{R} \mathbf{v} - \mathcal{R}e\{\mathbf{a}^H \mathbf{v}\}. \tag{2.26}$$

where $\mathcal{R}e\{.\}$ denotes the real part. The cost function is constructed in a quadratic form so that its gradient in terms of $\mathbf{v}$ describes the deviation of $\mathbf{a}$ from $\mathbf{R}\mathbf{v}$ [21]. By taking the gradient of (2.26) with respect to $\mathbf{v}$, equating it to a null vector and rearranging the expression we have

$$\mathbf{v} = \mathbf{R}^{-1}\mathbf{a}. \tag{2.27}$$

In order to derive the algorithm, we need to designate the snapshot index i and the iteration index k which is iteratively executed within each snapshot. Similar to the RLS algorithm, the data covariance matrix is estimated in a recursive fashion as:

$$\hat{\mathbf{R}}(i) = \lambda \hat{\mathbf{R}}(i-1) + \mathbf{x}(i)\mathbf{x}^H(i), \tag{2.28}$$

where λ is the forgetting factor, which is close to, but smaller than 1. Taking the gradient of (2.26) with respect to $\mathbf{v}_k(i)$ and choosing its negative direction, we obtain the negative gradient:

$$\mathbf{g}_k(i) = \mathbf{a} - \hat{\mathbf{R}}(i)\mathbf{v}_k(i). \tag{2.29}$$

The definition for the CG-based weight vector is given by [21, 23, 24]

$$\mathbf{v}_k(i) = \mathbf{v}_{k-1}(i) + \alpha_k(i)\mathbf{p}_k(i), \tag{2.30}$$

where $\alpha_k(i)$ is obtained by substituting (2.30) into (2.26) and taking the gradient with respect to $\alpha_k(i)$, which gives

$$\alpha_k(i) = \frac{\mathbf{g}_k^H(i)\mathbf{p}_k(i)}{\mathbf{p}_k^H(i)\hat{\mathbf{R}}(i)\mathbf{p}_k(i)}, \tag{2.31}$$

and the direction vector $\mathbf{p}_k(i)$ is updated by [21, 23, 24]

$$\mathbf{p}_k(i+1) = \mathbf{g}_k(i) + \beta_k(i)\mathbf{p}_k(i), \tag{2.32}$$

where $\beta_k(i)$ is given by [21, 23, 24]

$$\beta_k(i) = \frac{\mathbf{g}_k^H(i)\mathbf{g}_k(i)}{\mathbf{g}_{k-1}^H(i)\mathbf{g}_{k-1}(i)}. \tag{2.33}$$

After K iterations, the CG adaptive beamformer weight vector can be computed as

$$\mathbf{w}(i) = \frac{\mathbf{v}_K(i)}{\mathbf{a}^H \mathbf{v}_K(i)}. \tag{2.34}$$

Note that at the beginning of the next snapshot, $\mathbf{g}_k(i+1)$ and $\mathbf{p}_k(i+1)$ must be reset to $\mathbf{a} - \hat{\mathbf{R}}(i+1)\mathbf{v}_K(i)$ and $\mathbf{g}_k(i+1)$, respectively [1].

2.5 Robust Adaptive Beamforming

In this section, an explanation why RAB techniques are important for handling steering vector uncertainties and models for steering vector mismatch are provided. Furthermore, the most recent developed RAB algorithms are introduced and discussed.

2.5.1 Steering Vector Mismatch

When adaptive beamforming algorithms are applied to practical problems, the signal-to-interference-plus-noise ratio (SINR) performance may degrade when the data sample size is small and the convergence rate may reduce as the desired signal is presented in the training data [7]. Most importantly, the SINR performance of adaptive beamformers can suffer significant degradation because the underlying assumptions on the environment, signal sources or sensor array are usually non-ideal. This leads to a mismatch in the steering vector. To overcome the problem of steering vector mismatch, RAB techniques become a popular research area and various RAB algorithms have been developed.

In practical applications, different categories of steering vector mismatch include look direction and signal pointing errors, imperfect calibration and distorted antenna shape, manifold mismodeling due to source wavefront distortions, near-far field problem, signal fading and local scattering [7, 10].

Desired signal look direction mismatch is the simplest case for either modeling or handling. In this mismatch model, there exists an error for the DoA of the desired source signal (in some cases we also consider the interference signals). The error can be either a constant degree deviation or described by its statistical properties (e.g. uniform distribution within a certain range [14, 15]).

Near-far field mismatch is essentially caused by the spatial signature of the desired signal, which is assumed to be located in the near field of the antenna array, so that neither the array geometry nor the distance between the geometry center of the array and the signal source is negligible. In the case of ULA, the source is assumed to be located

on the line drawn from this geometrical center point in the normal direction to the array aperture, which is determined by the signal wavelength λ and the number of sensors M. The choice for the distance must be compatible with the array geometry parameters and also depends on M and λ [7].

In the local scattering mismatch case as shown in Fig. 2.4 [9], we consider the source signal distributed (or scattered) due to the multipath scattering effect caused by the presence of local scatterers [15]. This problem can be divided into two categories in terms of the signal signatures, called the coherent local scattering and incoherent local scattering.
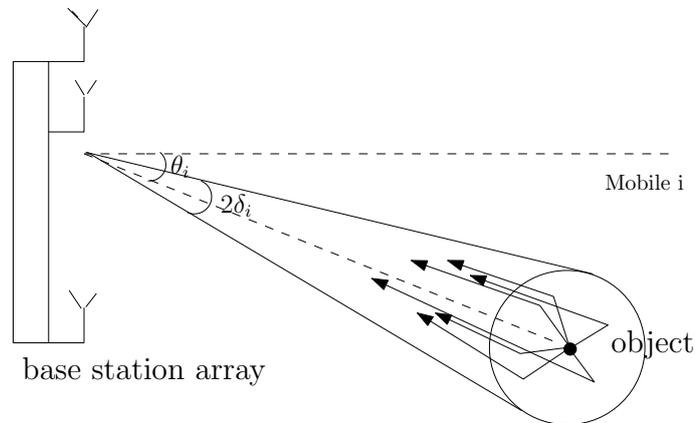
Figure 2.4: Local Scattering Effect of Detecting a Moving Object from a Base Station Array

In coherent local scattering [7], the source signal is assumed to have time-invariant signature and the corresponding steering vector is modeled as

$$\mathbf{a} = \mathbf{p} + \sum_{k=1}^{L} e^{j\varphi_k} \mathbf{b}(\theta_k), \tag{2.35}$$

where $\mathbf{p}$ corresponds to the direct path while $\mathbf{b}(\theta_k)(k = 1, \cdots, L)$ corresponds to the scattered paths. The angles $\theta_k(k = 1, \cdots, L)$ are randomly and independently drawn in each simulation run from a uniform generator with mean $10°$ and standard deviation $2°$. The angles $\varphi_k(k = 1, \cdots, L)$ are independently and uniformly taken from the interval $[0, 2\pi]$ in each simulation run. Notice that θ_k and φ_k change from trials while remaining constant over snapshots [7].

In incoherent local scattering [7], the source signal is assumed to have time-varying

signature and the corresponding steering vector is modeled by

$$\mathbf{a}(i) = s_0(i)\mathbf{p} + \sum_{k=1}^{L} s_k(i)\mathbf{b}(\theta_k), \tag{2.36}$$

where $s_k(i)(k = 0, \cdots, L)$ are i.i.d zero mean complex Gaussian random variables independently drawn from a random generator. The angles $\theta_k(k = 0, \cdots, L)$ are drawn independently in each simulation run from a uniform generator with fixed mean value and standard deviation. This time, $s_k(i)$ changes both from run to run and from snapshot to snapshot [7, 15].

2.5.2 Existing RAB Algorithms

In this subsection, we focus on the recently reported RAB algorithms. In [10], RAB design principles based on MVDR criterion have been discussed and summarized. These principles basically include: the generalized sidelobe canceller, diagonal loading [8, 9], eigenspace projection [18], worst-case optimization [7, 19] and steering vector estimation with presumed prior knowledge [11, 12].

The robust Capon beamformer (RCB) as discussed in [8] utilizes a diagonal loading method in which the loading factor is calculated based on a presumed uncertainty set for the SoI. It firstly started from an estimate of the desired signal power which is given by

$$\tilde{\sigma}^2 = \frac{1}{\mathbf{a}^H \mathbf{R}^{-1} \mathbf{a}}, \tag{2.37}$$

where $\mathbf{a}$ is the mismatched steering vector and $\mathbf{R}$ is the data covariance matrix. Furthermore, the RCB approach leads to the optimization problem given by

$$\begin{aligned} \min_{\mathbf{a}} \quad & \mathbf{a}^H \mathbf{R}^{-1} \mathbf{a} \\ \text{subject to} \quad & (\mathbf{a} - \bar{\mathbf{a}})^H \mathbf{C}^{-1}(\mathbf{a} - \bar{\mathbf{a}}) \leqslant 1, \end{aligned} \tag{2.38}$$

where both $\bar{\mathbf{a}}$ and $\mathbf{C}^{-1}$ are given. In the algorithm steps, an eigendecomposition technique and Newton's method are required to deliver an estimate of the loading factor λ, which further helps with the estimation of the desired signal power $\tilde{\sigma}^2$. This method has a complexity of $\mathcal{O}(M^3)$ and is difficult to achieve a satisfactory SINR performance.

Several of the most well-known online optimization programming based RAB approaches include the worst-case optimization method [7], the sequential quadratic programme (SQP) method [12] and the method in [11], all of which aim to solve online optimization programmes (i.e. second order cone programme (SOCP) and semi-definite programme (SDP)) with presumed prior knowledge so that to obtain an estimate for the desired signal steering vector. The worst-case optimization method and [11] use the same uncertainty constraint for the steering vector mismatch as in the RCB method. However, the SQP method uses a presumed steering vector that belongs to an uncertainty set $\mathcal{A} \triangleq \{\mathbf{p} + \mathbf{e}, \|\mathbf{e}\| \leqslant \epsilon\}$ where $\mathbf{p}$ is the presumed steering vector, $\mathbf{e}$ is the mismatch and ϵ is a known constant to restrict the uncertainty range. The presumed steering vector $\mathbf{p}$ is then iteratively updated by adding the orthogonal part of the error $\mathbf{e}$ and by enforcing that the updated version of $\mathbf{p}$ orthogonal to a subspace matrix $\mathbf{P}_\mathbf{p}^\perp$, which is also orthogonal to the actual steering vector $\mathbf{p} + \mathbf{e}$. This process can be expressed as an optimization programme described by

$$
\begin{aligned}
\min_{\mathbf{e}} \quad & (\mathbf{p} + \mathbf{e})^H \hat{\mathbf{R}}^{-1} (\mathbf{p} + \mathbf{e}) \\
\text{subject to} \quad & \mathbf{p}^H \mathbf{e} = 0, \mathbf{P}_\mathbf{p}^\perp (\mathbf{p} + \mathbf{e}) = \mathbf{0} \\
& (\mathbf{p} + \mathbf{e})^H \bar{\mathbf{C}} (\mathbf{p} + \mathbf{e}) \leqslant \mathbf{p}^H \bar{\mathbf{C}} \mathbf{p},
\end{aligned} \tag{2.39}
$$

where $\hat{\mathbf{R}}$ is the SCM and $\bar{\mathbf{C}} = \int_{\bar{\theta}} \mathbf{p}(\theta) \mathbf{p}^H(\theta) d\theta$, where $\bar{\theta}$ is the complement of θ, which is the angular sector in which the desired signal is assumed to be located, $\mathbf{p}(\theta)$ is the steering vector associated with a particular direction θ, [11, 12, 14]. However, because of the very high computational cost (at least $\mathcal{O}(M^{3.5})$) for the online optimization programmes, the methods of [7, 11, 12] lack computation efficiency. Additionally, the direct implementation of SCM in both the optimization objective function and computation for the weight may reduce the accuracy and final SINR performance.

Some recent design approaches have considered combining different design principles together to improve RAB performance. In the algorithms of [14, 15], the data covariance matrix and the desired signal steering vector are separately and sequentially estimated. In both of these algorithms, the steering vector is estimated using the SQP method. However, the data covariance matrix in [14] is estimated by a linear shrinkage model expressed as

$$
\tilde{\mathbf{R}} = \hat{\beta} \hat{\mathbf{R}} + \hat{\alpha} \mathbf{I}, \tag{2.40}
$$

where $\hat{\mathbf{R}}$ is the SCM, $\hat{\beta}$ and $\hat{\alpha}$ are positive shrinkage parameters which are derived by minimizing the mean squared error (MSE) $\mathrm{MSE}(\tilde{\mathbf{R}}) = \mathrm{E}[\|\tilde{\mathbf{R}} - \mathbf{R}\|^2]$, where $\mathbf{R}$ is the actual

covariance variance rather than the SCM [14]. Essentially this shrinkage method belongs to the class of diagonal loading approaches with the loading factor $\hat{\alpha}/\hat{\beta}$ computable and adaptable. The algorithm in [15] approaches the covariance matrix estimation in a totally different way, which directly estimates the interference-plus-noise covariance(INC) matrix $\tilde{\mathbf{R}}_{i+n}$ based on a INC matrix reconstruction method. It employs the Capon spatial spectrum estimator

$$\hat{P}(\theta) = \frac{1}{\mathbf{d}^H(\theta)\hat{\mathbf{R}}^{-1}\mathbf{d}(\theta)}, \tag{2.41}$$

where θ can be any possible angle, $\mathbf{d}(\theta)$ is the steering vector associated with angle θ and $\hat{\mathbf{R}}$ is the SCM. Furthermore, it is used for the INC matrix reconstruction as

$$\tilde{\mathbf{R}}_{i+n} = \int_{\bar{\theta}} \hat{P}(\theta)\mathbf{d}(\theta)\mathbf{d}^H(\theta)d(\theta). \tag{2.42}$$

The outstanding performance of [15] can be extremely close to the optimum SINR. However, it has high potential computational cost when the number of sample points taken within the angular sector $\bar{\theta}$ is large.

The common point of all the above algorithms introduced is the difficulty of estimating the steering vector in a computationally efficient way. Efforts have been made to avoid high complexity especially with online optimization programmes and an attractive algorithm named low-complexity mismatch estimation (LOCME) has been developed in [13]. LOCME aims to estimate the steering vector mismatch with a cost of $\mathcal{O}(M^3)$ and does not require any optimization programme or additional information from the steering vector. It describes the estimation of the array steering vector as the projection onto a predefined subspace of the correlation between the beamforming output signal and the array observation vector as [13]

$$\hat{\mathbf{a}} = \sqrt{M}\frac{\mathbf{Pd}}{\|\mathbf{Pd}\|}, \tag{2.43}$$

where $\mathbf{P}$ is the eigensubspace projection matrix which can be obtained if the angular range in which the steering vector is located is known and $\mathbf{d}$ is cross-correlation vector between the array observation data $\mathbf{x}$ and the beamformer output y, which is computed directly by $\mathbf{d} = E[\mathbf{x}y]$.

2.6 Distributed Beamforming

Distributed beamforming has been widely investigated in wireless communications and array processing in recent years [66–68]. It is key for situations in which the channels between the sources and the destination have poor quality so that devices cannot communicate directly and the destination relies on relays that receive and forward the signals [67]. Other advantages of distributed beamforming include the ability to significantly increase system power gain and save energy [66]. This section discusses the concepts and principles of distributed beamforming as well as the typical optimization criteria used. The concept and principles of relay selection and robust distributed beamforming are also presented and discussed.

2.6.1 Distributed Beamforming and Wireless Relay Networks

Distributed beamforming can be modelled as a relay network in which we consider a single or multiple (K) signal sources at the base station, a set of (M) distributed relays, each of which consists of only one sensor or antenna, and a destination. It is assumed that the quality of the channels between the signal sources and the destination is very poor so that direct communications is not possible. The M relays receive information transmitted by the signal sources and then retransmit to the destination as a beamforming procedure, in which a simple two-step amplify-and-forward (AF) protocol or decode-and-forward (DF) protocol can be applied for cooperative communications. In an AF protocol, the relay nodes send out amplified and phased versions of their received signals, which requires much less delays and relay power consumptions. In a DF protocol, the relay nodes operate as a black box that decode the received signal and re-encode them before transmitting, which ensures higher security but is less efficient in terms of delay and energy consumption. There are other protocols like compress-and-forward (CF) which involves quantization procedures and is not efficient in many situations, in CF, the relays quantize the received signal in one block and transmits the encoded version of the quantized received signal in the following block, which requires very high complexity if the quantization level is high or many blocks are used.

The relay network system can be modelled as shown in Fig. 2.5, assuming that an AF protocol is considered.
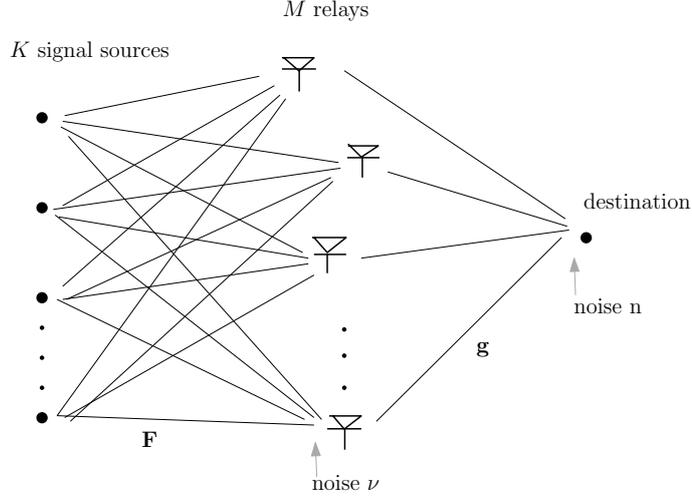
Figure 2.5: System Model of Relay Network

In the first step, the sources transmit the signals to the relays as

$$\mathbf{x} = \mathbf{F}\mathbf{s} + \boldsymbol{\nu}, \tag{2.44}$$

where $\mathbf{s} = [s_1, s_2, \cdots, s_K] \in \mathbb{C}^{1 \times K}$ are signal sources with zero mean, $[.]^T$ denotes the matrix transpose operator, $s_k = \sqrt{P_k}s$, $E[|s|^2] = 1$, P_k is the transmit power of the kth signal source, $k = 1, 2, \cdots, K$, s is the information symbol. Without loss of generality we can assume s_1 as the desired signal while the others are treated as interferers. $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K] \in \mathbb{C}^{M \times K}$ is the channel matrix between the signal sources and the relays, $\mathbf{f}_k = [f_{1,k}, f_{2,k}, \cdots, f_{M,k}]^T \in \mathbb{C}^{M \times 1}$, $f_{m,k}$ denotes the channel between the mth relay and the kth source ($m = 1, 2, \cdots, M$, $k = 1, 2, \cdots, K$). $\boldsymbol{\nu} = [\nu_1, \nu_2, \cdots, \nu_M]^T \in \mathbb{C}^{M \times 1}$ is the complex Gaussian noise vector at the relays and σ_ν^2 is the noise variance at each relay ($\nu_m \tilde{} \mathcal{C}N(0, \sigma_\nu^2)$), where $\mathcal{C}N(.)$ refers to complex Gaussian distribution. The vector $\mathbf{x} \in \mathbb{C}^{M \times 1}$ represents the received data at the relays. In the second step, the relays transmit $\mathbf{y} \in \mathbb{C}^{M \times 1}$ which is an amplified and phase-steered version of $\mathbf{x}$, which can be written as

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \tag{2.45}$$

where $\mathbf{W} = \mathrm{diag}[w_1, w_2, \cdots, w_M] \in \mathbb{C}^{M \times M}$ is a diagonal matrix whose diagonal entries denote the beamforming weights. The signal received at the destination is given by

$$z = \mathbf{g}^T \mathbf{y} + n, \tag{2.46}$$

24

where z is a scalar, $\mathbf{g} = [g_1, g_2, \cdots, g_M]^T \in \mathbb{C}^{M \times 1}$ is the complex Gaussian channel vector between the relays and the destination, n ($n \sim \mathcal{CN}(0, \sigma_n^2)$, and we assume that $\sigma_n^2 = \sigma_\nu^2$) is the noise at the destination and z is the received signal at the destination.

Note that both $\mathbf{F}$ and $\mathbf{g}$ are modeled as Rayleigh distributed (i.e., both the real and imaginary coefficients of the channel parameters have Gaussian distribution). Using the Rayleigh distribution for the channels, we also consider distance based large-scale channel propagation effects that include distance-based fading (or path loss) and shadowing. Distance-based fading represents how a signal is attenuated as a function of the distance and can be highly affected by the environment [69, 70]. An exponential based path loss model can be described by

$$\gamma = \frac{\sqrt{L}}{\sqrt{d^\rho}}, \tag{2.47}$$

where γ is the distance based path loss, L is the known path loss at the destination, d is the distance of interest relative to the destination and ρ is the path loss exponent, which can vary due to different environments and is typically set within 2 to 5 [69, 70], with a lower value representing a clear and uncluttered environment which has a slow attenuation and a higher value describing a cluttered and highly attenuating environment. Shadow fading describes the phenomenon where objects can obstruct the propagation of the signal attenuating the signal further, and can be modeled as a random variable with probability distribution given by [69, 70]

$$\beta = 10^{(\frac{\sigma_s \mathcal{N}(0,1)}{10})}, \tag{2.48}$$

where β is the shadowing parameter, $\mathcal{N}(0, 1)$ means the Gaussian distribution with zero mean and unit variance, σ_s is the shadowing spread in dB. The shadowing spread reflects the severity of the attenuation caused by shadowing, and is typically given between 0dB to 9dB [69, 70]. The channels modeled with both path-loss and shadowing are described by

$$\mathbf{F} = \gamma \beta \mathbf{F}_0, \tag{2.49}$$

$$\mathbf{g} = \gamma \beta \mathbf{g}_0, \tag{2.50}$$

where $\mathbf{F}_0$ and $\mathbf{g}_0$ denote the Rayleigh distributed channels without path-loss and shadowing [69, 70].

2.6.2 Optimization Criteria for Distributed Beamforming

Constraints are usually applied to relay systems in order to achieve desired objectives with environmental or self restrictions, which basically can be described as a single optimization problem. According to the literature, there are three different categories of optimization problems for relay systems. The first one involves the minimization of the total transmit power of the relays subject to constraints on the quality of service (QoS), which is usually referred to the system output SNR or SINR in distributed beamforming systems. In this scenario, the optimization problem can be expressed as

$$\min_{\mathbf{w}} \ P_T$$
$$\text{subject to} \quad \text{SNR(or SINR)} \geq \gamma, \tag{2.51}$$

where $\mathbf{w}$ is the beamforming weight vector, P_T is the total transmit power, $\gamma(\gamma > 0)$ is a predefined constant indicating the minimum required output SNR or SINR. It minimizes overall transmit power while ensuring the QoS is satisfied at the destination.

In the second scenario, the optimization problem is described as

$$\max_{\mathbf{w}} \ \text{SNR(or SINR)}$$
$$\text{subject to} \quad P_T \leq P_{T,max}, \tag{2.52}$$

which maximizes the output SNR or SINR while ensuring the total transmit power P_T does not exceed the threshold or the maximum allowable total transmit power $P_{T,max}$.

In the third scenario, we have the following optimization problem:

$$\max_{\mathbf{w}} \ SNR(or\ SINR)$$
$$\text{subject to} \quad P_m \leq P_{T,max} \ \text{for} \ m = 1, 2, \cdots, M, \tag{2.53}$$

where $P_{m,max}$ is the maximum allowable transmit power of the mth relay, from which each of the individual relay is constrained with a power limit. It should be emphasized that here we have M constraints in total instead of a single one.

2.6.3 Centralized and Cooperative Relay Networks with an MMSE Consensus Approach

For a centralized and cooperative relay network, we can always consider a minimum mean squared error (MMSE) consensus approach [71]. Assuming there are no interferers, then we define $\hat{s}_m = \phi_m x_m$ and

$$\phi_m = \arg \min_{\phi_m} E[|s_1 - \phi_m x_m|^2] = \frac{f_m^* P_1}{|f_m|^2 P_1 + \sigma_n^2}, \tag{2.54}$$

where P_1 is the desired signal power. Then we define $\tilde{s}_m = \frac{\hat{s}_m}{E[|\hat{s}_m|^2]}$ and the normalized relay weight $\tilde{w}_m$ as $\frac{w_m \phi_m}{E[|\hat{s}_m|^2]}$, so that the total transmission power can be expressed as $\sum_{m=1}^{M} E[|\tilde{w}_m \tilde{s}_m|^2] = \sum_{m=1}^{M} |\tilde{w}_m|^2$. Therefore, the following optimization problem under a total power constraint is considered

$$\min_{\tilde{w}_m} \sum_{m=1}^{M} \kappa_m E[|s_1 - g_m \tilde{w}_m \tilde{s}_m|^2]$$

$$\text{subject to } \sum_{m=1}^{M} |\tilde{w}_m|^2 \leq P_T, \tag{2.55}$$

where $\kappa_m > 0$ and the solution of (2.55) is given by

$$\tilde{w}_m = \frac{g_m^*}{\lambda/\kappa_m + |g_m|^2} \sqrt{\frac{\gamma_m}{\gamma_m + 1} P_1}, \tag{2.56}$$

where λ is the Langrange multiplier and $\gamma_m = \frac{|f_m|^2 P_1}{P_n}$.

In order to solve the optimization problem in (2.55), an MMSE consensus approach is employed to enable local information exchange and cooperations among the relay nodes. Suppose information is shared by all relays and each relay has an individual auxiliary beamforming vector denoted as $\tilde{\mathbf{w}}_m = [\tilde{w}_{1,m}, \tilde{w}_{2,m}, \cdots, \tilde{w}_{m,m}]^T$, then (2.55) can be reformulated as follows:

$$\min_{\{\tilde{\mathbf{w}}_m\}} \sum_{m=1}^{M} \kappa_m E[|s_1 - g_m \tilde{w}_{m,m} \tilde{s}_m|^2]$$

$$\text{subject to } ||\tilde{\mathbf{w}}_m||^2 \leq P_T, \tilde{\mathbf{w}}_m = \mathbf{w}, m = 1, 2, \cdots, M, \tag{2.57}$$

where the second constraint is a consensus constraint to impose all weight vectors to be the same. Then a dual-decomposition method is applied to decompose (2.57) to M sub-

optimization problems for each relay node as follows:

$$\min_{\tilde{\mathbf{w}}_m} \sum_{m=1}^{M} \kappa_m E[|s_1 - g_m \tilde{w}_{m,m} \tilde{s}_m|^2] \tag{2.58}$$

$$\text{subject to } ||\tilde{\mathbf{w}}_m||^2 \leq P_T, \tilde{\mathbf{w}}_m = \mathbf{w}.$$

Suppose each relay node is connected to a subset of relay nodes denoted by $\mathcal{M}_m$. The second constraint in (2.58) can be replaced by $\tilde{\mathbf{w}}_m = \tilde{\mathbf{w}}_q, q \in \mathcal{M}_m$ so that (2.58) can be reformulated as

$$\min_{\tilde{\mathbf{w}}_m} E[|s_1 - g_m w_m \tilde{s}_{m,1}|^2] + \lambda_m(i)(||\tilde{\mathbf{w}}_m||^2 - P_T)$$
$$+ \sum_{q \in \mathcal{M}_m} \tau_{m,q}^T (\tilde{\mathbf{w}}_m - \tilde{\mathbf{w}}_q), \tag{2.59}$$

where $\lambda_m(i)$ and $\tau_{m,q}$ are Lagrange multipliers.The proposed algorithmic solution relies on the computation of the optimal weights and Lagrange multipliers at the mth relay as

$$\tilde{w}_{t,m} = \begin{cases} \frac{g_m^*}{\lambda_m(i)+|g_m|^2} \left(\frac{g_m^*}{\lambda/\kappa_m+|g_m|^2} \sqrt{\frac{\gamma_m}{\gamma_m+1} P_1} - \frac{\sum_{q \in \mathcal{M}_m} \tau_{m,q;m}}{2} \right), \\ \text{if } t = m \\ -\frac{\sum_{q \in \mathcal{M}_m} \tau_{m,q;t}}{2\lambda_m(i)}, \\ \text{if } t \neq m \end{cases} \tag{2.60}$$

where $\tau_{m,q;t}$ denotes the tth element of $\tau_{m,q}$. The Lagrange multipliers are updated as follows

$$\lambda_m(i) = |\lambda_m(i-1) + \mu_\lambda(||\tilde{\mathbf{w}}_m||^2 - P_T)|, \tag{2.61}$$

$$\tau_{m,q}(i) = \tau_{m,q}(i-1) + \mu_\tau(\mathbf{u}_m - \mathbf{u}_q), \tag{2.62}$$

where μ_λ and μ_τ are step sizes with small positive values, $\mathbf{u}_m = [|w_{1,m}|, \cdots, |w_{M,m}|]^T$ and i is the time index.

2.6.4 Generalized Relay Networks with SNR Maximization Approaches

As discussed before, there are two criteria used for maximizing the system output SNR - the total relay transmit power constraint and individual relay power constraint. With the assumption that the second-order statistics of the CSI is perfectly known, [72] proved that

the system SNR maximization problem with a total relay transmit power constraint has a closed-form solution. It also showed that in the case of individual relay power constraints, the beamforming problem can be approximately written as a semidefinite programming (SDP) problem which can be efficiently solved using interior point methods.

The total relay transmit power can be rewritten as follows

$$P_T = \sum_{m=1}^{M} E[|y_m|^2] = \sum_{m=1}^{M} |w_m|^2 E[|x_m|^2] = \mathbf{w}^H \mathbf{D} \mathbf{w}, \tag{2.63}$$

where $\mathbf{w} = [w_1, w_2, \cdots, w_M]^T$, $\mathbf{D} = P_1 \text{diag}([E[|f_1|^2], E[|f_2|^2], \cdots, E[|f_M|^2]]) + P_n \mathbf{I}$. With the assumption that the relay noise $\nu_1, \nu_2, \cdots, \nu_M$, the destination noise n and the channel coefficients $g_1, g_2, \cdots, g_M$ are all independent from each other, the total noise power can be expressed by

$$P_{z,n} = E[\sum_{l,m=1}^{M} w_m w_l^* g_m g_l^*] E[|\nu_m|^2] + E[|n|^2] = \mathbf{w}^H \mathbf{Q} \mathbf{w} + P_n, \tag{2.64}$$

where $\mathbf{Q} = P_n E[\mathbf{g}\mathbf{g}^H]$. The power of the signal component P_1 can be expressed as

$$P_{z,1} = P_1 E[\sum_{l,m=1}^{M} w_m w_l^* f_m g_m f_l^* g_l^*] E[|s|^2] = \mathbf{w}^H \mathbf{R} \mathbf{w}, \tag{2.65}$$

where $\mathbf{R} = P_1 E[(\mathbf{f} \odot \mathbf{g})(\mathbf{f} \odot \mathbf{g})^H]$. Therefore, the optimization problem for the SNR maximization with total relay constraint can be expressed by [72]

$$\max_{\mathbf{w}} \frac{\mathbf{w}^H \mathbf{R} \mathbf{w}}{\mathbf{w}^H \mathbf{Q} \mathbf{w} + P_n} \tag{2.66}$$
$$\text{subject to } \mathbf{w}^H \mathbf{D} \mathbf{w} \leq P_T.$$

To solve the above optimization problem, the weight vector is rewritten as

$$\mathbf{w} = \sqrt{p} \mathbf{D}^{-1/2} \tilde{\mathbf{w}}, \tag{2.67}$$

where $\tilde{\mathbf{w}}$ satisfies $\tilde{\mathbf{w}}^H \tilde{\mathbf{w}} = 1$. Then (2.66) can be rewritten as

$$\max_{\{p, \tilde{\mathbf{w}}\}} \frac{p \tilde{\mathbf{w}}^H \tilde{\mathbf{R}} \tilde{\mathbf{w}}}{p \tilde{\mathbf{w}}^H \tilde{\mathbf{Q}} \tilde{\mathbf{w}} + P_n} \tag{2.68}$$
$$\text{subject to } ||\tilde{\mathbf{w}}||^2 = 1, p \leq P_T,$$

where $\tilde{\mathbf{R}} = \mathbf{D}^{1/2} \mathbf{R} \mathbf{D}^{1/2}$ and $\tilde{\mathbf{Q}} = \mathbf{D}^{1/2} \mathbf{Q} \mathbf{D}^{1/2}$. As the objective function in (2.68) increases monotonically with p regardless of $\tilde{\mathbf{w}}$, which means the objective function is

maximized when $p = P_T$, hence (2.68) can be simplified as:

$$\max_{\tilde{\mathbf{w}}} \frac{P_T \tilde{\mathbf{w}}^H \tilde{\mathbf{R}} \tilde{\mathbf{w}}}{P_T \tilde{\mathbf{w}}^H \tilde{\mathbf{Q}} \tilde{\mathbf{w}} + P_n}$$

$$\text{subject to } ||\tilde{\mathbf{w}}||^2 = 1,$$

(2.69)

or equivalently as

$$\max_{\tilde{\mathbf{w}}} \frac{P_T \tilde{\mathbf{w}}^H \tilde{\mathbf{R}} \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^H (P_n \mathbf{I} + P_T \tilde{\mathbf{Q}}) \tilde{\mathbf{w}}}$$

$$\text{subject to } ||\tilde{\mathbf{w}}||^2 = 1,$$

(2.70)

in which the objective function is maximized when $\tilde{\mathbf{w}}$ is chosen as the principal eigenvector of $(P_n \mathbf{I} + P_T \tilde{\mathbf{Q}})^{-1} \tilde{\mathbf{R}}$, which leads to the solution

$$\mathbf{w} = \sqrt{P_T} \mathbf{D}^{1/2} \mathcal{P}\{(P_n \mathbf{I} + \mathbf{D}^{1/2} \mathbf{Q} \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2} \mathbf{R} \mathbf{D}^{1/2}\},$$

(2.71)

and the maximum achievable SNR is given by

$$\text{SNR}_{max} = P_T \lambda_m ax\{(P_n \mathbf{I} + \mathbf{D}^{1/2} \mathbf{Q} \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2} \mathbf{R} \mathbf{D}^{1/2}\},$$

(2.72)

where λ_{max} is the maximum eigenvalue.

Differently, we consider the following optimization problem for the scenario of individual relay power constraint as discussed in [67, 72]:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^H \mathbf{R} \mathbf{w}}{\mathbf{w}^H \mathbf{Q} \mathbf{w} + P_n}$$

$$\text{subject to } \mathbf{D}_{mm} |w_m|^2 \leq P_m, \text{for } m = 1, 2, \cdots, M,$$

(2.73)

where P_m is the maximum allowable transmit power for the mth relay and $\mathbf{D}_{mm}$ refers to the mth diagonal entry of matrix $\mathbf{D}$. By defining $\mathbf{X} = \mathbf{w}\mathbf{w}^H$, (2.73) can be rewritten as

$$\max_{\mathbf{X}} \frac{\text{tr}(\mathbf{R}\mathbf{X})}{\text{tr}(\mathbf{Q}\mathbf{X}) + P_n}$$

$$\text{subject to } \mathbf{D}_{mm} \mathbf{X}_{mm} \leq P_m, \text{for } m = 1, 2, \cdots, M, \text{Rank}(\mathbf{x}) = 1, \mathbf{x} \succeq \mathbf{0},$$

(2.74)

where $\mathbf{X}_{mm}$ refers to the mth diagonal entry of $\mathbf{X}$. By using the idea of semidefinite relaxation and dropping the non-convex rank-one constraint, (2.75) can be reformulated as

$$\max_{\{\mathbf{X},t\}} t$$

$$\text{subject to } \text{tr}(\mathbf{X}(\mathbf{R} - t\mathbf{Q})) \geq P_n t,$$

(2.75)

$$\mathbf{X}_{mm} \leq P_m / \mathbf{D}_{mm}, \text{for } m = 1, 2, \cdots, M, \mathbf{x} \succeq \mathbf{0}.$$

It should be emphasized that for any value of t, the set of feasible $\mathbf{X}$ is convex. The problem (2.75) can be solved as a semidefinite programming (SDP) using interior points methods. The computational complexity and efficiency may vary based on the dynamics of solving an online convex programming with certain softwares.

2.6.5 Distributed Beamforming with Relay Selection

Distributed relays help by increasing system coverage and reducing power consumption. However, in most scenarios relays are either not ideally distributed in terms of locations or the channels involved with some of the relays have poor quality. Possible solutions can be categorized in two approaches. One is to adaptively adjust the power of each relay according to the qualities of its associated channels, known as adaptive power control or power allocation. Some power control methods based on channel magnitude and relative analysis has been studied in [73,75]. An alternative solution is to use relay selection, which selects a number of relays according to a criterion of interest while discarding the remaining relays. In [77], several optimum single-relay selection schemes and a multi-relay selection scheme using relay ordering based on maximizing the output SNR under individual relay power constraints are developed and discussed, but the beamforming weights are not optimized to enhance the SINR maximization. The work in [78] proposed a low-cost greedy search method for the uplink of cooperative direct sequence code-division multiple access systems, which approaches the performance of an exhaustive search. In [79], multi-relay selections algorithm have been developed to maximize the secondary receiver in a two-hop cognitive relay network.

From a general point of view, random relay selection is the simplest and most non-restrictive approach. With random relay selection, we choose the relays randomly. This can be done either by selecting a fixed number of random relays, or, with the number of selections to be decided randomly. We take a random decision for each relay that if it is to cooperate in the network with equal probability (i.e.,$p(\alpha_m = 0) = p(\alpha_m = 1) = 0.5$, where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_M]^T \in \{0, 1\}^{M \times 1}$ denotes the relay selection vector whose element equals either 0 which means the corresponding relay is unselected, or 1 which means the corresponding relay is selected). In case a fixed number of relays are required, the relay selection vector $\boldsymbol{\alpha}$ is also randomly chosen, however, with a fixed number of

ones and zeros (number of ones= M_{fix}, which is predefined).

Another popular approach is based on the exhaustive search method, in which we test every possible combination among all the relays, which means the change of status that each relay is chosen or not will contribute to a different possible combination. Also, if a minimum number of relays are required, then we can predefine M_{min} as the minimum required number of relays as an additional restriction. The exhaustive search method is expected to find out the best set of relays. However, the complexity can be extremely high depending on the total number of relays.

2.6.6 Robust Distributed Beamforming

In most practical scenarios, the channels that connect the signal sources and the relays may suffer quality degradation because of inevitable measurement, feedback delays, out-dated channel parameters, estimation and quantization errors in CSI [96–99] as well as propagation effects, which lead to an imperfect system CSI, which further results in unsatisfactory system performance or even system failure. Because of the above reasons, RDB techniques are hence in demand to reduce or mitigate the channel errors or uncertainties and preserve the relay system performance. In the literature, very limited work has been done in the research area of RDB. Most of the existing techniques adopt a worst-case optimization design to constraint the system SNR and aim to minimize the total relay transmit power as in a convex optimization problem [98, 103, 104]. Similar approaches also start with the same optimization problem and then reformulate it so that it can be solved with using a convex semi-definite programme (SDP) relaxation method [94, 96, 97, 108]. The intriguing work in [96] models the channel errors on their covariance matrices as a type of matrix perturbation. However, all of these existing techniques designate to minimize the total relay transmit power with constraints on the QoS (e.g. SNRs, SINRs). If we denote the channel uncertainties or errors as $\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_K] \in \mathbb{C}^{M \times K}$ (If that only the sources-to-destination channel $\mathbf{F}$ is considered for mismatch, whereas $\mathbf{g}$ is not affected), then we have

$$\hat{\mathbf{f}}_k = \mathbf{f}_k + \mathbf{e}_k, k = 1, 2, \cdots, K, \tag{2.76}$$

where $\mathbf{f}_k$ and $\hat{\mathbf{f}}_k$ are the kth true and mismatched channel components of $\mathbf{F}$, respectively. $\mathbf{e}_k$ for any $k = 1, \cdots, K$ follows a Gaussian distribution. Then, in a worst-case scenario

we assume that $\mathbf{e}_k$ falls in a hyper-spherical uncertainty set so that it satisfies the norm constraint $||\mathbf{e}_k|| \leq \epsilon_k$, where ϵ_k is a user-defined constant. The error uncertainly set can be hence written as

$$\mathcal{A}_k = \{\zeta_k | \zeta_k = \mathbf{f}_k + \mathbf{e}_k, ||\mathbf{e}_k|| \leq \epsilon_k\}, k = 1, \cdots, K. \tag{2.77}$$

Then the optimization problem that aims to minimize the total relay transmit power with a SNR contraint can be generally described by

$$\begin{aligned} \min_{\mathbf{w}} \ & P_T \\ \text{subject to} \ & SNR > \eta, \end{aligned} \tag{2.78}$$

where η is the minimum requirement for the system input SNR.

2.7 Summary

This chapter has firstly reviewed the background theories of sensor array processing and beamforming techniques. Then, introductions to the existing work in the literatures on the conventional adaptive beamforming algorithms and robust adaptive beamforming techniques have been presented. Lastly, the problem of distributed beamforming for wireless communication systems and the existing approaches and techniques have been discussed. This chapter is provided as a background support to the rest of the chapters where significant improvements and developments as well novel techniques are proposed. In the following chapters, we firstly introduce novel cost-efficient robust adaptive beamforming methods that based on recursive shrinkage methods, cross-correlation exploitations, subspace projections and low-rank techniques. Then, distributed beamforming and relay selection methods and robust distributed beamforming techniques are proposed.

Chapter 3

Low-Complexity Shrinkage-Based Mismatch Estimation (LOCSME) Algorithms for Robust Adaptive Beamforming

Contents

3.1 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 [18], imprecise knowledge of the antenna array, look-direction mismatch or local scattering may even lead to more significant performance loss [7].

3.1.1 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 [7], diagonal loading [8, 9, 37], and eigen-decomposition [18, 19]. 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 [11].

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 [7, 33–35] solve an online semi-definite programming (SDP) while using a matrix inversion to estimate the INC matrix. The method in [12] estimates the steering vector mismatch by solving an online sequential quadratic program (SQP) [12], while estimating the INC matrix using a shrinkage method [12]. Another similar method which jointly estimates the steering vector using SQP and the INC matrix using a covariance reconstruction method [15], presents 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.

35

3.1.2 Contributions

In this chapter, we develop an RAB algorithm with low complexity, which requires very little in terms of prior information, and has a superior performance to previously reported RAB algorithms. Our technique estimates the steering vector using a low-complexity shrinkage-based mismatch estimation (LOCSME) algorithm [31]. LOCSME estimates the covariance matrix of the input data and the INC matrix using the oracle approximating shrinkage (OAS) method. The only prior knowledge that LOCSME requires is the angular sector in which the desired signal steering vector lies. Given the sector, the subspace projection matrix of this sector can be computed in very simple steps [11–13, 15]. In the first step, an extension of the OAS method [16] is employed to perform shrinkage estimation for both the cross-correlation vector between the received data and the beamformer output and the received data covariance matrix. LOCSME is then used to estimate the mismatched steering vector and does not involve any optimization program, which results in a lower computational complexity. In a further step, we estimate the desired signal power using the desired signal steering vector and the received data. As the last step, a strategy which subtracts the covariance matrix of the desired signal from the data covariance matrix estimated by OAS is proposed to obtain the INC matrix. The advantage of this approach is that it circumvents the use of direction finding techniques for the interferers, which are required to obtain the INC matrix.

Then, we develop a stochastic gradient (SG) adaptive version of the LOCSME technique [31], denoted LOCSME-SG, which does not require matrix inversions or costly recursions to update the beamforming weights adaptively. In particular, the SCM is estimated only once using a knowledge-aided (KA) shrinkage [20, 32] algorithm along with the computation of the beamforming weights based on the estimated steering vector through SG recursions. Moreover, we also develop an adaptive LOCSME technique based on the conjugate gradient (CG) adaptive algorithm, resulting in CG type algorithms, denoted LOCSME-CCG and LOCSME-MCG. Different from LOCSME-SG, the CG type algorithms not only updates the beamforming weights, but can also estimate the mismatched steering vector, which sequentially performs the estimation of the mismatched vector by LOCSME in every snapshot. An analysis shows that both LOCSME-SG and LOCSME-CG achieve one degree lower complexity than the original LOCSME. Simulations also show an excellent performance which benefits from the precise estimation

provided by the shrinkage approach. Our contributions are summarized as follows:

- The derivation of LOCSME batch algorithm.

- The development of LOCSME type SG and CG algorithms.

- An investigation of the effect of shrinkage on the estimation accuracy of the algorithms.

- A study of the performance and the complexity of the proposed and existing algorithms.

This chapter is organized as follows. The system model and problem statement are described in Section 3.2. The derivation of the LOCSME algorithm and steering vector mismatch estimation are provided in Section 3.3. Section 3.4 presents the proposed adaptive LOCSME-SG algorithm whereas Section 3.5 presents the proposed LOCSME-CCG and LOCSME-MCG algorithms. Section 3.6 provides the shrinkage and complexity analyses. Section 3.7 presents the simulation results. Section 3.8 gives the summary.

3.2 System Model and Problem Statement

Consider a linear antenna array of M sensors and K narrowband signals which impinge on the array. The data received at the ith snapshot can be modeled as

$$\mathbf{x}(i) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(i) + \mathbf{n}(i), \tag{3.1}$$

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

$$y(i) = \mathbf{w}^H \mathbf{x}(i), \tag{3.2}$$

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

$$SINR = \frac{\sigma_1^2 |\mathbf{w}^H \mathbf{a}_1|^2}{\mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w}}. \tag{3.3}$$

where σ_1^2 is the desired signal power, $\mathbf{R}_{i+n}$ is the INC matrix. Assuming that the steering vector $\mathbf{a}_1$ is known precisely ($\mathbf{a}_1 = \mathbf{a}(\theta_1)$), then problem (3.3) can be cast as an optimization problem

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^H \mathbf{a}_1 = 1, \end{aligned} \tag{3.4}$$

which is known as the MVDR beamformer or Capon beamformer [1, 4]. The optimum weight vector is given by $\mathbf{w}_{opt} = \frac{\mathbf{R}_{i+n}^{-1} \mathbf{a}_1}{\mathbf{a}_1^H \mathbf{R}_{i+n}^{-1} \mathbf{a}_1}$. Since $\mathbf{R}_{i+n}$ is usually unknown in practice, it can be replaced by the SCM of the received data as

$$\hat{\mathbf{R}}_{i+n}(i) = \frac{1}{i} \sum_{k=1}^{i} \mathbf{x}(k) \mathbf{x}^H(k). \tag{3.5}$$

The problem we are interested in solving is how to design low-complexity robust adaptive beamforming algorithms that can preserve the SINR performance in the presence of uncertainties in the steering vector of a desired signal.

3.3   Batch LOCSME Algorithm

In this section, the proposed LOCSME algorithm for estimating the desired signal steering vector is introduced. The idea of LOCSME is to estimate the steering vector and the INC matrix separately as in previous approaches. The estimation of the steering vector is described as the projection onto a predefined subspace matrix of an iteratively shrinkage-estimated cross-correlation vector between the beamformer output and the array observation. The INC matrix is obtained by subtracting the desired signal covariance matrix from the data covariance matrix estimated by the OAS method.

3.3.1   Steering Vector Estimation using LOCSME

The cross-correlation between the array observation vector and the beamformer output can be expressed as

$$\mathbf{d} = E\{\mathbf{x}y^*\}. \tag{3.6}$$

We assume and emphasize that when $\mathbf{w}$ is determined such that the interference are sufficiently canceled such that they felled much below the noise floor and σ_1^2 that they could be considered to be negligible, in which case we have $|\mathbf{a}_m^H \mathbf{w}| \ll |\mathbf{a}_1^H \mathbf{w}|$ for $m = 2, \cdots, K$, all signal sources and the noise have zero mean, and the desired signal and every interferer are independent from each other. By substituting (3.1) and (3.2) into (3.6), we suppose the interferers are sufficiently canceled such that they fall much below the noise floor and the desired signal power is not affected by the interference so that $\mathbf{d}$ can be rewritten as

$$\mathbf{d} = E\{\sigma_1{}^2 \mathbf{a}_1^H \mathbf{w} \mathbf{a}_1 + \mathbf{n}\mathbf{n}^H \mathbf{w}\}. \tag{3.7}$$

In order to eliminate the unwanted part of $\mathbf{d}$ and obtain an estimate of the steering vector $\mathbf{a}_1$, $\mathbf{d}$ can be projected onto a subspace [13] that collects information about the desired signal. Here the prior knowledge amounts to providing an angular sector range in which the desired signal is located, say $[\theta_1 - \theta_e, \theta_1 + \theta_e]$. The subspace projection matrix $\mathbf{P}$ is given by

$$\mathbf{P} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_p][\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_p]^H, \tag{3.8}$$

where $\mathbf{c}_1, \cdots, \mathbf{c}_p$ are the p (which can be chosen manually by the user) principal eigenvectors of the matrix $\mathbf{C}$, which is defined by [12]

$$\mathbf{C} = \int\limits_{\theta_1 - \theta_e}^{\theta_1 + \theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta. \tag{3.9}$$

At this point, LOCSME will use the OAS method to compute the correlation vector $\mathbf{d}$ iteratively. The aim is to devise a method that estimates $\mathbf{d}$ more accurately with the help of the shrinkage technique. An accurate estimate of $\mathbf{d}$ can help to obtain a better estimate of the steering vector. Let us define

$$\hat{\mathbf{F}} = \hat{\nu}\mathbf{I}, \tag{3.10}$$

where $\hat{\nu} = \text{tr}(\hat{\mathbf{S}})/M$ and $\hat{\mathbf{S}} = \text{diag}(\mathbf{x}y^*) \in \mathbb{C}^{M \times M}$ is a diagonal matrix. Then, a reasonable tradeoff between covariance reduction and bias increase can be achieved by shrinkage

of $\hat{\mathbf{S}}$ towards $\hat{\mathbf{F}}$ [16] and subsequently using it in a vector shrinkage form, which results in

$$\hat{\mathbf{d}} = \hat{\rho}\,\text{diag}(\hat{\mathbf{F}}) + (1 - \hat{\rho})\,\text{diag}(\hat{\mathbf{S}}), \tag{3.11}$$

which is parameterized by the shrinkage coefficient $\hat{\rho}$. If we define $\hat{\mathbf{D}} = \text{diag}(\hat{\mathbf{d}})$ then the goal is to find the optimal value of $\hat{\rho}$ that minimizes the mean square error (MSE) of $E[\|\hat{\mathbf{D}}(i) - \hat{\mathbf{F}}(i-1)\|^2]$ in the ith snapshot, by taking the time index into account which leads to

$$\hat{\mathbf{d}}(i) = \hat{\rho}(i)\,\text{diag}(\hat{\mathbf{F}}(i)) + (1 - \hat{\rho}(i))\,\text{diag}(\hat{\mathbf{S}}(i)), \tag{3.12}$$

$$\hat{\rho}(i+1) = \frac{(1 - \frac{2}{M})\text{tr}(\hat{\mathbf{D}}(i)\hat{\mathbf{S}}^*(i)) + \text{tr}(\hat{\mathbf{D}}(i))\text{tr}(\hat{\mathbf{D}}^*(i))}{(i + 1 - \frac{2}{M})\text{tr}(\hat{\mathbf{D}}(i)\hat{\mathbf{S}}^*(i)) + (1 - \frac{i}{M})\text{tr}(\hat{\mathbf{D}}(i))\text{tr}(\hat{\mathbf{D}}^*(i))}, \tag{3.13}$$

where matrix $\hat{\mathbf{S}}(i)$ is estimated using the sample correlation vector (SCV) as

$$\hat{\mathbf{S}}(i) = \text{diag}\left(\frac{1}{i}\sum_{k=1}^{i}\mathbf{x}(k)y^*(k)\right). \tag{3.14}$$

As long as the initial value of $\hat{\rho}(0)$ is between 0 and 1, the iterative process in (3.12) and (3.13) is guaranteed to converge [16]. Once the correlation vector $\hat{\mathbf{d}}$ is obtained by the above OAS method, the steering vector is estimated by

$$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}, \tag{3.15}$$

where $\hat{\mathbf{a}}_1(i)$ gives the final estimate of the steering vector.

3.3.2 Interference-Plus-Noise Covariance Matrix Estimation

In order to compute the output SINR using (3.3), the INC matrix has to be estimated. The data covariance matrix (which contains the desired signal) is required. The SCM in (3.5) is necessary as a preliminary approximation. In the next step, similar to using OAS to estimate the cross-correlation vector $\hat{\mathbf{d}}$, the SCM is also processed with the OAS method as a further shrinkage estimation step. Let us define the following quantity

$$\hat{\mathbf{F}}_0 = \hat{\nu}_0\mathbf{I}, \tag{3.16}$$

where $\hat{\nu}_0 = \text{tr}(\hat{\mathbf{R}})/M$. Then, we use the shrinkage form again

$$\tilde{\mathbf{R}} = \hat{\rho}_0\hat{\mathbf{F}}_0 + (1 - \hat{\rho}_0)\hat{\mathbf{R}}. \tag{3.17}$$

By minimizing the MSE described by $E[\|\tilde{\mathbf{R}}(i) - \hat{\mathbf{F}}_0(i-1)\|^2]$, we obtain the following recursion

$$\tilde{\mathbf{R}}(i) = \hat{\rho}_0(i)\hat{\mathbf{F}}_0(i) + (1 - \hat{\rho}_0(i))\hat{\mathbf{R}}(i), \tag{3.18}$$

$$\hat{\rho}_0(i+1) = \frac{(1 - \frac{2}{M})\text{tr}(\tilde{\mathbf{R}}(i)\hat{\mathbf{R}}(i)) + \text{tr}^2(\tilde{\mathbf{R}}(i))}{(i+1-\frac{2}{M})\text{tr}(\tilde{\mathbf{R}}(i)\hat{\mathbf{R}}(i)) + (1 - \frac{i}{M})\text{tr}^2(\tilde{\mathbf{R}}(i))}. \tag{3.19}$$

Provided the initial value of $\hat{\rho}_0(0)$ is between 0 and 1, the iterative process in (3.18) and (3.19) is guaranteed to converge [16]. In order to eliminate the unwanted information of the desired signal in the covariance matrix and obtain the INC matrix, the desired signal power σ_1^2 must be obtained, which can be estimated directly using the desired signal steering vector. Let us rewrite the received data as

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{a}_k s_k + \mathbf{n}. \tag{3.20}$$

Pre-multiplying the above equation by $\mathbf{a}_1^H$, we have

$$\mathbf{a}_1^H \mathbf{x} = \mathbf{a}_1^H \mathbf{a}_1 s_1 + \mathbf{a}_1^H \left(\sum_{k=2}^{K} \mathbf{a}_k s_k + \mathbf{n} \right). \tag{3.21}$$

Here we assume that each of the interferers is orthogonal or approximately orthogonal to the desired signal. Specifically, the steering vector of each of the interferers is orthogonal ($\hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i) = 0$, $k = 2, 3, \cdots, K$), or approximately orthogonal ($\hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i) \ll \hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)$, $k = 2, 3, \cdots, K$) to the desired signal steering vector (i.e., $\hat{\mathbf{a}}_1(i)$), so that $\hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i)$ ($k = 2, 3, \cdots, K$) approaches zero and the term $\sum_{k=2}^{K} \hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i)s_k(i)$ in (3.21) can be neglected, resulting in

$$\mathbf{a}_1^H \mathbf{x} = \mathbf{a}_1^H \mathbf{a}_1 s_1 + \mathbf{a}_1^H \mathbf{n}. \tag{3.22}$$

Taking the expectation of $E[|\mathbf{a}_1^H \mathbf{x}|^2]$, we obtain

$$E[|\mathbf{a}_1^H \mathbf{x}|^2] = E[(\mathbf{a}_1^H \mathbf{a}_1 s_1 + \mathbf{a}_1^H \mathbf{n})^*(\mathbf{a}_1^H \mathbf{a}_1 s_1 + \mathbf{a}_1^H \mathbf{n})]. \tag{3.23}$$

If the noise is statistically independent from the desired signal, then we have

$$E[|\mathbf{a}_1^H \mathbf{x}|^2] = |\mathbf{a}_1^H \mathbf{a}_1|^2 |s_1|^2 + \mathbf{a}_1^H \mathbf{n}\mathbf{n}^H \mathbf{a}_1, \tag{3.24}$$

where $|s_1|^2$ is the desired signal power which can be replaced by its estimate $\hat{\sigma}_1^2$, $\mathbf{n}\mathbf{n}^H$ represents the noise covariance matrix $\mathbf{R}_n$ which can be replaced by $\sigma_n^2 \mathbf{I}_M$. Replacing $\mathbf{a}_1$ by its estimate $\hat{\mathbf{a}}_1(i)$ the desired signal power estimate is given by

$$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - \hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}. \tag{3.25}$$

As the last step, the desired signal covariance matrix is subtracted and the INC matrix is
given by

$$\tilde{\mathbf{R}}_{i+n}(i) = \tilde{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i). \tag{3.26}$$

The advantage of this step compared to SMI and existing methods is that it does not
require direction finding and is suitable for real-time applications. With the estimates for
the steering vector and the INC matrix, the beamformer is computed by

$$\hat{\mathbf{w}}(i) = \frac{\tilde{\mathbf{R}}_{i+n}^{-1}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\tilde{\mathbf{R}}_{i+n}^{-1}(i)\hat{\mathbf{a}}_1(i)}. \tag{3.27}$$

Table 3.1 summarizes LOCSME in steps. From a complexity point of view, the main
computational cost is due to the norm computations of the covariance matrix and the INC
matrix and weight vector computation. Each of these steps has a complexity of $\mathcal{O}(M^3)$.
Additionally, compared to the previous RAB algorithms in [11], [12], [12] and [15] which
have complexity equal or higher than $\mathcal{O}(M^{3.5})$, LOCSME has a lower cost ($\mathcal{O}(M^3)$).

3.4 Stochastic Gradient LOCSME Type Algorithm

In this section, we develop SG adaptive strategies based on the LOCSME robust beam-
forming technique, resulting in the proposed LOCSME-SG algorithm. CCG and MCG
based RAB algorithms named LOCSME-CCG and LOCSME-MCG are introduced in the
next section. These algorithms are developed for implementation purposes and are espe-
cially suitable for dynamic scenarios. In these adaptive algorithms, we employ the same
recursions as in LOCSME to estimate the steering vector and the desired signal power,
whereas the estimation procedures of the INC matrix and the beamforming weights are
different. In particular, LOCSME-SG employs a low-cost KA shrinkage method to esti-
mate the INC matrix and the weight vector update equation is derived from a reformulated
optimization problem.

Table 3.1: Proposed LOCSME Algorithm

Initialize:

$$\mathbf{C} = \int\limits_{\theta_1-\theta_e}^{\theta_1+\theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta$$

$[\mathbf{c}_1, \cdots, \mathbf{c}_p]$: p princical eigenvectors of $\mathbf{C}$

Subspace projection $\mathbf{P} = [\mathbf{c}_1, \cdots, \mathbf{c}_p][\mathbf{c}_1, \cdots, \mathbf{c}_p]^H$

$\hat{\mathbf{R}}(0) = \mathbf{0}; \hat{\mathbf{S}}(0) = \mathbf{0}; \mathbf{w}(0) = \mathbf{1};$

$\hat{\rho}(1) = \rho(0) = \hat{\rho}_0(1) = \rho_0(0) = 1;$

For each snapshot index $i = 1, 2, \cdots$:

$\hat{\mathbf{R}}(i) = \frac{1}{i}\sum\limits_{k=1}^{i}\mathbf{x}(k)\mathbf{x}^H(k)$

$\hat{\mathbf{S}}(i) = \text{diag}\left(\frac{1}{i}\sum\limits_{k=1}^{i}\mathbf{x}(k)y^*(k)\right)$

$\hat{\nu}(i) = \text{tr}(\hat{\mathbf{S}}(i))/M$

$\hat{\mathbf{F}}(i) = \hat{\nu}(i)\mathbf{I}$

$\hat{\mathbf{d}}(i) = \hat{\rho}(i)\text{diag}(\hat{\mathbf{F}}(i)) + (1 - \hat{\rho}(i))\text{diag}(\hat{\mathbf{S}}(i))$

$\hat{\mathbf{D}}(i) = \text{diag}(\hat{\mathbf{d}}(i))$

$\hat{\rho}(i + 1) = \frac{(1-\frac{2}{M})\text{tr}(\hat{\mathbf{D}}(i)\hat{\mathbf{S}}^*(i))+\text{tr}(\hat{\mathbf{D}}(i))\text{tr}(\hat{\mathbf{D}}^*(i))}{(i+1-\frac{2}{M})\text{tr}(\hat{\mathbf{D}}(i)\hat{\mathbf{S}}^*(i))+(1-\frac{i}{M})\text{tr}(\hat{\mathbf{D}}(i))\text{tr}(\hat{\mathbf{D}}^*(i))}$

$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}$

$\hat{\nu}_0(i) = \text{tr}(\hat{\mathbf{R}}(i))/M$

$\hat{\mathbf{F}}_0(i) = \hat{\nu}_0(i)\mathbf{I}$

$\tilde{\mathbf{R}}(i) = \hat{\rho}_0(i)\hat{\mathbf{F}}_0(i) + (1 - \hat{\rho}_0(i))\hat{\mathbf{R}}(i)$

$\hat{\rho}_0(i + 1) = \frac{(1-\frac{2}{M})\text{tr}(\tilde{\mathbf{R}}(i)\hat{\mathbf{R}}(i))+\text{tr}^2(\tilde{\mathbf{R}}(i))}{(i+1-\frac{2}{M})\text{tr}(\tilde{\mathbf{R}}(i)\hat{\mathbf{R}}(i))+(1-\frac{i}{M})\text{tr}^2(\tilde{\mathbf{R}}(i))}$

$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2-\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$

$\tilde{\mathbf{R}}(i) = \tilde{\mathbf{R}}(i) + \|\tilde{\mathbf{R}}(i)\|_2\mathbf{I}$

$\tilde{\mathbf{R}}_{i+n}(i) = \tilde{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)$

$\tilde{\mathbf{R}}_{i+n}(i) = \tilde{\mathbf{R}}_{i+n}(i)\frac{2\sigma_n^2}{\|\tilde{\mathbf{R}}_{i+n}(i)\|_2}$

$\hat{\mathbf{w}}(i) = \frac{\tilde{\mathbf{R}}_{i+n}^{-1}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\tilde{\mathbf{R}}_{i+n}^{-1}(i)\hat{\mathbf{a}}_1(i)}$

3.4.1 INC Matrix Estimation with a Knowledge-Aided Shrinkage Method

With the estimate of the desired signal power we subtract unwanted information of the interferences out from the array received data to obtain a modified array observation (MAO) vector. Consider a simple substraction step as

$$\mathbf{x}_{i+n}(i) = \mathbf{x}(i) - \hat{\sigma}_1(i)\hat{\mathbf{a}}_1(i). \tag{3.28}$$

Then the INC matrix can be estimated by

$$\hat{\mathbf{R}}_{i+n}(i) = \mathbf{x}_{i+n}(i)\mathbf{x}_{i+n}^H(i). \tag{3.29}$$

Now, we employ the idea of KA shrinkage method [20, 32] to help with our INC estimation. By applying a linear shrinkage model to the INC matrix, we have

$$\breve{\mathbf{R}}_{i+n}(i) = \eta(i)\mathbf{R}_0 + (1 - \eta(i))\hat{\mathbf{R}}_{i+n}(i), \tag{3.30}$$

where $\mathbf{R}_0$ is an initial guess for the INC matrix, $\eta(i)$ is the shrinkage parameter and $\eta(i) \in (0, 1)$. Here the shrinkage parameter is expected to be adaptively estimated. Employing an idea of adaptive filtering [20, 32], it is possible to set $y_{0f}(i) = [\mathbf{R}_0\hat{\mathbf{a}}_1(i)]^H\mathbf{x}(i)$ and $\hat{y}_f(i) = [\hat{\mathbf{R}}_{i+n}(i)\hat{\mathbf{a}}_1(i)]^H\mathbf{x}(i)$. To restrict $\eta(i)$ to a value greater than 0 and less than 1, a sigmoidal function is employed:

$$\eta(i) = \text{sgm}[\epsilon(i)] = \frac{1}{1 + e^{-\epsilon(i)}}, \tag{3.31}$$

where $\epsilon(i)$ is updated as [20, 32]

$$\epsilon(i + 1) = \epsilon(i) - \frac{\mu_\epsilon}{(\sigma_\epsilon + q(i))}(\eta(i)|y_{0f}(i) - \hat{y}_f(i)|^2$$
$$+ \mathcal{R}e\{(y_{0f}(i) - \hat{y}_f(i))\hat{y}_f^*(i)\})\eta(i)(1 - \eta(i)), \tag{3.32}$$

where μ_ϵ is the step size while σ_ϵ is a small positive constant, and $q(i)$ is updated as [20,32]

$$q(i + 1) = \lambda_q(i)(1 - \lambda_q)|y_{0f}(i) - \hat{y}_f(i)|^2, \tag{3.33}$$

where λ_q is a forgetting factor. The above steps formulate a completed INC matrix estimation in a single iteration.

3.4.2 Adaptive Computations of Beamforming Weight Vector

Now we resort to an SG adaptive strategy to reduce the complexity required by the matrix inversion. The optimization problem (3.4) can be re-expressed as

$$
\begin{aligned}
\underset{\mathbf{w}(i)}{\text{minimize}} \quad & \mathbf{w}^H(i)(\tilde{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) \\
\text{subject to} \quad & \mathbf{w}^H(i)\hat{\mathbf{a}}_1(i) = 1.
\end{aligned}
\tag{3.34}
$$

Then we can express the SG recursion as

$$
\mathbf{w}(i+1) = \mathbf{w}(i) - \mu \frac{\partial \mathcal{L}}{\partial \mathbf{w}(i)},
\tag{3.35}
$$

where $\mathcal{L} = \mathbf{w}^H(i)(\mathbf{x}(i)\mathbf{x}^H(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) + \lambda(\mathbf{w}^H(i)\hat{\mathbf{a}}_1(i) - 1)$. By substituting $\mathcal{L}$ into the SG equation (3.35) and letting $\mathbf{w}^H(i+1)\hat{\mathbf{a}}_1(i+1) = 1$, λ is obtained as

$$
\lambda = \frac{2(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i) - y(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i))}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)}.
\tag{3.36}
$$

By substituting λ back into (3.35) again, the weight update equation for LOCSME-SG is obtained as

$$
\mathbf{w}(i+1) = (\mathbf{I} - \mu\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) - \mu(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i) + y^*(i)(\mathbf{x}(i) - \frac{\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)})).
\tag{3.37}
$$

The adaptive SG recursion circumvents a matrix inversion when computing the weights using (3.27), which is unavoidable in LOCSME. Therefore, the computational complexity is reduced from $\mathcal{O}(M^3)$ in LOCSME to $\mathcal{O}(M^2)$ in LOCSME-SG. The proposed LOCSME-SG algorithm is summarized in Table 3.2.

3.5 Conjugate Gradient LOCSME Type Algorithms

In this section, we develop CG adaptive strategies based on the LOCSME robust beamforming technique, resulting in the LOCSME-CCG and LOCSME-MCG algorithms.

Table 3.2: Proposed LOCSME-SG Algorithm

Initialize:

$$\mathbf{C} = \int\limits_{\theta_1-\theta_e}^{\theta_1+\theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta$$

$[\mathbf{c}_1,\cdots,\mathbf{c}_p]$: p principal eigenvectors of $\mathbf{C}$

$\mathbf{P} = [\mathbf{c}_1,\cdots,\mathbf{c}_p][\mathbf{c}_1,\cdots,\mathbf{c}_p]^H$

$\hat{\mathbf{l}}(0) = \mathbf{0}; \mathbf{w}(0) = \mathbf{1}; \hat{\rho}(1) = \rho(0) = 1;$

For each snapshot index $i = 1, 2, \cdots$:

$\hat{\mathbf{l}}(i) = \frac{1}{i}\sum\limits_{k=1}^{i} \mathbf{x}(k)y^*(k)$

Steering vector mismatch estimation

$\hat{\nu}(i) = \sum_{m=1}^{M} l_m(i)/M$

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

$\hat{\rho}(i) = \frac{(1-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1)+\sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}{(i-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1)+(1-\frac{i}{M})\sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}$

$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}$

Desired signal power estimation

$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2-|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$

Computation of INC matrix

$\mathbf{x}_{i+n}(i) = \mathbf{x}(i) - \hat{\sigma}_1(i)\hat{\mathbf{a}}_1(i)$

$\hat{\mathbf{R}}_{i+n}(i) = \mathbf{x}_{i+n}(i)\mathbf{x}_{i+n}^H(i)$

$\breve{\mathbf{R}}_{i+n}(i) = \eta(i)\mathbf{R}_0 + (1 - \eta(i))\hat{\mathbf{R}}_{i+n}(i)$

$y_{0f}(i) = [\mathbf{R}_0\hat{\mathbf{a}}_1(i)]^H\mathbf{x}_{i+n}(i)$

$\hat{y}_f(i) = [\hat{\mathbf{R}}_{i+n}(i)\hat{\mathbf{a}}_1(i)]^H\mathbf{x}_{i+n}(i)$

$y_f(i) = \eta(i)y_{0f}(i) + (1 - \eta(i))\hat{y}_f(i)$

$\eta(i) = \frac{1}{1+e^{-\epsilon(i)}}$

$\epsilon(i + 1) = \epsilon(i) - \frac{\mu_\epsilon}{(\sigma_\epsilon+q(i))}(\eta(i)|y_{0f}(i) - \hat{y}_f(i)|^2$

$+\mathcal{R}\{(y_{0f}(i) - \hat{y}_f(i))\hat{y}_f^*(i)\})\eta(i)(1 - \eta(i))$

$q(i + 1) = \lambda_q(i)(1 - \lambda_q)|y_{0f}(i) - \hat{y}_f(i)|^2$

Computation of beamformer weights

$\mathbf{w}(i + 1) = (\mathbf{I} - \mu\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i)$

$-\mu(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i) + y^*(i)(\mathbf{x}(i) - \frac{\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)}))$

End snapshot

3.5.1 LOCSME-CCG algorithm

In order to introduce CG-based adaptive algorithms, we specifically divide them into two different algorithms, namely, LOCSME-CCG and its modified version LOCSME-MCG. In the approach of LOCSME-CCG, the SCV $\hat{\mathbf{l}}(i)$ is replaced by an estimate with a forgetting factor λ, which is a constant scalar less than and close to 1 as

$$\hat{\mathbf{l}}(i) = \lambda \hat{\mathbf{l}}(i-1) + \mathbf{x}(i) y^*(i), \tag{3.38}$$

before we employ it into the vector shrinkage method. The INC matrix is also estimated directly with this forgetting factor as

$$\hat{\mathbf{R}}(i) = \lambda \hat{\mathbf{R}}(i-1) + \mathbf{x}(i) \mathbf{x}^H(i). \tag{3.39}$$

In order to derive CG-based recursions we need to reformulate the cost function that needs to be minimized in [24] as follows

$$\underset{\hat{\mathbf{a}}_1(i), \mathbf{v}(i)}{\text{minimize}} \, \mathcal{J} = \mathbf{v}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{v}(i) - \mathcal{R}\{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)\}, \tag{3.40}$$

where $\mathbf{v}(i)$ is the CG-based weight vector. In LOCSME-CCG, we require a run of N iterations in each snapshot. In the nth iteration, $\hat{\mathbf{a}}_{1,n}(i)$ and $\mathbf{v}_n(i)$ are updated as follows

$$\hat{\mathbf{a}}_{1,n}(i) = \hat{\mathbf{a}}_{1,n-1}(i) + \alpha_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \tag{3.41}$$

$$\mathbf{v}_n(i) = \mathbf{v}_{n-1}(i) + \alpha_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i), \tag{3.42}$$

where $\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)$ and $\mathbf{p}_{\mathbf{v},n}(i)$ are direction vectors updated by

$$\mathbf{p}_{\hat{\mathbf{a}}_1,n+1}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) + \beta_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \tag{3.43}$$

$$\mathbf{p}_{\mathbf{v},n+1}(i) = \mathbf{g}_{\mathbf{v},n}(i) + \beta_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i), \tag{3.44}$$

where $\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)$ and $\mathbf{g}_{\mathbf{v},n}(i)$ are the negative gradients of the cost function in terms of $\hat{\mathbf{a}}_1(i)$ and $\mathbf{v}(i)$, respectively, which are expressed as

$$\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) = -\frac{\partial \mathcal{J}}{\partial \hat{\mathbf{a}}_{1,n}(i)} = \hat{\sigma}_1^2(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\hat{\mathbf{a}}_{1,n}(i) + \mathbf{v}_n(i), \tag{3.45}$$

$$\mathbf{g}_{\mathbf{v},n}(i) = -\frac{\partial \mathcal{J}}{\partial \mathbf{v}_n(i)} = \mathbf{g}_{\mathbf{v},n-1}(i) - \alpha_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\mathbf{x}(i)\mathbf{x}^H(i))\mathbf{p}_{\mathbf{v},n}(i). \tag{3.46}$$

The scaling parameters $\alpha_{\hat{\mathbf{a}}_1,n}(i)$, $\alpha_{\mathbf{v},n}(i)$ can be obtained by substituting (3.41) and (3.42) into (3.40) and minimizing with respect to $\alpha_{\hat{\mathbf{a}}_1,n}(i)$ and $\alpha_{\mathbf{v},n}(i)$, respectively. The solutions are given by

$$\alpha_{\hat{\mathbf{a}}_1,n}(i) = -\frac{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}{\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}, \tag{3.47}$$

$$\alpha_{\mathbf{v},n}(i) = \frac{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{p}_{\mathbf{v},n}(i)}{\mathbf{p}_{\mathbf{v},n}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_{1,n}(i)\hat{\mathbf{a}}_{1,n}^H(i))\mathbf{p}_{\mathbf{v},n}(i)}. \tag{3.48}$$

The parameters $\beta_{\hat{\mathbf{a}}_1,n}(i)$ and $\beta_{\mathbf{v},n}(i)$ should be chosen to provide conjugacy for direction vectors [21, 24] which results in

$$\beta_{\hat{\mathbf{a}}_1,n}(i) = \frac{\mathbf{g}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}(i)}, \tag{3.49}$$

$$\beta_{\mathbf{v},n}(i) = \frac{\mathbf{g}_{\mathbf{v},n}^H(i)\mathbf{g}_{\mathbf{v},n}(i)}{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{g}_{\mathbf{v},n-1}(i)}. \tag{3.50}$$

After $\hat{\mathbf{a}}_{1,n}(i)$ and $\mathbf{v}_n(i)$ are updated for N iterations, the beamforming weight vector $\mathbf{w}(i)$ can be computed by

$$\mathbf{w}(i) = \frac{\mathbf{v}_N(i)}{\hat{\mathbf{a}}_{1,N}^H(i)\mathbf{v}_N(i)}, \tag{3.51}$$

while the estimated steering vector is also updated to $\hat{\mathbf{a}}_{1,N}(i)$. Table 3.3 summarizes the LOCSME-CCG algorithm.

3.5.2 LOCSME-MCG algorithm

In LOCSME-MCG, we let only one iteration be performed per snapshot [21, 24], which further reduces the complexity compared to LOCSME-CCG. Here we denote the CG-based weights and steering vector updated by snapshots rather than inner iterations as

$$\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i-1) + \alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i), \tag{3.52}$$

$$\mathbf{v}(i) = \mathbf{v}(i-1) + \alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i). \tag{3.53}$$

As can be seen, the subscripts of all the quantities for inner iterations are eliminated. Then, we employ the degenerated scheme to ensure $\alpha_{\hat{\mathbf{a}}_1}(i)$ and $\alpha_{\mathbf{v}}(i)$ satisfy the convergence bound [24] given by

$$0 \leq \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i) \leq 0.5\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1), \tag{3.54}$$

Table 3.3: Proposed LOCSME-CCG Algorithm

Initialize:

$$\mathbf{C} = \int_{\theta_1 - \theta_e}^{\theta_1 + \theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta, \; [\mathbf{c}_1, \cdots, \mathbf{c}_p]: \text{p principal eigenvectors of } \mathbf{C}$$

$\mathbf{P} = [\mathbf{c}_1, \cdots, \mathbf{c}_p][\mathbf{c}_1, \cdots, \mathbf{c}_p]^H$

$\hat{\mathbf{l}}(0) = \mathbf{0}; \; \hat{\mathbf{R}}(0) = \mathbf{I}; \; \mathbf{w}(1) = \mathbf{v}_0(1) = \mathbf{1}; \; \hat{\rho}(1) = \rho(0) = 1; \; \lambda = 0.98;$

For each snapshot index $i = 1, 2, \cdots$:

$\hat{\mathbf{l}}(i) = \lambda\hat{\mathbf{l}}(i-1) + \mathbf{x}(i)y^*(i)$

$\hat{\mathbf{R}}(i) = \lambda\hat{\mathbf{R}}(i-1) + \mathbf{x}(i)\mathbf{x}^H(i)$

Steering vector mismatch estimation

$\hat{\nu}(i) = \sum \hat{\mathbf{l}}(i)/M$

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

$\hat{\rho}(i) = \dfrac{(1-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + \sum \hat{\mathbf{d}}(i-1)\sum^* \hat{\mathbf{d}}(i-1)}{(i-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + (1-\frac{i}{M})\sum \hat{\mathbf{d}}(i-1)\sum^* \hat{\mathbf{d}}(i-1)}$

$\hat{\mathbf{a}}_1(i) = \dfrac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}$

Desired signal power estimation

$\hat{\sigma}_1^2(i) = \dfrac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$

CCG-based estimations of steering vector mismatch and beamformer weights

$\hat{\mathbf{a}}_{1,0}(i) = \hat{\mathbf{a}}_1(i)$

$\mathbf{g}_{\hat{\mathbf{a}}_1,0}(i) = \hat{\sigma}_1^2(i)\mathbf{v}_0(i)\mathbf{v}_0^H(i)\hat{\mathbf{a}}_{1,0}(i) + \mathbf{v}_0(i)$

$\mathbf{g}_{\mathbf{v},0}(i) = \hat{\mathbf{a}}_{1,0}(i) - \hat{\mathbf{R}}(i)\mathbf{v}_0(i)$

$\mathbf{p}_{\hat{\mathbf{a}}_1,0}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,0}(i); \; \mathbf{p}_{\mathbf{v},0}(i) = \mathbf{g}_{\mathbf{v},0}(i)$

For each iteration index $n = 1, 2, \cdots, N$:

$\alpha_{\hat{\mathbf{a}}_1,n}(i) = -\dfrac{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}{\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}, \quad \alpha_{\mathbf{v},n}(i) = \dfrac{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{p}_{\mathbf{v},n}(i)}{\mathbf{p}_{\mathbf{v},n}^H(i)(\hat{\mathbf{R}}(i)-\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_{1,n}(i)\hat{\mathbf{a}}_{1,n}^H(i))\mathbf{p}_{\mathbf{v},n}(i)}$

$\hat{\mathbf{a}}_{1,n}(i) = \hat{\mathbf{a}}_{1,n-1}(i) + \alpha_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \quad \mathbf{v}_n(i) = \mathbf{v}_{n-1}(i) + \alpha_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i)$

$\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) = \hat{\sigma}_1^2(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\hat{\mathbf{a}}_{1,n}(i) + \mathbf{v}_n(i)$

$\mathbf{g}_{\mathbf{v},n}(i) = \mathbf{g}_{\mathbf{v},n-1}(i) - \alpha_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\mathbf{x}(i)\mathbf{x}^H(i))\mathbf{p}_{\mathbf{v},n}(i)$

$\beta_{\hat{\mathbf{a}}_1,n}(i) = \dfrac{\mathbf{g}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}(i)}, \quad \beta_{\mathbf{v},n}(i) = \dfrac{\mathbf{g}_{\mathbf{v},n}^H(i)\mathbf{g}_{\mathbf{v},n}(i)}{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{g}_{\mathbf{v},n-1}(i)}$

$\mathbf{p}_{\hat{\mathbf{a}}_1,n+1}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) + \beta_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \quad \mathbf{p}_{\mathbf{v},n+1}(i) = \mathbf{g}_{\mathbf{v},n}(i) + \beta_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i)$

End iteration

computation of beamformer weights

$\mathbf{v}_0(i+1) = \mathbf{v}_N(i)$

$\mathbf{w}(i) = \dfrac{\mathbf{v}_N(i)}{\hat{\mathbf{a}}_{1,N}^H(i)\mathbf{v}_N(i)}$

End snapshot

$$0 \le \mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i) \le 0.5\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1). \tag{3.55}$$

Instead of updating the negative gradient vectors $\mathbf{g}_{\hat{\mathbf{a}}_1}(i)$ and $\mathbf{g}_{\mathbf{v}}(i)$ in iterations, now we utilize the forgetting factor to re-express them in one snapshot as

$$\mathbf{g}_{\hat{\mathbf{a}}_1}(i) = (1-\lambda)\mathbf{v}(i) + \lambda\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1) + \hat{\sigma}_1^2(i)\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i), \tag{3.56}$$

$$\mathbf{g}_{\mathbf{v}}(i) = (1-\lambda)\hat{\mathbf{a}}_1(i) + \lambda\mathbf{g}_{\mathbf{v}}(i-1) - \alpha_{\mathbf{v}}(i)(\hat{\mathbf{R}}(i)$$
$$- \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{v}(i-1). \tag{3.57}$$

Pre-multiplying (3.56) and (3.57) by $\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)$ and $\mathbf{p}_{\mathbf{v}}^H(i)$, respectively, and taking expectations we obtain

$$E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i)] = E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)(\mathbf{v}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1)(i)] + \lambda E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]$$
$$- \lambda E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)] + E[\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\hat{\sigma}_1^2(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)], \tag{3.58}$$

$$E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i)] = \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]$$
$$- E[\alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)], \tag{3.59}$$

where in (3.59) we have $E[\hat{\mathbf{R}}(i)\mathbf{v}(i-1)] = E[\hat{\mathbf{a}}_1(i)]$. After substituting (3.59) back into (3.55) we obtain the bounds for $\alpha_{\mathbf{v}}(i)$ as follows

$$\frac{(\lambda - 0.5)E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]}{E[\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)]} \le E[\alpha_{\mathbf{v}}(i)]$$
$$\le \frac{\lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]}{E[\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)]}. \tag{3.60}$$

Then we can introduce a constant parameter $\eta_{\mathbf{v}} \in [0, 0.5]$ to restrict $\alpha_{\mathbf{v}}(i)$ within the bounds in (3.60) as

$$\alpha_{\mathbf{v}}(i) = \frac{\lambda(\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1) - \mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)) - \eta_{\mathbf{v}}\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)}{\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)}. \tag{3.61}$$

Similarly, we can also obtain the bounds for $\alpha_{\hat{\mathbf{a}}_1}(i)$. For simplicity let us define $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)] = A$, $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)] = B$, $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)] = C$ and $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\hat{\sigma}_1^2(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)] = D$. Substituting equation (3.58) into (3.54) gives

$$\frac{\lambda(B-A) - B + C}{D} \le E[\alpha_{\hat{\mathbf{a}}_1}(i)] \le \frac{\lambda(B-A) - B + C + 0.5A}{D}, \tag{3.62}$$

in which we can introduce another constant parameter $\eta_{\hat{\mathbf{a}}_1} \in [0, 0.5]$ to restrict $\alpha_{\hat{\mathbf{a}}_1}(i)$ within the bounds in (3.62) as

$$E[\alpha_{\hat{\mathbf{a}}_1}(i)] = \frac{\lambda(B - A) - B + C + \eta_{\hat{\mathbf{a}}_1} A}{D}, \tag{3.63}$$

or

$$\alpha_{\hat{\mathbf{a}}_1}(i) = [\lambda(\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i - 1)) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)$$
$$+ \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i) + \eta_{\hat{\mathbf{a}}_1}\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i - 1)]/[\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)]. \tag{3.64}$$

Then we can update the direction vectors $\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$ and $\mathbf{p}_{\mathbf{v}}(i)$ by

$$\mathbf{p}_{\hat{\mathbf{a}}_1}(i + 1) = \mathbf{g}_{\hat{\mathbf{a}}_1}(i) + \beta_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i), \tag{3.65}$$

$$\mathbf{p}_{\mathbf{v}}(i + 1) = \mathbf{g}_{\mathbf{v}}(i) + \beta_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i), \tag{3.66}$$

where $\beta_{\hat{\mathbf{a}}_1}(i)$ and $\beta_{\mathbf{v}}(i)$ are updated by

$$\beta_{\hat{\mathbf{a}}_1}(i) = \frac{[\mathbf{g}_{\hat{\mathbf{a}}_1}(i) - \mathbf{g}_{\hat{\mathbf{a}}_1}(i - 1)]^H \mathbf{g}_{\hat{\mathbf{a}}_1}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1}^H(i - 1)\mathbf{g}_{\hat{\mathbf{a}}_1}(i - 1)}, \tag{3.67}$$

$$\beta_{\mathbf{v}}(i) = \frac{[\mathbf{g}_{\mathbf{v}}(i) - \mathbf{g}_{\mathbf{v}}(i - 1)]^H \mathbf{g}_{\mathbf{v}}(i)}{\mathbf{g}_{\mathbf{v}}^H(i - 1)\mathbf{g}_{\mathbf{v}}(i - 1)}. \tag{3.68}$$

Finally we can update the beamforming weights by

$$\mathbf{w}(i) = \frac{\mathbf{v}(i)}{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)}, \tag{3.69}$$

The LOCSME-MCG algorithm is summarized in Table 3.4. The MCG approach employs the forgetting factor λ and constant η for estimating $\alpha(i)$, which means its performance may depend on a suitable choice of these parameters. However, it requires much lower complexity for the elimination of inner recursions compared to CCG and presents a similar performance in the simulations.

3.6 Performance Analysis

This section investigates the effects of shrinkage approaches and the computational complexity of the proposed algorithms. Firstly we rewrite the vector shrinkage recursions

Table 3.4: Proposed LOCSME-MCG Algorithm

Initialize:

$\mathbf{C} = \int_{\theta_1-\theta_e}^{\theta_1+\theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta$, $[\mathbf{c}_1,\cdots,\mathbf{c}_p]$: p principal eigenvectors of $\mathbf{C}$

$\mathbf{P} = [\mathbf{c}_1,\cdots,\mathbf{c}_p][\mathbf{c}_1,\cdots,\mathbf{c}_p]^H$

$\hat{\mathbf{l}}(0) = \mathbf{0}; \hat{\mathbf{R}}(0) = \mathbf{I}; \mathbf{w}(1) = \mathbf{v}(0) = \mathbf{1}; \hat{\rho}(1) = \rho(0) = 1;$

$\lambda = 0.95; \eta_\mathbf{v} = \eta_{\hat{\mathbf{a}}_1} = 0.1;$

$\mathbf{g}_\mathbf{v}(0) = \mathbf{p}_\mathbf{v}(1) = \hat{\mathbf{R}}(0)\mathbf{v}(1); \mathbf{g}_{\hat{\mathbf{a}}_1}(0) = \mathbf{p}_{\hat{\mathbf{a}}_1}(1) = \mathbf{v}(0);$

For each snapshot index $i = 1, 2, \cdots$:

$\hat{\mathbf{l}}(i) = \lambda\hat{\mathbf{l}}(i-1) + \mathbf{x}(i)y^*(i)$

$\hat{\mathbf{R}}(i) = \lambda\hat{\mathbf{R}}(i-1) + \mathbf{x}(i)\mathbf{x}^H(i)$

Steering vector mismatch estimation

$\hat{\nu}(i) = \sum \hat{\mathbf{l}}(i)/M$

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

$\hat{\rho}(i) = \frac{(1-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1)+\sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}{(i-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1)+(1-\frac{i}{M})\sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}$

$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}$

Desired signal power estimation

$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2-|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$

MCG-based estimations of steering vector mismatch and beamformer weights

$\alpha_{\hat{\mathbf{a}}_1}(i) = [\lambda(\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) + \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)$

$+\eta_{\hat{\mathbf{a}}_1}\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]/[\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)]$

$\alpha_\mathbf{v}(i) = \frac{\lambda(\mathbf{p}_\mathbf{v}^H(i)\mathbf{g}_\mathbf{v}(i-1)-\mathbf{p}_\mathbf{v}^H(i)\hat{\mathbf{a}}_1(i))-\eta_\mathbf{v}\mathbf{p}_\mathbf{v}^H(i)\mathbf{g}_\mathbf{v}(i-1)}{\mathbf{p}_\mathbf{v}^H(i)(\hat{\mathbf{R}}(i)-\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_\mathbf{v}(i)}$

$\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i-1) + \alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$

$\mathbf{v}(i) = \mathbf{v}(i-1) + \alpha_\mathbf{v}(i)\mathbf{p}_\mathbf{v}(i)$

$\mathbf{g}_{\hat{\mathbf{a}}_1}(i) = (1-\lambda)\mathbf{v}(i) + \lambda\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1) + \hat{\sigma}_1^2(i)\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)$

$\mathbf{g}_\mathbf{v}(i) = (1-\lambda)\hat{\mathbf{a}}_1(i) + \lambda\mathbf{g}_\mathbf{v}(i-1) - \alpha_\mathbf{v}(i)(\hat{\mathbf{R}}(i)-\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_\mathbf{v}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{v}(i-1)$

$\beta_{\hat{\mathbf{a}}_1}(i) = \frac{[\mathbf{g}_{\hat{\mathbf{a}}_1}(i)-\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]^H\mathbf{g}_{\hat{\mathbf{a}}_1}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1}^H(i-1)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)}$

$\beta_\mathbf{v}(i) = \frac{[\mathbf{g}_\mathbf{v}(i)-\mathbf{g}_\mathbf{v}(i-1)]^H\mathbf{g}_\mathbf{v}(i)}{\mathbf{g}_\mathbf{v}^H(i-1)\mathbf{g}_\mathbf{v}(i-1)}$

$\mathbf{p}_{\hat{\mathbf{a}}_1}(i+1) = \mathbf{g}_{\hat{\mathbf{a}}_1}(i) + \beta_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$

$\mathbf{p}_\mathbf{v}(i+1) = \mathbf{g}_\mathbf{v}(i) + \beta_\mathbf{v}(i)\mathbf{p}_\mathbf{v}(i)$

Computation of beamformer weights

$\mathbf{w}(i) = \frac{\mathbf{v}(i)}{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)}$

End snapshot

into a matrix shrinkage recursion. Then we employ an eigen-decomposition approach to examine the eigenvalues dispersion for the vector shrinkage and matrix shrinkage cases by exploring the MSE of their eigenvalues [30], and give reasons why shrinkage gives an important contribution to the performance. Then we present a complexity analysis for the proposed algorithms and comparisons to the existing RAB algorithms. It is clear that the proposed algorithms achieve one degree lower complexity than most of the existing ones.

3.6.1 Shrinkage Analysis

First of all, we modify the vector shrinkage formula (3.12) to the following full rank matrix form

$$\hat{\mathbf{D}}(i) = \hat{\rho}(i)\hat{\mathbf{V}}(i) + (1 - \hat{\rho}(i))\hat{\mathbf{L}}(i), \tag{3.70}$$

where $\hat{\mathbf{V}}(i)$, $\hat{\mathbf{D}}(i)$ and $\hat{\mathbf{L}}(i)$ are all diagonal matrix, having each of their diagonal entries identical to $\hat{\nu}(i)$, elements of the optimal shrinkage estimator $\hat{\mathbf{d}}(i)$ and elements of the SCV $\hat{\mathbf{l}}(i)$, respectively, whereas all the three matrices have their other entries equal to zero. Associated with (3.18), it can be seen they share the same linear shrinkage formula. Now, we carry out eigenvalue decompositions for every matrix in (3.12). Since the eigenvalues of a diagonal matrix are simply its diagonal entries, the eigenvalues of $\hat{\mathbf{D}}(i)$, $\hat{\mathbf{V}}(i)$ and $\hat{\mathbf{L}}(i)$ can be expressed as

$$\{\hat{d}_1(i), \cdots, \hat{d}_M(i)\}, \tag{3.71}$$

$$\{\hat{\nu}(i), \cdots, \hat{\nu}(i)\}, \tag{3.72}$$

$$\{\hat{l}_1(i), \cdots, \hat{l}_M(i)\}, \tag{3.73}$$

respectively. Since in each iteration, $\hat{\mathbf{D}}(i-1)$ and $\hat{\mathbf{V}}(i)$ are estimated (known) qualities and $\hat{\mathbf{L}}(i)$ is the quantity to be estimated, we have

$$E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{V}}(i)\|^2] = E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1) + \hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\|^2]$$
$$= E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2] + E[\|\hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\|^2] + 2E[\langle\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1), \hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\rangle]$$
$$= E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2] + \|\hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\|^2 + 2\langle E[\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)], \hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\rangle,$$
$$\tag{3.74}$$

where $\langle,\rangle$ denotes the inner product (i.e., element-wise products between two matrices in this case, or known as Hadamard product) and we have $E[\hat{\mathbf{L}}(i)] = \hat{\mathbf{D}}(i-1)$, then the

inner product term in the above equation equals 0, which yields the following

$$E[\|\hat{\mathbf{L}}(i) - \hat{\nu}(i)\mathbf{I}\|^2] - \|\hat{\mathbf{D}}(i-1) - \hat{\nu}(i)\mathbf{I}\|^2 = E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2]. \quad (3.75)$$

Equation (3.75) can be interpreted in terms of the eigenvalues of the matrices if we rewrite it as

$$E\Big[\frac{1}{M}\sum_{m=1}^{M}(\hat{l}_m(i)-\hat{\nu}(i))^2\Big] - \frac{1}{M}\sum_{m=1}^{M}(\hat{d}_m(i-1)-\hat{\nu}(i))^2 = E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2]. \quad (3.76)$$

Note that in (3.76), $\hat{\nu}(i)$ actually represents the mean value of the SCV $\hat{\mathbf{l}}(i)$ or the diagonal entries of matrix $\hat{\mathbf{V}}(i)$. Similarly to the matrix shrinkage in (3.18), we can carry on the same analysis even though the matrices are no longer diagonal but will lead to a more general result. Assuming the eigenvalues of the matrices $\tilde{\mathbf{R}}(i)$, $\hat{\mathbf{F}}_0(i)$, and $\hat{\mathbf{R}}(i)$ are

$$\{\lambda_1(i), \cdots, \lambda_M(i)\}, \quad (3.77)$$

$$\{f_1(i), \cdots, f_M(i)\}, \quad (3.78)$$

$$\{\gamma_1(i), \cdots, \gamma_M(i)\}, \quad (3.79)$$

respectively. Then we have

$$E[\|\hat{\mathbf{R}}(i) - \hat{\mathbf{F}}_0(i)\|^2] = E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1) + \tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2]$$
$$= E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2] + E[\|\tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2] + 2E[\langle\hat{\mathbf{R}}(i)-\tilde{\mathbf{R}}(i-1), \tilde{\mathbf{R}}(i-1)-\hat{\mathbf{F}}_0(i)\rangle]$$
$$= E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2] + \|\tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2 + 2\langle E[\hat{\mathbf{R}}(i)-\tilde{\mathbf{R}}(i-1)], \tilde{\mathbf{R}}(i-1)-\hat{\mathbf{F}}_0(i)\rangle, \quad (3.80)$$

where the inner product term equals 0 because of $E[\hat{\mathbf{R}}(i)] = \tilde{\mathbf{R}}(i-1)$, which results in

$$E[\|\hat{\mathbf{R}}(i) - \hat{\mathbf{F}}_0(i)\|^2] - \|\tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2 = E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2]. \quad (3.81)$$

Noting that $\hat{\mathbf{F}}_0(i) = \hat{\nu}_0(i)\mathbf{I}$, then (3.81) is equivalent to

$$E[\|\hat{\mathbf{R}}(i) - \hat{\nu}_0(i)\mathbf{I}\|^2] - \|\tilde{\mathbf{R}}(i-1) - \hat{\nu}_0(i)\mathbf{I}\|^2 = E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2], \quad (3.82)$$

which can be rewritten in an alternative form as

$$E[\frac{1}{M}\sum_{m=1}^{M}(\gamma_m(i) - \hat{\nu}_0(i))^2] - \frac{1}{M}\sum_{m=1}^{M}(\lambda_m(i-1) - \hat{\nu}_0(i))^2 = E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2]. \quad (3.83)$$

Because the expectation on the right hand side of equation (3.76) and (3.83) are always non-negative, so we have their left hand side always equal or larger than 0, which yields

$$E\Big[\frac{1}{M}\sum_{m=1}^{M}(\hat{l}_m(i) - \hat{\nu}(i))^2\Big] \geq \frac{1}{M}\sum_{m=1}^{M}(\hat{d}_m(i-1) - \hat{\nu}(i))^2, \qquad (3.84)$$

$$E\Big[\frac{1}{M}\sum_{m=1}^{M}(\gamma_m(i) - \hat{\nu}_0(i))^2\Big] \geq \frac{1}{M}\sum_{m=1}^{M}(\lambda_m(i-1) - \hat{\nu}_0(i))^2. \qquad (3.85)$$

Since we also know that

$$E[\hat{\nu}(i)] = \frac{1}{M}\sum_{m=1}^{M}\hat{d}_m(i-1), \qquad (3.86)$$

$$E[\hat{\nu}_0(i)] = \frac{1}{M}\sum_{m=1}^{M}\lambda_m(i-1), \qquad (3.87)$$

which express the expected mean of the eigenvalues of the sampled matrix $\hat{\mathbf{L}}(i)$ and $\hat{\mathbf{R}}(i)$ in snapshot i, respectively. Then equations (3.84) and (3.85) indicate that the expected MSE of the eigenvalues of $\hat{\mathbf{L}}(i)$ or $\hat{\mathbf{R}}(i)$ in snapshot i is always larger or equal to those of the optimal shrinkage estimator $\hat{\mathbf{D}}(i-1)$ or $\tilde{\mathbf{R}}(i-1)$ obtained from the previous snapshot. In other words, the eigenvalues of the sampled matrix are more dispersedly distributed (here we should have $\hat{d}_1(i-1) > \hat{l}_1(i) > 0$, $\hat{d}_m(i-1) < \hat{l}_m(i)$ and $\lambda_1(i-1) > \gamma_1(i) > 0$, $\lambda_m(i-1) < \gamma_m(i)$) based on their expected mean value than those of the optimal shrinkage estimator from the last snapshot. Shrinking the sampled matrix to a matrix with less dispersed eigenvalues can lead to an improved covariance matrix estimator as reported in [17].

3.6.2 Complexity Analysis

In this part, we analyze the computational complexity in terms of flops (total number of additions and multiplications) required by the proposed RAB algorithms. The proposed RAB algorithms avoid costly matrix inversion and multiplication procedures, which are unavoidable in the existing RAB algorithms. The complexity comparison among different algorithms are listed in Table 3.5. It should be noted that LOCSME-CCG has its complexity dependent on the number of inner iterations N, which can be properly selected within the range of $5 - 10$. However, the low-complexity worst-case (LCWC) algorithm of [19] also requires N inner iterations per snapshots, which significantly varies in different snapshots and is usually much larger than the value of N in the proposed LOCSME-CCG

Table 3.5: Complexity Comparison

RAB Algorithms	Flops/snapshot
LOCSME [31]	$4M^3 + 3M^2 + 20M$
RCB [8]	$2M^3 + 11M^2$
Algorithm of [12]	$M^{3.5} + 7M^3 + 5M^2 + 3M$
LOCME [13]	$2M^3 + 4M^2 + 5M$
LCWC [19]	$N(2M^2 + 7M)$
LOCSME-SG	$15M^2 + 30M$
LOCSME-CCG	$(5 + 8N)M^2 + (21 + 32N)M$
LOCSME-MCG	$13M^2 + 77M$

algorithm. It is clear that our proposed algorithms have one degree lower complexity in terms of the number of sensors M, which are dominated by $\mathcal{O}(M^2)$, resulting in great advantages when M is large. Fig. 3.1 gives illustrations of the complexity comparison of the listed algorithms, where the values of N for [19] and the proposed LOCSME-CCG are selected as 50 and 10, respectively.
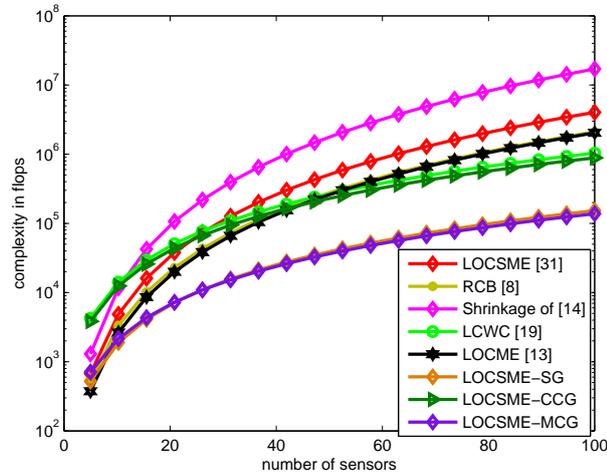
Figure 3.1: Complexity versus number of sensors

3.7   Simulations

The simulations are carried out under both coherent and incoherent local scattering mismatch [9] 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 observed. The desired signal is assumed to arrive at $\theta_1 = 10°$ while there are other two interferers impinging on the antenna array from directions $\theta_2 = 30°$ and $\theta_3 = 50°$. The signal-to-interference ratio (SIR) is fixed at 0dB. For the curves with the optimum beamforming in each of the comparisons, we employ the MVDR beamformer and assume that the DoA of the desired signal is perfectly known (without mismatch) and that the covariance matrix of the received data is also perfectly known perfectly so that the output SINR can be directly computed with (3.3). For our proposed algorithms, the angular sector in which the desired signal is assumed to be located is chosen as $[\theta_1 - 5°, \theta_1 + 5°]$ and the number of eigenvectors of the subspace projection matrix p is selected manually with the help of simulations. The results focus on the beamformer output SINR performance versus the number of snapshots, or a variation of input SNR (-10dB to 30dB).

3.7.1   Mismatch due to Coherent Local Scattering

If we choose the number of scatters as 4, then the steering vector of the desired signal affected by a time-invariant coherent local scattering effect is modeled as

$$\mathbf{a}_1 = \mathbf{p} + \sum_{k=1}^{4} e^{j\varphi_k} \mathbf{b}(\theta_k), \tag{3.88}$$

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 $10°$ and standard deviation $2°$. The angles $\varphi_k(k = 1, 2, 3, 4)$ are independently and uniformly taken from the interval $[0, 2\pi]$ in each simulation run. Notice that θ_k and φ_k change from trials while remaining constant over snapshots.

Fig. 3.2 and Fig. 3.3 illustrate the performance comparisons of SINR versus snapshots

and SINR versus SNR, respectively, in terms of the mentioned RAB algorithms in the last section under coherent scattering case. Specifically to obtain Fig. 3.2, we assume the noise power is known and select $\mu = 0.2$, $\mu_\epsilon = 1$, $\sigma_\epsilon = 0.001$, $\lambda_q = 0.99$, $\mathbf{R}_0 = 10\mathbf{I}$ for LOCSME-SG, $\lambda = 0.95$ for LOCSME-CCG and $\lambda = 0.95$, $\eta = 0.2$ for LOCSME-MCG. However, selection of these parameters may vary according to different input SNR as in Fig. 3.3. The proposed algorithms outperform the other algorithms and are very close to the standard LOCSME, especially for LOCSME-CCG and LOCSME-MCG.
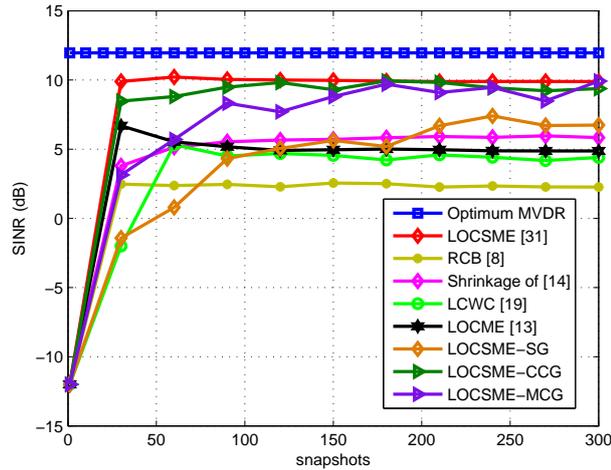
Figure 3.2: Coherent local scattering, SINR versus snapshots
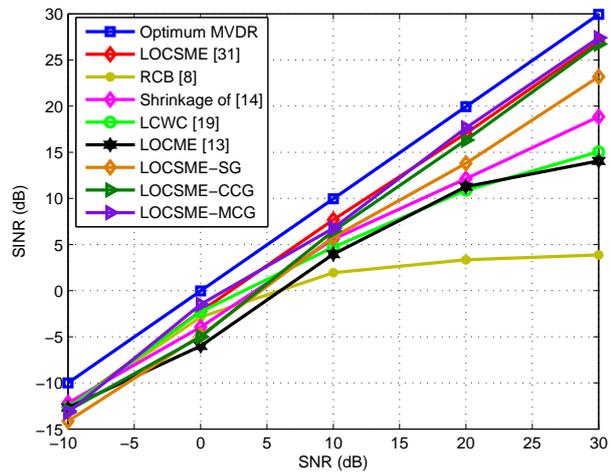
Figure 3.3: Coherent local scattering, SINR versus SNR

In Fig. 3.4, we use an maximum likelihood (ML)-based method to estimate the noise power in LOCSME, LOCSME-SG, LOCSME-CCG and LOCSME-MCG in the same

scenario of Fig. 3.2. It is clear that no noticeable differences between their performance can be observed by comparing Fig. 3.2 and Fig. 3.4.
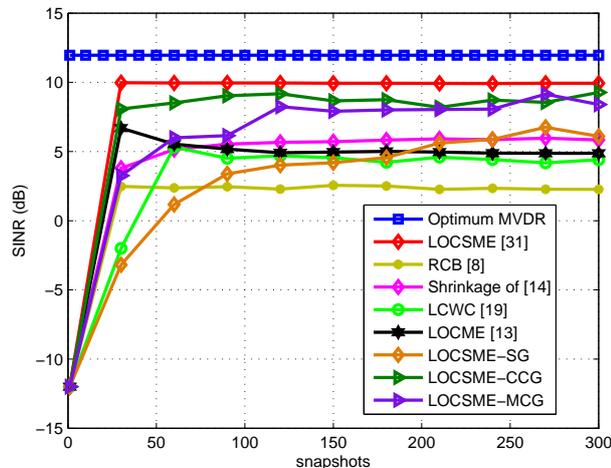
Figure 3.4: coherent local scattering, SINR versus snapshots

3.7.2 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

$$\mathbf{a}_1(i) = s_0(i)\mathbf{p} + \sum_{k=1}^{4} s_k(i)\mathbf{b}(\theta_k), \tag{3.89}$$

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 $10°$ and standard deviation $2°$. This time, $s_k(i)$ changes both from run to run and from snapshot to snapshot.

Fig. 3.5 and Fig. 3.6 illustrate the performance comparisons of SINR versus snapshots and SINR versus SNR, respectively, in terms of the mentioned RAB algorithms in the last section under incoherent scattering case. To obtain Fig. 3.5, we select $\mu = 0.1$, $\mu_\epsilon = 5$, $\sigma_\epsilon = 0.001$, $\lambda_q = 0.99$, $\mathbf{R}_0 = 50\mathbf{I}$ for LOCSME-SG, $\lambda = 0.99$ for LOCSME-CCG and $\lambda = 0.95$, $\eta = 0.3$ for LOCSME-MCG. However, we have optimized the parameters to give the best possible performance at different input SNRs.
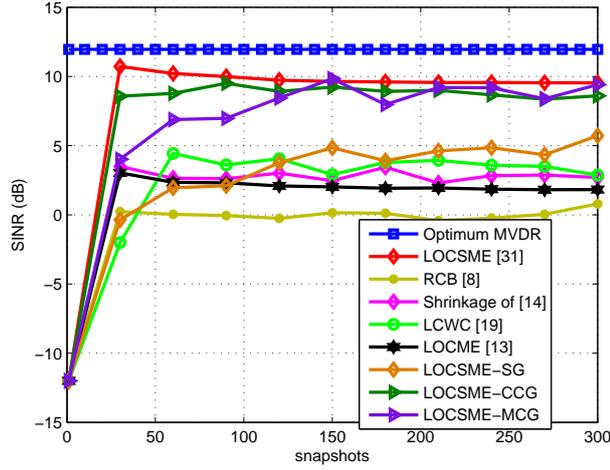
Figure 3.5: incoherent local scattering, SINR versus snapshots
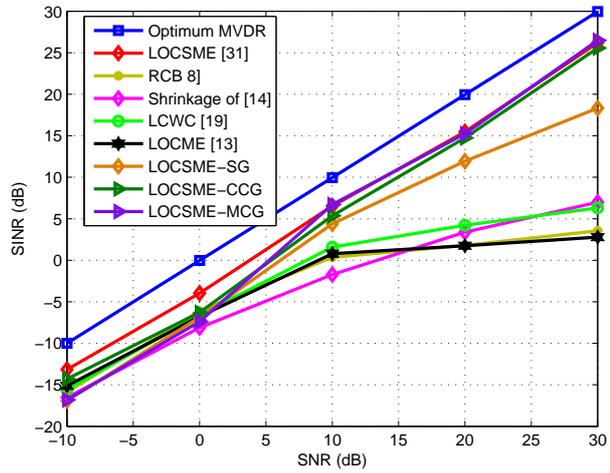
Figure 3.6: incoherent local scattering, SINR versus SNR

Differently 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, the proposed algorithms are still able to outperform 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 [15] that the desired signal always presents in any kind of diagonal loading technique. However, the proposed algorithms have im-

Table 3.6: Changes of Interferers

Snapshots	DoAs
$0 - 150$	$\theta_1 = 10°, \theta_2 = 30°, \theta_3 = 50°.$
$150 - 300$	$\theta_1 = 15°, \theta_2 = 25°, \theta_3 = 35°.$

proved the estimation accuracy, so that the high SNR degradation is successfully avoided as can be seen in Fig. 3.5 and Fig. 3.6.

We assess the SINR performance versus the number of snapshots of the selected algorithms in a specific time-varying scenario with the desired signal operating at 12 dB. The scenario is characterized by a set of source signals which have associated DoAs from the beginning of their operation until 150 snapshots. The DoAs of these source signals suddenly change at 150 snapshots but remain the same powers, which requires the beamforming algorithms to adjust to the new environment as described in Table 3.6. The result of this scenario is shown in Fig. 3.7.
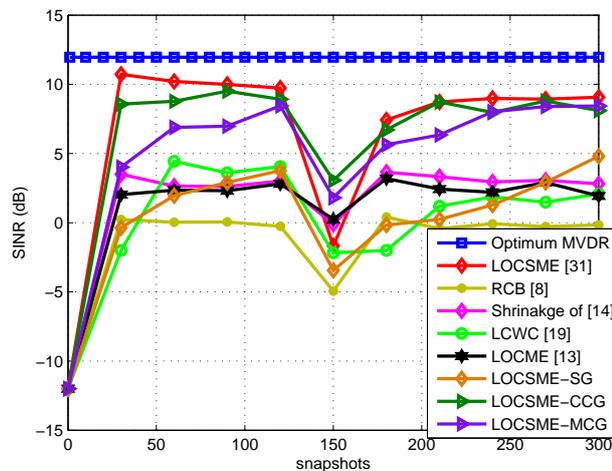
Figure 3.7: Scenario with incoherent local scattering and time-varying DoAs

In addition, it should also be emphasized that performance comparisons with the conventional adaptive algorithms (i.e. SG, CCG or MCG without combined to LOCSME) are not included, as they are not recognized as RAB algorithms and have much worse performance in the presence of uncertainties. Actually, as mentioned in the introduction, it has already been shown that conventional adaptive beamforming algorithms are

extremely sensitive to the statistical characteristics of the sampled data (i.e. data size and
data accuracy). Especially, when these algorithms suffer environment uncertainties (i.e.
steering vector mismatch), significant further performance degradation is unavoidable.

3.8   Summary

This chapter introduces low-complexity adaptive RAB algorithms developed from the
LOCSME RAB method. In each of these algorithms, we have derived recursions for the
weight vector update and exploited effective shrinkage methods, both of which require
low complexity without losing any noticeable performance. Additionally, in the CG-
based RAB algorithms we have enabled the estimation for the mismatch steering vector
inside the CG recursions to enhance the robustness. Both complexity and performance
comparisons are provided and analyzed. Simulation results have shown that the proposed
algorithms achieved excellent output SINR performance and are suitable for operation in
high input SNR.

Chapter 4

Orthogonal Krylov Subspace Projection Mismatch Estimation for Robust Adaptive Beamforming

Contents

4.1 Introduction

Some beamforming systems in advanced applications require antenna arrays with a very large number of sensor elements, which leads to the increase of computational complexity and the decrease of the convergence rate for computing the parameters of the beamformer.

In the case of large sensor arrays the above mentioned RAB methods may encounter problems for their application. This is because in these RAB algorithms, a cubic (i.e., $\mathcal{O}(M^3)$) or greater computational cost is required to compute the beamforming parameters. Therefore, dimensionality reduction (or rank-reduction) methods [47]- [60] have been developed to reduce the complexity and improve the convergence rate.

4.1.1 Prior and Related Work

In the recent years, great efforts have been devoted to the investigation of robust dimensionality reduction techniques for RAB. The beamspace approach of [48] projects the data onto a lower dimension subspace by using a beamspace matrix, whose columns are determined by linearly independent constrained optimization problems. A more effective approach (i.e., [41]- [45], [49], [50], [62]) is based on preprocessing the array observation data using a Krylov subspace. However, there are different ways to generate the Krylov subspace and the choice usually depends on the cost and the performance. The Arnoldi method [38, 39, 46] and the Lanczos iterations [38–40] are typical approaches used to generate orthogonal Krylov subspaces, whereas [44] also introduces a method to generate non-orthogonal ones. However, the main challenge in these techniques is the model order determination. Specifically, the model order must be properly chosen to ensure robustness to over-determination of the system model order [42]. Another effective approach to dimensionality reduction is the joint iterative optimization (JIO) techniques [51]- [56], [57]-[60], [85], [63], [64], which employ a subspace projection matrix and jointly and iteratively optimize the bases of the subspace and the beamformer weights. The work in [52] has developed a recursive least squares (RLS) adaptive algorithm based on widely-linear processing using the JIO technique. The study in [54] has devised efficient stochastic gradient (SG) and RLS RAB algorithms from a modified JIO (MJIO) scheme.

4.1.2 Contributions

In this chapter, we propose and study novel RAB algorithms that are based on low-rank and cross-correlation techniques. In the proposed techniques, we exploit the prior knowl-

edge that the steering vector mismatch of the desired signal is located within an angular sector which is assumed known. The proposed algorithms are based on the exploitation of the cross-correlation between the array observation data and the output of the beamformer, which avoids costly optimization procedures. We firstly construct a linear system (considered in high dimension) involving the mismatched steering vector and the statistics of the sampled data. Then we employ an iterative full orthogonalization method (FOM) [38, 39] to compute an orthogonal Krylov subspace whose model order is determined by both the minimum sufficient rank [42], which ensures no information loss when capturing the signal of interest (SoI) with interferers, and the execute-and-stop criterion of FOM [38, 39], which automatically avoids overestimating the number of bases of the computed subspace. The estimated vector that contains the cross-correlation between the array observation data and the beamformer output is projected onto the Krylov subspace, in order to update the steering vector mismatch, resulting in the proposed orthogonal Krylov subspace projection mismatch estimation (OKSPME) method.

Furthermore, based on the OKSPME method, we have also devised adaptive stochastic gradient (SG), conventional conjugate gradient (CCG) and modified conjugate gradient (MCG) algorithms derived from the proposed optimization problems to reduce the cost for computing the beamforming weights, resulting in the proposed OKSPME-SG, OKSPME-CCG and OKSPME-MCG RAB algorithms. We remark that the steering vector is also estimated and updated using the CG-based recursions to produce an even more precise estimate. Derivations of the proposed algorithms are presented and discussed along with an analysis of their computational complexity.

Moreover, we develop an analysis of the mean squared error (MSE) between the estimated and the actual steering vectors for the general approach of using a presumed angular sector associated with subspace projections. This analysis mathematically describes how precise the steering vector mismatch can be estimated. Upper and lower bounds are derived and compared with the approach in [12]. Another analysis on the computational complexity of the proposed and existing algorithms is also provided.

In the simulations, we consider local scattering scenarios (both coherent and incoherent) to model the mismatch effects. We also study the performance of the proposed algorithms by testing the output signal-to-interference-plus-noise ratio (SINR) of the beam-

former with respect to training snapshots and different input SNRs. The number of sensor elements and interferers is also varied and compared in each scenario to provide a comprehensive performance study. In summary, the contributions of this part of the work are:

- The proposed OKSPME RAB method.

- The development of the modified SG and CG type OKSPME RAB algorithms.

- An analysis of the computational complexity and the MSE performance of the proposed and existing RAB algorithms.

The remaining sections of this chapter are organized as follows: The system model and problem statement are described in Section 4.2. Section 4.3 introduces the proposed OKSPME method, whereas Section 4.4 introduces the proposed OKSPME, OKSPME-SG, OKSPME-CCG and OKSPME-MCG robust adaptive algorithms. Section 4.5 provides the MSE analysis of the steering vector estimation and the complexity analysis. Section 4.6 presents and discusses the simulation results. Section 4.7 gives the summary.

4.2   System Model and Problem Statement

Let us consider a linear antenna array of M sensors and K narrowband signals which impinge on the array. The data received at the ith snapshot can be modeled as

$$\mathbf{x}(i) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(i) + \mathbf{n}(i), \tag{4.1}$$

where $\mathbf{s}(i) \in \mathbb{C}^{K \times 1}$ are uncorrelated source signals, $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_K]^T \in \mathbb{R}^K$ is a vector containing the directions of arrival (DoAs) and $[.]^T$ denotes the transpose, $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1) + \mathbf{e}, \cdots, \mathbf{a}(\theta_K)] = [\mathbf{a}_1, \cdots, \mathbf{a}_K] \in \mathbb{C}^{M \times K}$ is the matrix which contains the steering vector for each DoA and $\mathbf{e}$ is the steering vector mismatch of the desired signal, $\mathbf{n}(i) \in \mathbb{C}^{M \times 1}$ is assumed to be complex Gaussian noise with zero mean and variance σ_n^2. The beamformer output is given by

$$y(i) = \mathbf{w}^H \mathbf{x}(i), \tag{4.2}$$

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

$$SINR = \frac{\sigma_1^2 |\mathbf{w}^H \mathbf{a}_1|^2}{\mathbf{w}^H \mathbf{R}_{I+N} \mathbf{w}}. \tag{4.3}$$

where σ_1^2 is the desired signal power, $\mathbf{R}_{I+N}$ is the interference-plus-noise covariance (INC) matrix. The problem of maximizing the SINR in (4.3) can be cast as the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathbf{w}^H \mathbf{R}_{I+N} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^H \mathbf{a}_1 = 1, \end{aligned} \tag{4.4}$$

which is known as the MVDR beamformer or Capon beamformer [1, 8]. The optimum weight vector is given by

$$\mathbf{w}_{opt} = \frac{\mathbf{R}_{I+N}^{-1} \mathbf{a}_1}{\mathbf{a}_1^H \mathbf{R}_{I+N}^{-1} \mathbf{a}_1}.$$

Since $\mathbf{R}_{I+N}$ is usually unknown in practice, it can be replaced by the sample covariance matrix (SCM) of the received data as

$$\hat{\mathbf{R}}(i) = \frac{1}{i} \sum_{k=1}^{i} \mathbf{x}(k) \mathbf{x}^H(k). \tag{4.5}$$

Using the SCM for directly computing the weights will lead to the sample matrix inversion (SMI) beamformer $\mathbf{w}_{SMI} = \frac{\hat{\mathbf{R}}^{-1} \mathbf{a}_1}{\mathbf{a}_1^H \hat{\mathbf{R}}^{-1} \mathbf{a}_1}$. However, the SMI beamformer requires a large number of snapshots to converge and is sensitive to steering vector mismatches [7, 11]. As previously mentioned, most of the conventional and existing RAB algorithms are computationally costly especially when encountering arrays with a very large number of sensors. Therefore, the RAB design problem we are interested in solving includes the following aspects:

- To design cost-efficient algorithms that are robust against values of SNRs and interferers in the presence of uncertainties in the steering vector of a desired signal.

- The proposed algorithms must preserve their robustness and low-complexity features for large sensor arrays.

4.3 Proposed OKSPME Method

In this section, the proposed OKSPME method is introduced. This method aims to construct a linear system involving only known or estimated statistics and then projects an estimated cross-correlation vector between the array observation data and the beamformer output onto an orthogonal Krylov subspace, in order to update the steering vector mismatch with reduced complexity. The SCM of the array observation data is estimated by (4.5). The cross-correlation vector between the array observation data and the beamformer output can be expressed as $\mathbf{d} = E[\mathbf{x}y^*]$ (where $[.]^*$ denotes complex conjugation) or equivalently as

$$\mathbf{d} = E[(\mathbf{A}\mathbf{s} + \mathbf{n})(\mathbf{A}\mathbf{s} + \mathbf{n})^H \mathbf{w}]. \tag{4.6}$$

We assume and emphasize that when $\mathbf{w}$ is determined such that the interference are sufficiently canceled such that they felled much below the noise floor and σ_1^2 that they could be considered to be negligible, in which case we have $|\mathbf{a}_k^H \mathbf{w}| \ll |\mathbf{a}_1^H \mathbf{w}|$ for $k = 2, \cdots, K$ and all signals have zero mean, the cross-correlation vector $\mathbf{d}$ can be rewritten as

$$\mathbf{d} = E[(\mathbf{A}\mathbf{s} + \mathbf{n})(s_1^* \mathbf{a}_1^H \mathbf{w} + \mathbf{n}^H \mathbf{w})]. \tag{4.7}$$

Note that we also assume that the desired signal is uncorrelated from the interferers and the noise, i.e., $E[s_k s_1^*] = 0$ and $E[s_k \mathbf{a}_k s_1^* \mathbf{a}_1^H \mathbf{w}] = 0$ for $k = 2, \cdots, K$. With this assumption the desired signal power is not statistically affected by the interference and (4.7) can be rewritten as

$$\mathbf{d} = E[\sigma_1^2 \mathbf{a}_1^H \mathbf{w} \mathbf{a}_1 + \mathbf{n}\mathbf{n}^H \mathbf{w}], \tag{4.8}$$

where $\sigma_1^2 = |s_1 s_1^*| = |s_1|^2$, which can be estimated by the sample cross-correlation vector (SCV) given by

$$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^{i} \mathbf{x}(k) y^*(k). \tag{4.9}$$

4.3.1 Desired Signal Power Estimation

In this subsection, we describe an iterative method for the desired signal power (σ_1^2) estimation based on our work in [31], which can be accomplished by directly using the desired signal steering vector. In the adopted method, we need to choose an initial guess

for the steering vector mismatch within the presumed angular sector, say $\hat{\mathbf{a}}_1(0)$ and set $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$. By adding the snapshot index i, we have

$$\mathbf{x}(i) = \hat{\mathbf{a}}_1(i)s_1(i) + \sum_{k=2}^{K} \mathbf{a}_k(i)s_k(i) + \mathbf{n}(i), \qquad (4.10)$$

where $\hat{\mathbf{a}}_1(i)$ $(i = 1, 2, \cdots)$ designate the estimate of the initial guess of the steering vector at the ith snapshot.

Pre-multiplying the above equation by $\hat{\mathbf{a}}_1^H(i)$ we have

$$\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i) = \hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1(i) + \sum_{k=2}^{K} \hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i)s_k(i) + \mathbf{n}(i). \qquad (4.11)$$

Here we assume that each of the interferers is orthogonal or approximately orthogonal to the desired signal (i.e., the correlation coefficients between each of the interferers and the desired signal is close to zero). Specifically, the steering vector of each of the interferers is orthogonal $(\hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i) = 0, k = 2, 3, \cdots, K)$, or approximately orthogonal $(|\hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i)| \ll |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|, k = 2, 3, \cdots, K)$ to the desired signal steering vector (i.e., $\hat{\mathbf{a}}_1(i)$), so that $\hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i)$ $(k = 2, 3, \cdots, K)$ approaches zero and the term $\sum_{k=2}^{K} \hat{\mathbf{a}}_1^H(i)\mathbf{a}_k(i)s_k(i)$ in (4.11) can be neglected, resulting in

$$\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i) = \hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1(i) + \hat{\mathbf{a}}_1^H(i)\mathbf{n}(i). \qquad (4.12)$$

Taking the expectation of $|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2$, we obtain

$$E[|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2] = E[(\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1(i) + \hat{\mathbf{a}}_1^H(i)\mathbf{n}(i))^*(\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1(i) + \hat{\mathbf{a}}_1^H(i)\mathbf{n}(i))].$$
$$(4.13)$$

Assuming that the noise is statistically independent from the desired signal, then we have

$$E[|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2] = |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2 E[|s_1(i)|^2] + \hat{\mathbf{a}}_1^H(i)E[\mathbf{n}(i)\mathbf{n}^H(i)]\hat{\mathbf{a}}_1(i), \qquad (4.14)$$

where $E[\mathbf{n}(i)\mathbf{n}^H(i)]$ represents the noise covariance matrix $\mathbf{R}_n(i)$ that can be replaced by $\sigma_n^2 \mathbf{I}_M$, where the noise variance σ_n^2 can be easily estimated by a specific estimation method. A possible approach is to use a Maximum Likelihood (ML) based method as in [36]. Replacing the desired signal power $E[|s_1(i)|^2]$ and the noise variance σ_n^2 by their

estimates $\hat{\sigma}_1^2(i)$ and $\hat{\sigma}_n^2(i)$, respectively, we obtain

$$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\hat{\sigma}_n^2(i)}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}. \tag{4.15}$$

The expression in (4.15) has a low complexity ($\mathcal{O}(M)$) and can be directly implemented if the desired signal steering vector and the noise level are accurately estimated.

4.3.2 Orthogonal Krylov Subspace Approach for Steering Vector Mismatch Estimation

An orthogonal Krylov subspace strategy is proposed in order to estimate the mismatch with reduced cost and deal with situations in which the model order is time-varying. Our idea is based on constructing a linear system, which considers the steering vector mismatch as the solution, and solving it by using an iterative Krylov subspace projection method. To this end, consider a general high-dimensional linear system model given by

$$\mathbf{B}\mathbf{a}_1 = \mathbf{b}, \tag{4.16}$$

where $\mathbf{B} \in \mathbb{C}^{M \times M}$ and $\mathbf{b} \in \mathbb{C}^{M \times 1}$. Then we need to express $\mathbf{B}$ and $\mathbf{b}$ only using available information (known statistics or estimated parameters), so that we can solve the linear system with the Krylov subspace of order m ($m \ll M$) described by

$$\mathbf{K}_m = span\{\mathbf{b}, \mathbf{B}\mathbf{b}, \mathbf{B}^2\mathbf{b}, \cdots, \mathbf{B}^m\mathbf{b}\}. \tag{4.17}$$

Taking the complex conjugate of (4.12), we have

$$\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1^*(i) + \mathbf{n}^H(i)\hat{\mathbf{a}}_1(i). \tag{4.18}$$

Pre-multiplying both sides of (4.18) by the terms of (4.10), then adding an extra term $\delta\mathbf{I}\hat{\mathbf{a}}_1(i)$ (where δ is a small positive number defined by the user) and simplifying the terms, we obtain

$$(\mathbf{x}(i)\mathbf{x}^H(i) + \delta\mathbf{I})\hat{\mathbf{a}}_1(i) \approx \hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1(i)s_1^*(i) + \mathbf{n}(i)\mathbf{n}^H(i)\hat{\mathbf{a}}_1(i). \tag{4.19}$$

Replacing $\mathbf{x}(i)\mathbf{x}^H(i) + \delta\mathbf{I}$ by $\hat{\mathbf{R}}(i)$, $s_1(i)s_1^*(i)$ by $\hat{\sigma}_1^2(i)$ and $\mathbf{n}(i)\mathbf{n}^H(i)$ by $\hat{\sigma}_n^2(i)\mathbf{I}_M$, we obtain

$$\hat{\mathbf{R}}(i)\hat{\mathbf{a}}_1(i) \approx \underbrace{\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)\hat{\sigma}_1^2(i) + (\hat{\sigma}_n^2(i) + \delta)\hat{\mathbf{a}}_1(i)}_{\hat{\mathbf{b}}(i)}, \tag{4.20}$$

Table 4.1: Arnoldi-modified Gram-Schmidt algorithm

For $j = 1, 2, \cdots$ do:

    Compute $\mathbf{u}_j = \hat{\mathbf{R}}\mathbf{t}_j$

    For $l = 1, 2, \cdots, j$, do:

        $h_{l,j} = <\mathbf{u}_j, \mathbf{t}_l>$

        $\mathbf{u}_j = \mathbf{u}_j - h_{l,j}\mathbf{t}_l$

    End do.

[39]   Compute $h_{j,j+1} = \|\mathbf{u}_j\|$.

    If $h_{j,j+1} = 0$ or $j \geq K+1$,

        set $m = j$;

        break;

    Else compute $\mathbf{t}_{j+1} = \frac{\mathbf{u}_j}{h_{j,j+1}}$.

End do.

in which by further defining the expression on the right-hand side as $\hat{\mathbf{b}}(i)$, we can rewrite (4.20) as

$$\hat{\mathbf{R}}(i)\hat{\mathbf{a}}_1(i) \approx \hat{\mathbf{b}}(i). \tag{4.21}$$

As can be seen, (4.21) shares the same form as the linear system of equations in (4.16) and $\hat{\mathbf{b}}(i)$ can be expressed in terms of $\hat{\mathbf{a}}_1(i)$, $\hat{\sigma}_1^2(i)$ and $\hat{\sigma}_n^2(i)$ whereas $\hat{\mathbf{R}}(i)$ can be estimated by (4.5). In the following step, we employ the Arnoldi-modified Gram-Schmidt algorithm from the FOM method [38,39] associated with the minimum sufficient rank criterion discussed in [42] to compute an orthogonal Krylov subspace. We define a residue vector to represent the estimation error in the ith snapshot as

$$\hat{\mathbf{r}}(i) = \hat{\mathbf{b}}(i) - \hat{\mathbf{R}}(i)\hat{\mathbf{a}}_1(i), \tag{4.22}$$

and let

$$\mathbf{t}_1(i) = \frac{\hat{\mathbf{r}}(i)}{\|\hat{\mathbf{r}}(i)\|}. \tag{4.23}$$

Then the Krylov subspace bases can be computed using the modified Arnoldi-modified Gram-Schmidt algorithm as in Table 4.1 (the snapshot index i is omitted here for simplicity).

In Table 4.1, $<,>$ denotes the inner product of two vectors and the parameters $h_{l,j}$ $(l, j = 1, 2, \cdots, m)$ are real-valued coefficients, the model order is determined once if one of the following situations is satisfied:

- The execute-and-stop criterion [42] of the original Arnoldi-modified Gram-Schmidt algorithm is satisfied (i.e., $h_{j,j+1} = 0$).

- The minimum sufficient rank for addressing the SoI and interferers is achieved (i.e., $j \geq K + 1$, where K is the number of signal sources), so that no more subspace components are necessary for capturing the SoI from all the existing signal sources.

Now by inserting the snapshot index, we have

$$\hat{\mathbf{T}}(i) = [\mathbf{t}_1(i), \mathbf{t}_2(i), \cdots, \mathbf{t}_m(i)], \tag{4.24}$$

and the Krylov subspace projection matrix is computed by

$$\hat{\mathbf{P}}(i) = \hat{\mathbf{T}}(i)\hat{\mathbf{T}}^H(i). \tag{4.25}$$

It should be emphasized that the Krylov subspace matrix $\hat{\mathbf{T}}(i)$ obtained here is constructed by starting with the residue vector $\hat{\mathbf{r}}(i)$. In other words, $\hat{\mathbf{T}}(i)$ is constructed in a way that it consists of the estimation error of the steering vector. In order to extract the estimation error information and use it to update the steering vector mismatch, we can project the SCV $\hat{\mathbf{d}}(i)$ in (4.9) onto $\hat{\mathbf{P}}(i)$ and add the estimation error to the current estimate of $\hat{\mathbf{a}}_1(i)$ as

$$\hat{\mathbf{a}}_1(i + 1) = \hat{\mathbf{a}}_1(i) + \frac{\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)}{\|\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)\|}. \tag{4.26}$$

4.3.3 INC Matrix and Beamformer Weight Vector Computation

Since we have estimated both the desired signal power $\hat{\sigma}_1^2(i)$ and the mismatched steering vector in the previous subsections, the INC matrix can be obtained by subtracting the desired signal covariance matrix out from the SCM as

$$\hat{\mathbf{R}}_{I+N}(i) = \hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i). \tag{4.27}$$

The beamformer weight vector is computed by

$$\hat{\mathbf{w}}(i) = \frac{\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}, \tag{4.28}$$

which has a computationally costly matrix inversion $\hat{\mathbf{R}}_{I+N}^{-1}(i)$. The proposed OKSPME method is summarized in Table 4.2. In the next section, we will introduce adaptive algorithms to avoid matrix inversions and reduce the complexity.

4.4 Proposed Adaptive Algorithms

This section presents adaptive strategies based on the OKSPME robust beamforming method, resulting in the proposed OKSPME-SG, OKSPME-CCG and OKSPME-MCG algorithms, which are especially suitable for dynamic scenarios. In the proposed adaptive algorithms, we estimate the desired signal power and its steering vector with the same recursions as in OKSPME, whereas the estimation procedure of the beamforming weights is different. In particular, we start from a reformulated optimization problem and use SG and CG-based adaptive recursions to derive the weight update equations, which reduce the complexity by an order of magnitude as compared to that of OKSPME.

4.4.1 OKSPME-SG Adaptive Algorithm

We resort to an SG adaptive strategy and consider the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w}(i)}{\text{minimize}} \quad & \mathbf{w}^H(i)(\hat{\mathbf{R}}(i) - \hat{\mathbf{R}}_1(i))\mathbf{w}(i) \\ \text{subject to} \quad & \mathbf{w}^H(i)\hat{\mathbf{a}}_1(i) = 1, \end{aligned} \tag{4.29}$$

where $\hat{\mathbf{R}}(i)$ can be written as $\mathbf{x}(i)\mathbf{x}^H(i)$ and $\hat{\mathbf{R}}_1(i)$ represents the desired signal covariance matrix and can be written as $\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)$.

Then we can express the SG recursion as

$$\mathbf{w}(i+1) = \mathbf{w}(i) - \mu\frac{\partial\mathcal{L}}{\partial\mathbf{w}(i)}, \tag{4.30}$$

Table 4.2: Proposed OKSPME method

Initialization:

$\hat{\mathbf{w}}(0) = \mathbf{1}$;

Choose an initial guess $\hat{\mathbf{a}}_1(0)$ within the sector and set $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$;

For each snapshot $i = 1, 2, \cdots$:

$\hat{\mathbf{R}}(i) = \frac{1}{i} \sum\limits_{k=1}^{i} \mathbf{x}(k)\mathbf{x}^H(k)$

$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum\limits_{k=1}^{i} \mathbf{x}(k)y^*(k)$

Step 1. Compute the desired signal power

$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\hat{\sigma}_n^2(i)}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$

Step 2. Determine the Krylov subspace

$\hat{\mathbf{b}}(i) = \hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)\hat{\sigma}_1^2(i) + \hat{\sigma}_n^2(i)\hat{\mathbf{a}}_1(i)$

$\hat{\mathbf{r}}(i) = \hat{\mathbf{b}}(i) - \hat{\mathbf{R}}(i)\hat{\mathbf{a}}_1(i)$

$\mathbf{t}_1(i) = \frac{\hat{\mathbf{r}}(i)}{\|\hat{\mathbf{r}}(i)\|}$

Apply the algorithm in Table 4.1 to determine m and $\mathbf{t}_1(i), \cdots, \mathbf{t}_m(i)$

$\hat{\mathbf{T}}(i) = [\mathbf{t}_1(i), \mathbf{t}_2(i), \cdots, \mathbf{t}_m(i)]$

Step 3. Update the steering vector

$\hat{\mathbf{P}}(i) = \hat{\mathbf{T}}(i)\hat{\mathbf{T}}^H(i)$

$\hat{\mathbf{a}}_1(i+1) = \hat{\mathbf{a}}_1(i) + \frac{\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)}{\|\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)\|}$

$\hat{\mathbf{a}}_1(i+1) = \hat{\mathbf{a}}_1(i+1)/\|\hat{\mathbf{a}}_1(i+1)\|$

Step 4. Compute the weight vector

$\hat{\mathbf{R}}_{I+N}(i) = \hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)$

$\hat{\mathbf{w}}(i) = \frac{\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}$

End snapshot

---

where $\mathcal{L} = \mathbf{w}^H(i)(\mathbf{x}(i)\mathbf{x}^H(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) + \mathcal{R}e\{\lambda_{\mathcal{L}}\}(\mathbf{w}^H(i)\hat{\mathbf{a}}_1(i) - 1)$ and μ is the step size.

By substituting $\mathcal{L}$ into the SG equation (4.30) and letting $\mathbf{w}^H(i + 1)\hat{\mathbf{a}}_1(i + 1) = 1$, $\mathcal{R}e\{\lambda_{\mathcal{L}}\}$ is obtained as

$$\mathcal{R}e\{\lambda_{\mathcal{L}}\} = \frac{2(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i) - y(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i))}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)}. \tag{4.31}$$

By substituting $\mathcal{R}e\{\lambda_{\mathcal{L}}\}$ in (4.30) again, the weight update equation for OKSPME-SG is obtained as

$$\mathbf{w}(i + 1) = (\mathbf{I} - \mu\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i)$$
$$- \mu(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i) + y^*(i)(\mathbf{x}(i) - \frac{\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)})). \tag{4.32}$$

The adaptive SG recursion circumvents a matrix inversion when computing the weights using (4.28), which is unavoidable in OKSPME. Therefore, the computational complexity is reduced from $\mathcal{O}(M^3)$ in OKSPME to $\mathcal{O}(M^2)$ in OKSPME-SG. It is also important that the step size μ should satisfy $0 < \mu < \frac{1}{\hat{\sigma}_1^2(i)}$ to guarantee that $\mathbf{I} - \mu\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)$ is always a positive-definite matrix so that (4.32) is ensured converging to a solution. To implement OKSPME-SG, **Step 1**, **Step 2** and **Step 3** from Table 4.2 and (4.32) are required.

4.4.2 OKSPME-CCG Adaptive Algorithm

In this subsection, the OKSPME-CCG algorithm is proposed. In CG-based approaches, we usually employ a forgetting factor (e.g. λ) to estimate the second-order statistics of the data or the SCM [1, 24], which can be expressed by

$$\hat{\mathbf{R}}(i) = \lambda\hat{\mathbf{R}}(i - 1) + \mathbf{x}(i)\mathbf{x}^H(i), \tag{4.33}$$

whereas the SCV $\hat{\mathbf{d}}(i)$ can be estimated with the same forgetting factor as described by

$$\hat{\mathbf{d}}(i) = \lambda\hat{\mathbf{d}}(i - 1) + \mathbf{x}(i)y^*(i). \tag{4.34}$$

The proposed optimization problem that leads to the OKSPME-CCG algorithm is described by

$$\underset{\hat{\mathbf{a}}_1(i),\mathbf{v}(i)}{\text{minimize}} \mathcal{J} = \mathbf{v}^H(i)(\hat{\mathbf{R}}(i) - \hat{\mathbf{R}}_1(i))\mathbf{v}(i) - \hat{\mathbf{a}}_1^H(i)\mathbf{v}(i), \tag{4.35}$$

where $\mathbf{v}(i)$ is the CG-based weight vector. In OKSPME-CCG, we require N iterations for each snapshot. In the nth iteration, $\hat{\mathbf{a}}_{1,n}(i)$ and $\mathbf{v}_n(i)$ are updated as follows

$$\hat{\mathbf{a}}_{1,n}(i) = \hat{\mathbf{a}}_{1,n-1}(i) + \alpha_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \tag{4.36}$$

$$\mathbf{v}_n(i) = \mathbf{v}_{n-1}(i) + \alpha_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i), \tag{4.37}$$

where $\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)$ and $\mathbf{p}_{\mathbf{v},n}(i)$ are direction vectors updated by

$$\mathbf{p}_{\hat{\mathbf{a}}_1,n+1}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) + \beta_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \tag{4.38}$$

$$\mathbf{p}_{\mathbf{v},n+1}(i) = \mathbf{g}_{\mathbf{v},n}(i) + \beta_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i), \tag{4.39}$$

where $\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)$ and $\mathbf{g}_{\mathbf{v},n}(i)$ are the negative gradients of the cost function in terms of $\hat{\mathbf{a}}_1(i)$ and $\mathbf{v}(i)$, respectively, which are expressed as

$$\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) = -\frac{\partial \mathcal{J}}{\partial \hat{\mathbf{a}}_{1,n}(i)} = \hat{\sigma}_1^2(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\hat{\mathbf{a}}_{1,n}(i) + \mathbf{v}_n(i), \tag{4.40}$$

$$\mathbf{g}_{\mathbf{v},n}(i) = -\frac{\partial \mathcal{J}}{\partial \mathbf{v}_n(i)} = \mathbf{g}_{\mathbf{v},n-1}(i) - \alpha_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\mathbf{x}(i)\mathbf{x}^H(i))\mathbf{p}_{\mathbf{v},n}(i). \tag{4.41}$$

The scaling parameters $\alpha_{\hat{\mathbf{a}}_1,n}(i)$, $\alpha_{\mathbf{v},n}(i)$ can be obtained by substituting (4.36) and (4.37) into (4.35) and minimizing the cost function with respect to $\alpha_{\hat{\mathbf{a}}_1,n}(i)$ and $\alpha_{\mathbf{v},n}(i)$, respectively. The solutions are given by

$$\alpha_{\hat{\mathbf{a}}_1,n}(i) = -\frac{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}{\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}, \tag{4.42}$$

$$\alpha_{\mathbf{v},n}(i) = \frac{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{p}_{\mathbf{v},n}(i)}{\mathbf{p}_{\mathbf{v},n}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_{1,n}(i)\hat{\mathbf{a}}_{1,n}^H(i))\mathbf{p}_{\mathbf{v},n}(i)}. \tag{4.43}$$

The parameters $\beta_{\hat{\mathbf{a}}_1,n}(i)$ and $\beta_{\mathbf{v},n}(i)$ should be chosen to provide conjugacy for direction vectors [24], which results in

$$\beta_{\hat{\mathbf{a}}_1,n}(i) = \frac{\mathbf{g}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}(i)}, \tag{4.44}$$

$$\beta_{\mathbf{v},n}(i) = \frac{\mathbf{g}_{\mathbf{v},n}^H(i)\mathbf{g}_{\mathbf{v},n}(i)}{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{g}_{\mathbf{v},n-1}(i)}. \tag{4.45}$$

After $\hat{\mathbf{a}}_{1,n}(i)$ and $\mathbf{v}_n(i)$ are updated for N iterations, the beamforming weight vector $\mathbf{w}(i)$ can be computed by

$$\mathbf{w}(i) = \frac{\mathbf{v}_N(i)}{\hat{\mathbf{a}}_{1,N}^H(i)\mathbf{v}_N(i)} \tag{4.46}$$

The computational cost of OKSPME-CCG algorithm is $\mathcal{O}(NM^2)$, which is higher than the cost required in OKSPME-SG due to the inner iterations at every snapshot. The proposed OKSPME-CCG is summarized in Table 4.3.

4.4.3 OKSPME-MCG Adaptive Algorithm

In OKSPME-MCG, we let only one iteration be performed per snapshot, which further reduces the complexity compared to OKSPME-CCG. Here we denote the CG-based weights and steering vector updated by snapshots rather than inner iterations as

$$\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i-1) + \alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i), \tag{4.47}$$

$$\mathbf{v}(i) = \mathbf{v}(i-1) + \alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i). \tag{4.48}$$

As can be seen, the subscripts of all the quantities for inner iterations are eliminated. Then, we employ the degenerated scheme to ensure $\alpha_{\hat{\mathbf{a}}_1}(i)$ and $\alpha_{\mathbf{v}}(i)$ satisfy the convergence bound [24] given by

$$0 \leq \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i) \leq 0.5\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1), \tag{4.49}$$

$$0 \leq \mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i) \leq 0.5\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1). \tag{4.50}$$

Instead of updating the negative gradient vectors $\mathbf{g}_{\hat{\mathbf{a}}_1}(i)$ and $\mathbf{g}_{\mathbf{v}}(i)$ in iterations, now we utilize the forgetting factor to re-express them in one snapshot as

$$\mathbf{g}_{\hat{\mathbf{a}}_1}(i) = (1-\lambda)\mathbf{v}(i) + \lambda\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1) + \hat{\sigma}_1^2(i)\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i), \tag{4.51}$$

$$\mathbf{g}_{\mathbf{v}}(i) = 1-\lambda)\hat{\mathbf{a}}_1(i) + \lambda\mathbf{g}_{\mathbf{v}}(i-1) - \alpha_{\mathbf{v}}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{v}(i-1). \tag{4.52}$$

Table 4.3: Proposed OKSPME-CCG algorithm

Initialization:

$\hat{\mathbf{w}}(1) = \mathbf{v}_0(1) = \mathbf{1}$; λ;

Choose an initial guess $\hat{\mathbf{a}}_1(0)$ within the sector and set $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$;

For each snapshot $i = 1, 2, \cdots$:

$\quad \hat{\mathbf{R}}(i) = \frac{1}{i} \sum\limits_{k=1}^{i} \mathbf{x}(k)\mathbf{x}^H(k)$

$\quad \hat{\mathbf{d}}(i) = \frac{1}{i} \sum\limits_{k=1}^{i} \mathbf{x}(k)y^*(k)$

Step 1 from Table 4.2

Step 2 from Table 4.2

Step 3 from Table 4.2

Steering Vector and Weight Vector Estimations

$\hat{\mathbf{a}}_{1,0}(i) = \hat{\mathbf{a}}_1(i)$

$\mathbf{g}_{\hat{\mathbf{a}}_1,0}(i) = \hat{\sigma}_1^2(i)\mathbf{v}_0(i)\mathbf{v}_0^H(i)\hat{\mathbf{a}}_{1,0}(i) + \mathbf{v}_0(i)$

$\mathbf{g}_{\mathbf{v},0}(i) = \hat{\mathbf{a}}_{1,0}(i) - \hat{\mathbf{R}}(i)\mathbf{v}_0(i)$

$\mathbf{p}_{\hat{\mathbf{a}}_1,0}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,0}(i)$; $\mathbf{p}_{\mathbf{v},0}(i) = \mathbf{g}_{\mathbf{v},0}(i)$

For each iteration index $n = 1, 2, \cdots, N$:

$\quad \alpha_{\hat{\mathbf{a}}_1,n}(i) = -\dfrac{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}{\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}$

$\quad \alpha_{\mathbf{v},n}(i) = \dfrac{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{p}_{\mathbf{v},n}(i)}{\mathbf{p}_{\mathbf{v},n}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_{1,n}(i)\hat{\mathbf{a}}_{1,n}^H(i))\mathbf{p}_{\mathbf{v},n}(i)}$

$\quad \hat{\mathbf{a}}_{1,n}(i) = \hat{\mathbf{a}}_{1,n-1}(i) + \alpha_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)$

$\quad \mathbf{v}_n(i) = \mathbf{v}_{n-1}(i) + \alpha_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i)$

$\quad \mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) = \hat{\sigma}_1^2(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\hat{\mathbf{a}}_{1,n}(i) + \mathbf{v}_n(i)$

$\quad \mathbf{g}_{\mathbf{v},n}(i) = \mathbf{g}_{\mathbf{v},n-1}(i) - \alpha_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\mathbf{x}(i)\mathbf{x}^H(i))\mathbf{p}_{\mathbf{v},n}(i)$

$\quad \beta_{\hat{\mathbf{a}}_1,n}(i) = \dfrac{\mathbf{g}_{\hat{\mathbf{a}}_1,n}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}(i)}$

$\quad \beta_{\mathbf{v},n}(i) = \dfrac{\mathbf{g}_{\mathbf{v},n}^H(i)\mathbf{g}_{\mathbf{v},n}(i)}{\mathbf{g}_{\mathbf{v},n-1}^H(i)\mathbf{g}_{\mathbf{v},n-1}(i)}$

$\quad \mathbf{p}_{\hat{\mathbf{a}}_1,n+1}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) + \beta_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)$

$\quad \mathbf{p}_{\mathbf{v},n+1}(i) = \mathbf{g}_{\mathbf{v},n}(i) + \beta_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i)$

End iteration

$\mathbf{v}_0(i+1) = \mathbf{v}_N(i)$

$\mathbf{w}(i) = \dfrac{\mathbf{v}_N(i)}{\hat{\mathbf{a}}_{1,N}^H(i)\mathbf{v}_N(i)}$

End snapshot

Pre-multiplying (4.51) and (4.52) by $\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)$ and $\mathbf{p}_{\mathbf{v}}^H(i)$, respectively, and taking expectations we obtain

$$E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i)] = E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)(\mathbf{v}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1)(i)] + \lambda E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]$$
$$- \lambda E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)] + E[\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\hat{\sigma}_1^2(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)], \quad (4.53)$$

$$E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i)] = \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]$$
$$- E[\alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)], \quad (4.54)$$

where in (4.54) we have $E[\hat{\mathbf{R}}(i)\mathbf{v}(i-1)] = E[\hat{\mathbf{a}}_1(i)]$. After substituting (4.54) in (4.50) we obtain the bounds for $\alpha_{\mathbf{v}}(i)$ as follows

$$\frac{(\lambda - 0.5)E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]}{E[\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)]} \leq E[\alpha_{\mathbf{v}}(i)]$$
$$\leq \frac{\lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]}{E[\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)]}. \quad (4.55)$$

Then we can introduce a constant parameter $\eta_{\mathbf{v}} \in [0, 0.5]$ to restrict $\alpha_{\mathbf{v}}(i)$ within the bounds in (4.55) as

$$\alpha_{\mathbf{v}}(i) = \frac{\lambda(\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1) - \mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)) - \eta_{\mathbf{v}}\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)}{\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)}. \quad (4.56)$$

Similarly, we can also obtain the bounds for $\alpha_{\hat{\mathbf{a}}_1}(i)$. For simplicity let us define $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)] = A$, $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)] = B$, $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)] = C$ and $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\hat{\sigma}_1^2(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)] = D$. Substituting (4.53) into (4.49) gives

$$\frac{\lambda(B - A) - B + C}{D} \leq E[\alpha_{\hat{\mathbf{a}}_1}(i)] \leq \frac{\lambda(B - A) - B + C + 0.5A}{D}, \quad (4.57)$$

in which we can introduce another constant parameter $\eta_{\hat{\mathbf{a}}_1} \in [0, 0.5]$ to restrict $\alpha_{\hat{\mathbf{a}}_1}(i)$ within the bounds in (4.57) as

$$E[\alpha_{\hat{\mathbf{a}}_1}(i)] = \frac{\lambda(B - A) - B + C + \eta_{\hat{\mathbf{a}}_1}A}{D}, \quad (4.58)$$

or

$$\alpha_{\hat{\mathbf{a}}_1}(i) = [\lambda(\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)$$
$$+ \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i) + \eta_{\hat{\mathbf{a}}_1}\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]/[\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)]. \quad (4.59)$$

Then we can update the direction vectors $\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$ and $\mathbf{p}_\mathbf{v}(i)$ by

$$\mathbf{p}_{\hat{\mathbf{a}}_1}(i+1) = \mathbf{g}_{\hat{\mathbf{a}}_1}(i) + \beta_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i), \tag{4.60}$$

$$\mathbf{p}_\mathbf{v}(i+1) = \mathbf{g}_\mathbf{v}(i) + \beta_\mathbf{v}(i)\mathbf{p}_\mathbf{v}(i), \tag{4.61}$$

where $\beta_{\hat{\mathbf{a}}_1}(i)$ and $\beta_\mathbf{v}(i)$ are updated by

$$\beta_{\hat{\mathbf{a}}_1}(i) = \frac{[\mathbf{g}_{\hat{\mathbf{a}}_1}(i) - \mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]^H \mathbf{g}_{\hat{\mathbf{a}}_1}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1}^H(i-1)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)}, \tag{4.62}$$

$$\beta_\mathbf{v}(i) = \frac{[\mathbf{g}_\mathbf{v}(i) - \mathbf{g}_\mathbf{v}(i-1)]^H \mathbf{g}_\mathbf{v}(i)}{\mathbf{g}_\mathbf{v}^H(i-1)\mathbf{g}_\mathbf{v}(i-1)}. \tag{4.63}$$

Finally we can update the beamforming weights by

$$\mathbf{w}(i) = \frac{\mathbf{v}(i)}{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)}, \tag{4.64}$$

The MCG approach employs the forgetting factor λ and constant η for estimating $\alpha(i)$, which means its performance may depend on a suitable choice of these parameters. The proposed OKSPME-MCG algorithm requires a complexity of $\mathcal{O}(M^2)$. However, the cost is usually much lower compared to CCG approach for the elimination of inner recursions and it presents a similar performance in most studied scenarios. From an implementation point of view, the choice of using the CCG and MCG algorithms is based on the stationarity of the system: the CCG algorithm is more suitable for scenarios in which the system is stationary and we can compute the beamformer with K iterations while the MCG algorithm is suggested for non-stationary scenarios as we only run one iteration per snapshot and can track variations in the environment. Table 4.4 summarizes the OKSPME-MCG algorithm.

4.5 Analysis

In this section, we present an analysis of the following aspects of the proposed and existing algorithms:

- An analysis of the MSE between the estimated and actual steering vectors for the general approach that employs a presumed angular sector.

Table 4.4: Proposed OKSPME-MCG algorithm

Initialization:

$\hat{\mathbf{w}}(1) = \mathbf{v}(0) = \mathbf{1}$; λ; $\eta_{\mathbf{v}} = \eta_{\hat{\mathbf{a}}_1}$;

Choose an initial guess $\hat{\mathbf{a}}_1(0)$ within the sector and set $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$;

$\mathbf{g}_{\mathbf{v}}(0) = \mathbf{p}_{\mathbf{v}}(1) = \hat{\mathbf{a}}_1(1)$; $\mathbf{g}_{\hat{\mathbf{a}}_1}(0) = \mathbf{p}_{\hat{\mathbf{a}}_1}(1) = \mathbf{v}(0)$;

For each snapshot $i = 1, 2, \cdots$:

$\hat{\mathbf{R}}(i) = \frac{1}{i} \sum_{k=1}^{i} \mathbf{x}(k)\mathbf{x}^H(k)$

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

Step 1 from Table 4.2

Step 2 from Table 4.2

Step 3 from Table 4.2

Steering Vector and Weight Vector Estimations

$\alpha_{\hat{\mathbf{a}}_1}(i) = [\lambda(\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)$

$+\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i) + \eta_{\hat{\mathbf{a}}_1}\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]$

$/[\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)]$

$\alpha_{\mathbf{v}}(i) = \frac{\lambda(\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1) - \mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)) - \eta_{\mathbf{v}}\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)}{\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)}$

$\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i-1) + \alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$

$\mathbf{v}(i) = \mathbf{v}(i-1) + \alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i)$

$\mathbf{g}_{\hat{\mathbf{a}}_1}(i) = (1-\lambda)\mathbf{v}(i) + \lambda\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)$

$+\hat{\sigma}_1^2(i)\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)$

$\mathbf{g}_{\mathbf{v}}(i) = (1-\lambda)\hat{\mathbf{a}}_1(i) + \lambda\mathbf{g}_{\mathbf{v}}(i-1) - \alpha_{\mathbf{v}}(i)(\hat{\mathbf{R}}(i)$

$-\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{v}(i-1)$

$\beta_{\hat{\mathbf{a}}_1}(i) = \frac{[\mathbf{g}_{\hat{\mathbf{a}}_1}(i) - \mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]^H\mathbf{g}_{\hat{\mathbf{a}}_1}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1}^H(i-1)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)}$

$\beta_{\mathbf{v}}(i) = \frac{[\mathbf{g}_{\mathbf{v}}(i) - \mathbf{g}_{\mathbf{v}}(i-1)]^H\mathbf{g}_{\mathbf{v}}(i)}{\mathbf{g}_{\mathbf{v}}^H(i-1)\mathbf{g}_{\mathbf{v}}(i-1)}$

$\mathbf{p}_{\hat{\mathbf{a}}_1}(i+1) = \mathbf{g}_{\hat{\mathbf{a}}_1}(i) + \beta_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$

$\mathbf{p}_{\mathbf{v}}(i+1) = \mathbf{g}_{\mathbf{v}}(i) + \beta_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i)$

$\mathbf{w}(i) = \frac{\mathbf{v}(i)}{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)}$

End snapshot

- MSE analysis results of the proposed OKSPME method and the SQP method in [12] and their relationships and differences.

- A complexity analysis for the proposed and existing algorithms.

4.5.1 MSE analysis

Firstly, we present the MSE analysis of the general approach that employs a presumed angular sector. Since we have the steering vector estimate $\hat{\mathbf{a}}_1(i)$ in the ith snapshot, by denoting the true steering vector as $\mathbf{a}_1$, we can express the MSE of the estimate $\hat{\mathbf{a}}_1(i)$ as

$$\text{MSE}\{\hat{\mathbf{a}}_1(i)\} = \text{tr}(E[(\hat{\mathbf{a}}_1(i){-}\mathbf{a}_1)(\hat{\mathbf{a}}_1(i){-}\mathbf{a}_1)^H]) = E[(\hat{\mathbf{a}}_1(i){-}\mathbf{a}_1)^H(\hat{\mathbf{a}}_1(i){-}\mathbf{a}_1)]. \quad (4.65)$$

In the approach that employs an angular sector, we usually choose an initial guess (i.e., $\hat{\mathbf{a}}_1(0)$) from the presumed sector. Let us express the accumulated estimation error as

$$\hat{\mathbf{e}}(i) = \hat{\mathbf{a}}_1(i) - \hat{\mathbf{a}}_1(0), \quad (4.66)$$

then (4.65) can be rewritten as

$$\text{MSE}\{\hat{\mathbf{a}}_1(i)\} = E[(\hat{\mathbf{a}}_1(0) + \hat{\mathbf{e}}(i) - \mathbf{a}_1)^H(\hat{\mathbf{a}}_1(0) + \hat{\mathbf{e}}(i) - \mathbf{a}_1)]. \quad (4.67)$$

The initial guess $\hat{\mathbf{a}}_1(0)$ can be described as the true steering vector plus a guess error vector (i.e., ϵ):

$$\hat{\mathbf{a}}_1(0) = \mathbf{a}_1 + \epsilon. \quad (4.68)$$

Taking expectation of both sides of the above, we have

$$E[\hat{\mathbf{a}}_1(0)] = \mathbf{a}_1 + E[\epsilon]. \quad (4.69)$$

Substituting (4.68) into (4.67), taking into account that the accumulated estimation error is uncorrelated with the initial guess error and simplifying the expression, we obtain

$$\text{MSE}\{\hat{\mathbf{a}}_1(i)\} = E[\epsilon^H \epsilon] + E[\epsilon^H]E[\hat{\mathbf{e}}(i)] + E[\hat{\mathbf{e}}^H(i)]E[\epsilon] + E[\hat{\mathbf{e}}^H(i)\hat{\mathbf{e}}(i)]. \quad (4.70)$$

Furthermore, it should be emphasized that both ϵ and $\hat{\mathbf{e}}(i)$ are in vector forms, which means that their second-order statistics can be re-expressed in terms of their first-order statistics of their Euclidean norms. Then we can re-express (4.70) as

$$\text{MSE}\{\hat{\mathbf{a}}_1(i)\} = E[\|\epsilon\|^2] + E[\|\hat{\mathbf{e}}(i)\|^2] + 2E[\epsilon^H]E[\hat{\mathbf{e}}(i)]. \quad (4.71)$$

Since both $\|\epsilon\|$ and $\|\hat{\mathbf{e}}(i)\|$ are scalars we have

$$E[\|\epsilon\|^2] = \text{Var}[\|\epsilon\|] + E^2[\|\epsilon\|], \tag{4.72}$$

$$E[\|\hat{\mathbf{e}}(i)\|^2] = \text{Var}[\|\hat{\mathbf{e}}(i)\|] + E^2[\|\hat{\mathbf{e}}(i)\|]. \tag{4.73}$$

At this stage, we can employ Popoviciu's inequality [61] to obtain the upper bounds for the variances of the norms of the random vectors ϵ and $\hat{\mathbf{e}}(i)$, which are given by

$$\text{Var}[\|\epsilon\|] \leq \frac{(\sup\|\epsilon\| - \inf\|\epsilon\|)^2}{4}, \tag{4.74}$$

$$\text{Var}[\|\hat{\mathbf{e}}(i)\|] \leq \frac{(\sup\|\hat{\mathbf{e}}(i)\| - \inf\|\hat{\mathbf{e}}(i)\|)^2}{4}. \tag{4.75}$$

However, the last term in (4.71) is not analytical when conducting a norm analysis. Actually, $E[\epsilon]$ depends on how the presumed sector is chosen: if the sector is chosen in an unbiased manner (i.e., the true steering vector lies in the centre of the sector), then we have $E[\epsilon] = \mathbf{0}$ by symmetry criterion, in which case we can omit the last terms of (4.71). For convenience of carrying out the norm analysis as the next step, we focus on the unbiased case only, so that the MSE only depends on the expectation, the infimum and the supremum of $\|\epsilon\|$ and $\|\hat{\mathbf{e}}(i)\|$. In Fig. 4.1, we utilize Euclidean geometry to illustrate the relationships among the norms of the errors and the norm of the steering vector, which is a fixed parameter due to the re-normalization procedure after it is estimated each time.

According to Fig. 4.1, we can use θ (i.e., half of the angular sector, assumed less than $\pi/4$) and $\|\mathbf{a}_1\|$ to obtain $E[\|\epsilon\|]$ by the following (any angular parameter appeared in the equations should be measured in radians rather than degrees): $\|\epsilon\|$ is equivalent to the chord length which corresponds to the arc of a variable τ, which can be any value from 0 to θ with equal probability, in other words, the choice of τ is uniformly distributed within $[0, \theta]$. If the sample size of the selected ϵ is large enough, we can approximately describe its probability density function (pdf) as a continuous function given by

$$f(\tau) = \frac{1}{\theta}. \tag{4.76}$$

Meanwhile, we are also able to calculate the chord length $\|\epsilon\|$ from a simple geometric criterion as

$$\|\epsilon\| = 2\|\mathbf{a}_1\| \sin\frac{\tau}{2}. \tag{4.77}$$

Then the expectation of $\|\epsilon\|$ can be computed by

$$E[\|\epsilon\|] = \int_0^\theta \|\epsilon\| f(\tau) d\tau, \tag{4.78}$$

from which after simplification we obtain

$$E[\|\epsilon\|] = \frac{8\|\mathbf{a}_1\|\sin^2\frac{\theta}{4}}{\theta}. \tag{4.79}$$

At this point, we can also compute the variance of $\|\epsilon\|$ by using (4.79) as

$$\mathrm{Var}[\|\epsilon\|] = \int_0^\theta (\|\epsilon\| - E[\|\epsilon\|])^2 \mathrm{f}(\tau)\mathrm{d}\tau, \tag{4.80}$$

from which after simplification we obtain

$$\mathrm{Var}[\|\epsilon\|] = 2\|\mathbf{a}_1\|^2(1 - \frac{\sin\theta}{\theta} - \frac{32\sin^4\frac{\theta}{4}}{\theta^2}). \tag{4.81}$$

In addition, it is clear that we have $\inf\|\epsilon\| = 0$ and $\sup\|\epsilon\| = 2\|\mathbf{a}_1\|\sin\frac{\theta}{2}$, which can be substituted in (4.74) and result in

$$\mathrm{Var}[\|\epsilon\|] \le \|\mathbf{a}_1\|^2\sin^2\frac{\theta}{2}. \tag{4.82}$$

We can see that the right-hand side of (4.81) satisfies the inequality in (4.82). After substituting (4.79) and (4.81) in (4.72), we obtain

$$E[\|\epsilon\|^2] = 2\|\mathbf{a}_1\|^2(1 - \frac{\sin\theta}{\theta}). \tag{4.83}$$
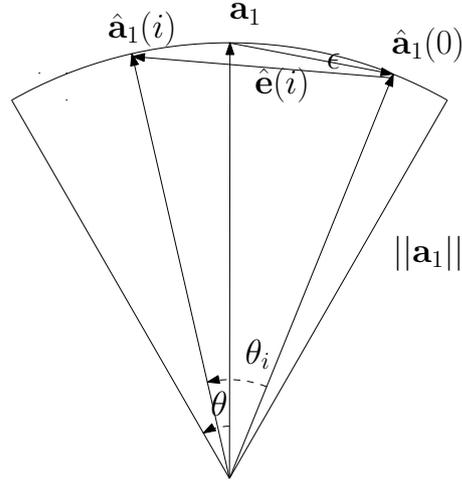
Figure 4.1: Euclidean norm interpretation of the MSE

Regarding the computation of the norm of the accumulated estimation error $\|\hat{\mathbf{e}}(i)\|$, we need to emphasize that even though the steering vector is always re-normalized each time after it is updated, the piecewise estimation error in each snapshot does not directly

84
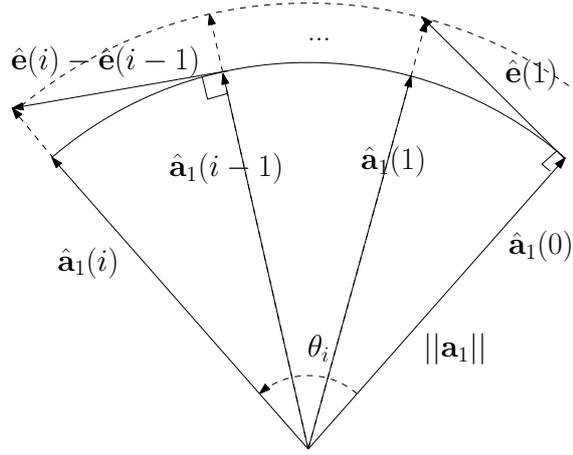
Figure 4.2: Update scheme of the SQP method
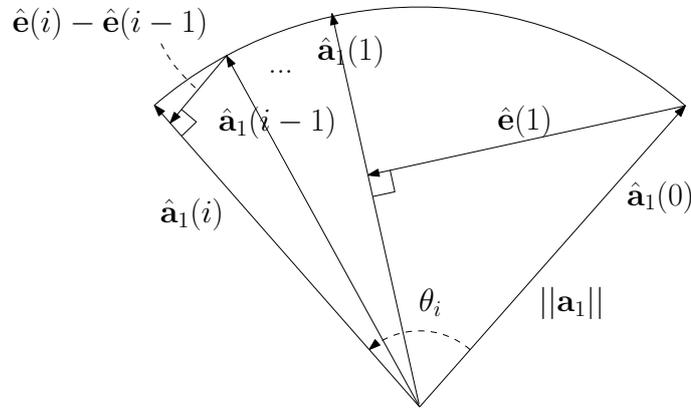
Figure 4.3: Update scheme of the OKSPME method

update the steering vector to its normalized version, which means it is inappropriate to calculate the estimation error by geometric methods directly from Fig. 4.1 because the accumulated estimation error partially comes from the unnormalized steering vectors. However, we can obtain the infimum and supremum values for $\|\hat{\mathbf{e}}(i)\|$ if we assume the update scheme is unidirectional (i.e., the steering vector is updated from $\hat{\mathbf{a}}_1(0)$ to $\hat{\mathbf{a}}_1(i)$ in a single direction within the sector), with the unnormalized steering vectors considered.

We firstly look at the SQP method scenario in [12]. The steering vector update scheme is shown in Fig. 4.2. It is necessary to emphasize that now we focus on the angular sector range of θ_i (i.e., the angle difference between the initially guessed steering vector and its estimate in the ith snapshot) rather than θ. In [12], an online-optimization program was used to iteratively solve for the piecewise estimation error in every snapshot, which was

85

always orthogonal to the current steering vector estimate. Let us consider that at each time instant the steering vector is updated, its direction changes by $\theta_{i,k}$, where i is the snapshot index and k ($1 \leq k \leq i$) is the index for the kth update. Since the total direction change in a snapshot is θ_i, then we have

$$\theta_i = \sum_{k=1}^{i} \theta_{i,k}, \tag{4.84}$$

and the norm of the accumulated estimation error is no greater than the sum of the norms of all the piecewise estimation errors, which is given by the inequality

$$\|\hat{\mathbf{e}}(i)\| \leq \sum_{k=1}^{i} \|\mathbf{a}_1\| \tan \theta_{i,k}. \tag{4.85}$$

If we assume θ_i is less than $\pi/2$, then the right-hand side of (4.85) achieves its maximum value when $\theta_{i,k} = \tan \theta_i$, which is also the supremum of $\|\hat{\mathbf{e}}(i)\|$ and equals

$$\|\hat{\mathbf{e}}(i)\|_{max} = \|\mathbf{a}_1\| \tan \theta_i. \tag{4.86}$$

On the other hand, we notice that the piecewise estimation error vector can never enter into the angular sector, but at most move along with the arc if the number of iterations is large enough. In this case, we can approximately and geometrically illustrate the arc length corresponding with θ_i as the lower bound by taking the limit $i \to \infty$, i.e.,

$$\lim_{i \to \infty} \|\hat{\mathbf{e}}(i)\| = \theta_i \|\mathbf{a}_1\|, \tag{4.87}$$

which is actually the infimum of $\|\hat{\mathbf{e}}(i)\|$ and cannot be achieved since the number of snapshots or iterations are always limited in practical situations. By combining (4.86) and (4.87), $\|\hat{\mathbf{e}}(i)\|$ is bounded by

$$\inf\|\hat{\mathbf{e}}(i)\| = \theta_i\|\mathbf{a}_1\| < \|\hat{\mathbf{e}}(i)\| \leq \|\mathbf{a}_1\| \tan \theta_i = \sup\|\hat{\mathbf{e}}(i)\|. \tag{4.88}$$

Different from the SQP method, the proposed OKSPME method utilizes the Krylov subspace and the cross-correlation vector projection approach to extract the error information then use it to update the steering vector. From (4.9) we have

$$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^{i} \mathbf{x}(k)y^*(k) = \frac{1}{i} \sum_{k=1}^{i} \mathbf{x}(k)(\mathbf{w}^H(k)\mathbf{x}(k))^*$$

$$= \frac{1}{i} \sum_{k=1}^{i} \mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}(k) = \frac{1}{i} \sum_{k=1}^{i} \hat{\mathbf{R}}(k)\mathbf{w}(k). \tag{4.89}$$

Note that an initialization for vector $\hat{\mathbf{d}}$ or matrix $\hat{\mathbf{R}}$ should be considered to ensure $\hat{\mathbf{R}}$ is full-rank and invertible, which can be done by either setting $\hat{\mathbf{d}}(0) = \delta \mathbf{I} \mathbf{w}(0)$ or $\hat{\mathbf{R}}(0) = \delta \mathbf{I}$. We also know that

$$\mathbf{w}(k) = \frac{\hat{\mathbf{R}}^{-1}(k)\hat{\mathbf{a}}_1(k)}{\hat{\mathbf{a}}_1^H(k)\hat{\mathbf{R}}^{-1}(k)\hat{\mathbf{a}}_1(k)} = \frac{\hat{\mathbf{R}}^{-1}(k)\hat{\mathbf{a}}_1(k)}{\hat{\sigma}_1^2(k)}. \tag{4.90}$$

Pre-multiplying (4.90) by $\hat{\mathbf{R}}(k)$ on both sides we obtain

$$\hat{\mathbf{R}}(k)\mathbf{w}(k) = \frac{\hat{\mathbf{a}}_1(k)}{\hat{\sigma}_1^2(k)}, \tag{4.91}$$

which is then substituted in (4.89) and results in

$$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^{i} \frac{\hat{\mathbf{a}}_1(k)}{\hat{\sigma}_1^2(k)}, \tag{4.92}$$

where $\hat{\sigma}_1^2(k)$ is a scalar, which means the SCV contains the direction of the desired signal steering vector. Projecting $\hat{\mathbf{d}}(i)$ onto the Krylov subspace represented by $\hat{\mathbf{P}}(i)$ is therefore similar to projecting $\hat{\mathbf{a}}_1(i)$. In our method, the estimation of $\hat{\mathbf{d}}(i)$ is separate from the update of $\hat{\mathbf{a}}_1(i)$, which means the steering vector estimation error used for the updates is obtained from $\hat{\mathbf{d}}(i)$, so that in the kth ($1 \leq k < i$) snapshot, the error does not have to be orthogonal to $\hat{\mathbf{a}}_1(k)$, but should be orthogonal to another potentially better estimate $\hat{\mathbf{a}}_1(j)$ ($1 \leq k < j \leq i$), resulting in a situation where the error is located inside the sector (see Fig. 4.3). There are two benefits in the case which the error is inside the sector: faster convergence rate and smaller estimation error. We can obtain the infimum and supremum values in a similar way. By applying the inequality that the norm of the accumulated estimation error is no greater than the sum of the norms of all the piecewise estimation errors, we have

$$\|\hat{\mathbf{e}}(i)\| \leq \sum_{k=1}^{i} \|\mathbf{a}_1\| \sin \theta_{i,k}, \tag{4.93}$$

where the parameters $\theta_{i,k}$ ($k = 1, 2, \cdots, i$) satisfy the constraint in (4.84). However, the right-hand side of (4.93) achieves its maximum value when all these parameters are equal (i.e., $\theta_{i,1} = \theta_{i,2} = \cdots = \theta_{i,i} = \frac{\theta_i}{i}$) and it is given by

$$\|\hat{\mathbf{e}}(i)\|_{max} = i\|\mathbf{a}_1\| \sin \frac{\theta_i}{i}, \tag{4.94}$$

The right-hand side of (4.94) can be treated as a function of i which is an increasing function on $i = 1, 2, \cdots, \infty$. Therefore, we can take the limit of it to obtain the upper bound of $\|\hat{\mathbf{e}}(i)\|_{max}$, and so as to $\|\hat{\mathbf{e}}(i)\|$. In fact, when $i \to \infty$, the piecewise estimation

error moves along the arc corresponding with θ_i, resulting in the upper bound obtained is the same as the lower bound of the SQP method case, which is given by the right-hand side expression of (4.87) and defines the supremum of $\|\hat{\mathbf{e}}(i)\|$ in this case. Since we have already assumed that θ_i is less than $\pi/2$ so that $\hat{\mathbf{e}}(i)$ must be inside of the angular sector but its Euclidean norm cannot be smaller than the orthogonal distance between $\hat{\mathbf{a}}_1(0)$ to $\hat{\mathbf{a}}_1(i)$, so this orthogonal distance can define the lower bound of $\|\hat{\mathbf{e}}(i)\|$, which is actually the infimum and calculated by $\|\mathbf{a}_1\|\sin\theta_i$. Then, in the OKSPME method, $\|\hat{\mathbf{e}}(i)\|$ is bounded by

$$\inf\|\hat{\mathbf{e}}(i)\| = \|\mathbf{a}_1\|\sin\theta_i \leq \|\hat{\mathbf{e}}(i)\| < \theta_i\|\mathbf{a}_1\| = \sup\|\hat{\mathbf{e}}(i)\|. \tag{4.95}$$

By taking expectations of both (4.88) and (4.95), we obtain

$$E[\theta_i]\|\mathbf{a}_1\| < \{E[\|\hat{\mathbf{e}}(i)\|]\}_{SQP} \leq \|\mathbf{a}_1\|\tan(E[\theta_i]), \tag{4.96}$$

$$\|\mathbf{a}_1\|\sin(E[\theta_i]) \leq \{E[\|\hat{\mathbf{e}}(i)\|]\}_{OKSPME} < E[\theta_i]\|\mathbf{a}_1\|. \tag{4.97}$$

On the other side, by substituting (4.88) and (4.95) in (4.75), we obtain

$$0 \leq \{\mathrm{Var}[\|\hat{\mathbf{e}}(i)\|]\}_{SQP} \leq \frac{\|\mathbf{a}_1\|^2(\tan\theta_i - \theta_i)^2}{4}, \tag{4.98}$$

$$0 \leq \{\mathrm{Var}[\|\hat{\mathbf{e}}(i)\|]\}_{OKSPME} \leq \frac{\|\mathbf{a}_1\|^2(\theta_i - \sin\theta_i)^2}{4}. \tag{4.99}$$

Substituting (4.96), (4.98) and (4.97), (4.99) in (4.73), respectively, we have

$$E^2[\theta_i]\|\mathbf{a}_1\|^2 < \{E[\|\hat{\mathbf{e}}(i)\|^2]\}_{SQP} \leq \frac{\|\mathbf{a}_1\|^2(\tan\theta_i - \theta_i)^2}{4} + \|\mathbf{a}_1\|^2\tan^2(E[\theta_i]), \tag{4.100}$$

$$\|\mathbf{a}_1\|^2\sin^2(E[\theta_i]) \leq \{E[\|\hat{\mathbf{e}}(i)\|^2]\}_{OKSPME} < \frac{\|\mathbf{a}_1\|^2(\theta_i - \sin\theta_i)^2}{4} + E^2[\theta_i]\|\mathbf{a}_1\|^2. \tag{4.101}$$

However, $E[\theta_i]$ also has its lower and upper bounds. Since our analysis focuses on the unbiased case only as mentioned, the true steering vector is located in the center of the angular sector and the estimate $\hat{\mathbf{a}}_1(i)$ is always closer to the center than $\hat{\mathbf{a}}_1(0)$. Let us assume that even if the estimate $\hat{\mathbf{a}}_1(i)$ always happens to be very close to either edge of the sector, no matter how $\hat{\mathbf{a}}_1(0)$ is chosen within the sector, θ_i will vary from 0 to 2θ with equal probability, or equivalently, uniformly distributed within $[0, 2\theta)$, in which case we can obtain the upper bound for $E[\theta_i]$ by taking the average between 0 to 2θ, which is obtained as θ. On the other hand, if we assume that the estimate $\hat{\mathbf{a}}_1(i)$ always happens to be exactly at the center of the sector, resulting in that θ_i can only vary from 0 to θ, or

uniformly distributed within $[0, \theta]$ in which case $E[\theta_i] = \theta/2$, resulting in the lower bound of $E[\theta_i]$ is $\theta/2$. Therefore, the upper and lower bounds for $\mathrm{MSE}\{\hat{\mathbf{a}}_1(i)\}$ can be further obtained by substituting $E[\theta_i]_{max} \to \theta$, $[\theta_i]_{max} \to 2\theta$ and $E[\theta_i]_{min} = \theta/2$, $[\theta_i]_{min} = 0$ into the upper and lower bounds of (4.100) and (4.101) respectively, resulting in

$$\frac{\theta^2}{4}\|\mathbf{a}_1\|^2 < \{E[\|\hat{\mathbf{e}}(i)\|^2]\}_{SQP} < \frac{\|\mathbf{a}_1\|^2(\tan 2\theta - 2\theta)^2}{4} + \|\mathbf{a}_1\|^2 \tan^2 \theta, \qquad (4.102)$$

$$\|\mathbf{a}_1\|^2 \sin^2 \frac{\theta}{2} \leq \{E[\|\hat{\mathbf{e}}(i)\|^2]\}_{OKSPME} < \frac{\|\mathbf{a}_1\|^2(2\theta - \sin 2\theta)^2}{4} + \theta^2\|\mathbf{a}_1\|^2. \qquad (4.103)$$

Finally, by combining the expectation of the mean-squared initial guess error $E[\|\epsilon\|^2]$ in (4.83) with (4.102) and (4.103), we obtain the bounds for the MSE of the steering vector estimate $\mathrm{MSE}\{\hat{\mathbf{a}}_1(i)\}$ as

$$(2 - \frac{2\sin\theta}{\theta} + \frac{\theta^2}{4})\|\mathbf{a}_1\|^2 < \{\mathrm{MSE}\{\hat{\mathbf{a}}_1(i)\}\}_{SQP}$$
$$< (2 - \frac{2\sin\theta}{\theta} + \frac{(\tan 2\theta - 2\theta)^2}{4} + \tan^2 \theta)\|\mathbf{a}_1\|^2, \quad (4.104)$$

$$(2 - \frac{2\sin\theta}{\theta} + \sin^2 \frac{\theta}{2})\|\mathbf{a}_1\|^2 \leq \{\mathrm{MSE}\{\hat{\mathbf{a}}_1(i)\}\}_{OKSPME}$$
$$< (2 - \frac{2\sin\theta}{\theta} + \frac{(2\theta - \sin 2\theta)^2}{4} + \theta^2)\|\mathbf{a}_1\|^2. \quad (4.105)$$

From (4.104) and (4.105), we can see that the MSEs now only depend on two parameters: the norm of the true steering vector and the angular sector spread. The lower and upper bounds of the proposed OKSPME method are lower than those of the SQP method. As mentioned before, it is important that the presumed angular sector spread 2θ must be less than $\pi/2$ (i.e., 90°) to ensure the previous assumption of $\theta_i < \pi/2$ is always valid.

4.5.2   Complexity Analysis

The computational complexity analysis is discussed in this subsection. We measure the total number of additions and multiplications (i.e., flops) in terms of the number of sensors M performed for each snapshot for the proposed algorithms and the existing ones and list them in Table 4.5 (we assume the noise power is known so does not need to be estimated for ease of comparison). Note that the SQP method in [12] has a highly-variant computational complexity in different snapshots, due to the online optimization program based on

Table 4.5: Complexity Comparison

RAB Algorithms	Flops
LOCSME [31]	$4M^3 + 3M^2 + 20M$
RCB [8]	$2M^3 + 11M^2$
SQP [12]	$\mathcal{O}(M^{3.5})$
LOCME [13]	$2M^3 + 4M^2 + 5M$
LCWC [19]	$2nM^2 + 7nM$
OKSPME	$M^3 + (4m + 11)M^2$ $+(3m^2 + 5m + 20)M$
OKSPME-SG	$(4m + 7)M^2$ $+(3m^2 + 5m + 33)M$
OKSPME-CCG	$(4m + 8n + 8)M^2$ $+(3m^2 + 5m + 33n + 29)M$
OKSPME-MCG	$(4m + 14)M^2$ $+(3m^2 + 5m + 86)M$

random choices of the presumed steering vector. However, it is usually in $\mathcal{O}(M^{3.5})$. The complexity of the LCWC algorithm in [19] often requires a much larger n than that in the proposed LOCSME-CCG algorithm. It is obvious that all of the proposed algorithms have their complexity depending on the Krylov subspace model order m, which is determined from Table 4.1 and is no larger than $K + 1$. For the convenience of comparison, we eliminate all parameters except M by setting them to common values (the values of n in LCWC and OKSPME-CCG is set to 50 and 5 respectively, $m = K + 1$ where $K = 3$) and illustrate their complexity with M varying from 10 to 100 as shown in Fig. 4.4. As can be seen that the proposed OKSPME-SG and OKSPME-MCG algorithms have lower complexity than the other algorithms.

4.6 Simulations

In this section, we present and discuss the simulation results of the proposed RAB algorithms by comparing them to some of the existing RAB techniques. We consider a uniform linear array (ULA) of omnidirectional sensors with half wavelength spacing. To
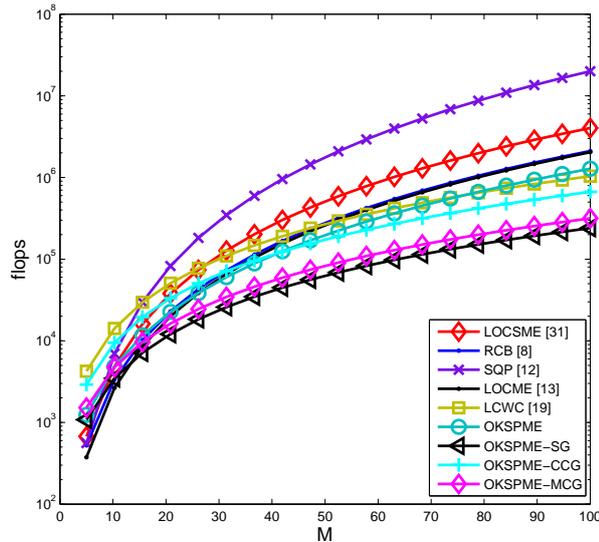
Figure 4.4: Complexity Comparison

produce all the figures (if unspecified in a few scenario), 100 repetitions are executed to
obtain each point of the curves and a maximum of $i = 300$ snapshots are observed. The
desired signal is assumed to arrive at $\theta_1 = 10°$. The signal-to-interference ratio (SIR)
is fixed at 0dB. As the prior knowledge, the angular sector in which the desired signal
is assumed to be located is chosen as $[\theta_1 - 5°, \theta_1 + 5°]$. The results focus on the beam-
former output SINR performance versus the number of snapshots, or a variation of input
SNR (-10dB to 30dB) and both coherent and incoherent local scattering mismatch [9]
scenarios are considered.

4.6.1 Mismatch due to Coherent Local Scattering

All simulations in this subsection consider coherent local scattering. With time-invariant
coherent local scattering, if we choose the number of scatters as 4, the steering vector of
the desired signal is modeled as

$$\mathbf{a}_1 = \mathbf{p} + \sum_{k=1}^{4} e^{j\varphi_k} \mathbf{b}(\theta_k), \tag{4.106}$$

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

91

in each simulation run from a uniform generator with mean $10°$ and standard deviation $2°$. The angles $\varphi_k(k = 1, 2, 3, 4)$ are independently and uniformly taken from the interval $[0, 2\pi]$ in each simulation run. Both θ_k and φ_k change from trials while remaining constant over snapshots.

We firstly compare our proposed methods with some classical RAB methods (i.e., standard diagonal loading method with a fixed loading factor equal to 10 times the noise variance, the RCB method in [8] which estimates the loading factor iteratively, and the method that solves an online quadratic optimization programming, which refers to the SQP method [12]). The numbers of sensors and signal sources (including the desired signal) are set to $M = 10$ and $K = 3$, respectively. For this case only, we set the interferences-to-noise ratio (INR) to 20dB and illustrate the SINR performance versus snapshots within 100 snapshots in Fig. 4.5. The two interferers are arranged to be in the directions of $\theta_2 = 30°$ and $\theta_3 = 50°$, respectively. The other user-defined parameters, if unspecified, (e.g. the step size μ and the forgetting factor λ) are manually optimized to give the best algorithm performance, which is also applied for the other simulation scenarios.
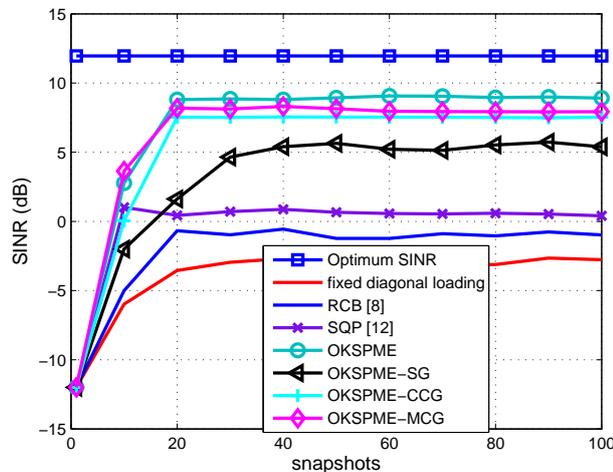

Figure 4.5: Coherent local scattering, SINR versus snapshots, $M = 10$, $K = 3$, INR = 20dB

We then set the number of sensors to $M = 12$, the number of signal sources as (including the desired signal) $K = 3$ and illustrate the SINR versus snapshots and the SINR versus input SNR performance in Fig. 4.6 and Fig. 4.7 respectively. The two inter-

Table 4.6: Changes of Interferers

Snapshots	Number of Interferers $(K-1)$	DoAs
$0-150$	2	$\theta_2 = 30°, \theta_3 = 50°.$
$151-300$	5	$\theta_2 = 20°, \theta_3 = 30°, \theta_4 = 40°,$ $\theta_5 = 50°, \theta_6 = 60°.$

ferers are arranged to be in the directions of $\theta_2 = 30°$ and $\theta_3 = 50°$, respectively. In either Fig. 4.6 or Fig. 4.7, we can see that the proposed OKSPME method has a very similar or slightly better performance compared to the LOCSME algorithm of [31] and both of them have the best performance. Furthermore, the proposed OKSPME-CCG and OKSPME-MCG algorithms also achieve very close performance to OKSPME.

In Fig. 4.8, we assess the SINR performance versus snapshots of those selected algorithms in a specific time-varying scenario which encounters a halfway redistribution of the interferers at a certain snapshot. In this case, the number of sensors is kept at $M = 12$, whereas the details of the interferers are given in Table 4.6.

In Figs. 4.9 and 4.10, we set the number of signal sources to $K = 3$, but increase the number of sensors from $M = 12$ to $M = 40$ and study the SINR versus snapshots and the SINR versus input SNR performance of the selected and proposed dimensionality reduction RAB algorithms, respectively. We set the reduced-dimension as $D = 4$ for the beamspace based algorithm [48] in all simulations. This time, it is clear that the proposed OKSPME, OKSPME-SG, OKSPME-CCG and OKSPME-MCG algorithms all have a certain level of performance degradation compared to the scenario where $M = 12$. The proposed OKSPME based algorithms achieve better performances than the beamspace approach.
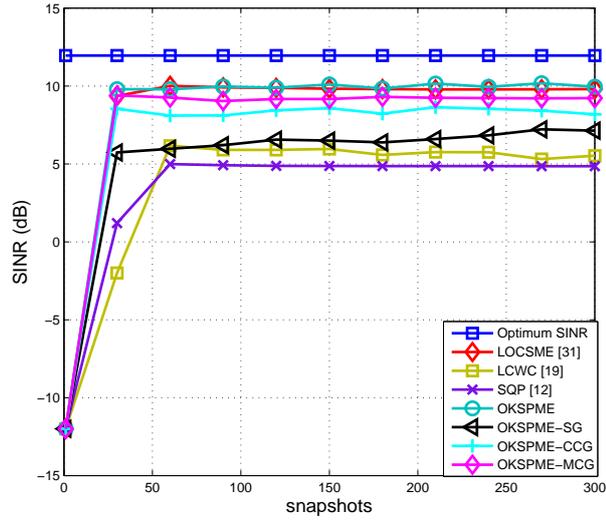
Figure 4.6: Coherent local scattering, SINR versus snapshots, $M = 12$, $K = 3$
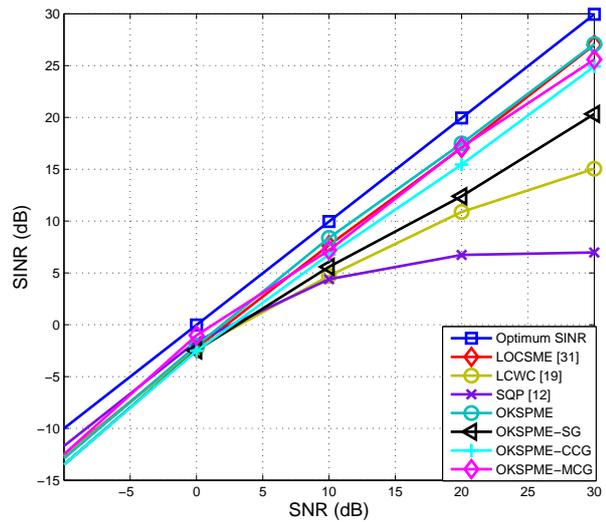
Figure 4.7: Coherent local scattering, SINR versus SNR, $M = 12$, $K = 3$

4.6.2 Mismatch due to Incoherent Local Scattering

In this case, the desired signal affected by incoherent local scattering has a time-varying
signature and its steering vector is modeled by

$$\mathbf{a}_1(i) = s_0(i)\mathbf{p} + \sum_{k=1}^{4} s_k(i)\mathbf{b}(\theta_k), \tag{4.107}$$

94
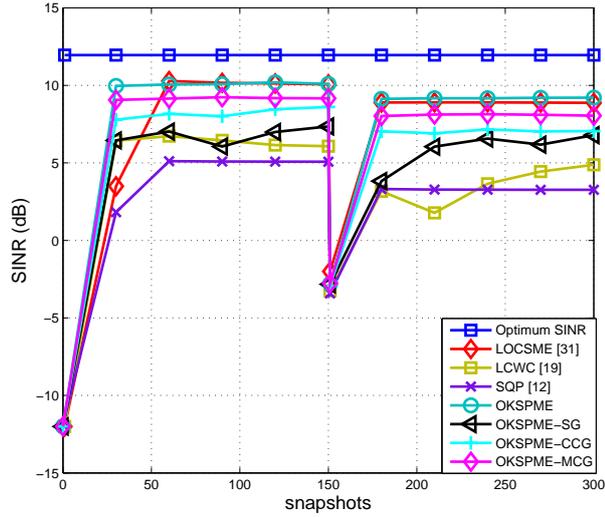
Figure 4.8: Coherent local scattering, SINR versus snapshots, $M = 12$
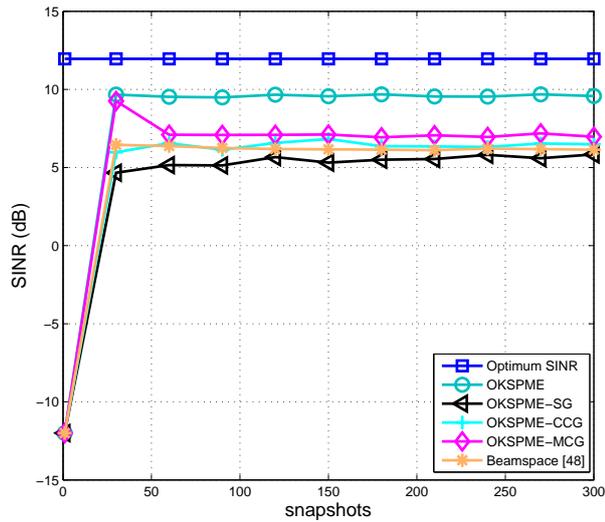
Figure 4.9: Coherent local scattering, SINR versus snapshots, $M = 40$, $K = 3$

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 $10°$ and standard deviation $2°$. At this time, $s_k(i)$ changes both from run to run and from snapshot to snapshot. In order to show the effects caused by incoherent scattering only, we set the parameters $M = 40$ and $K = 3$, study the SINR versus SNR performance of the selected algorithms in Fig. 4.11 and compare the results with Fig. 4.10. As a result, a performance
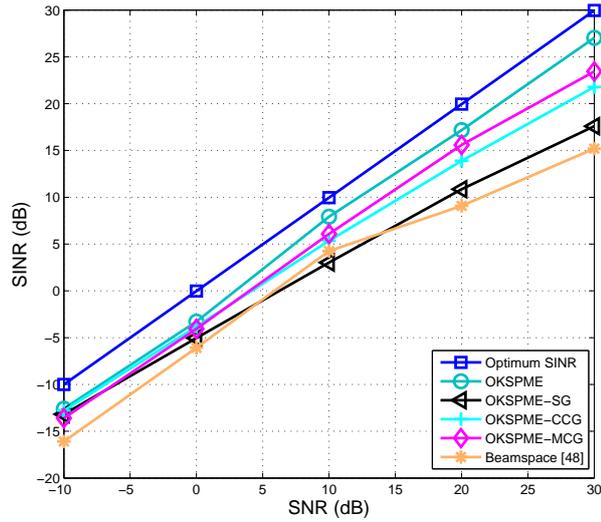
Figure 4.10: Coherent local scattering, SINR versus SNR, $M = 40$, $K = 3$

degradation is observed for all the studied algorithms. This is because the time-varying nature of incoherent scattering results in more dynamic and environmental uncertainties in the system, which increases the steering vector mismatch.
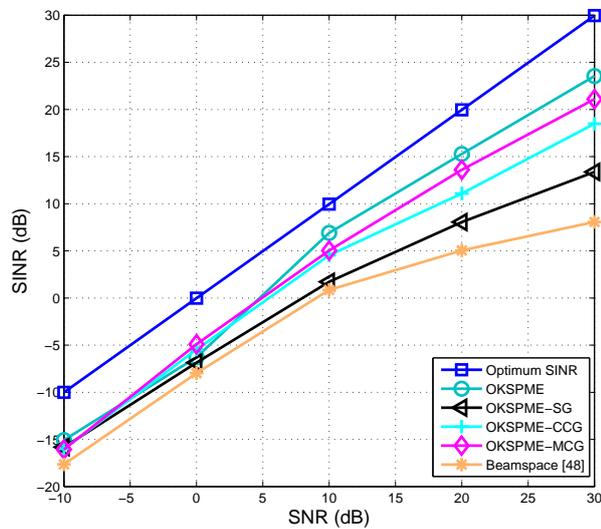
Figure 4.11: Incoherent local scattering, SINR versus SNR, $M = 40$, $K = 3$

4.7   Summary

We have developed the OKSPME algorithm based on the exploitation of cross-correlation mismatch estimation and the orthogonal Krylov subspace. In addition, low complexity RAB algorithms, OKSPME-SG, OKSPME-CCG and OKSPME-MCG have been developed to enable the beamforming weights to be updated recursively without matrix inversions. A detailed steering vector estimation MSE analysis for the general RAB design approach that relies on a presumed angular sector as prior knowledge has been provided. The computational complexity of the proposed and some of the existing algorithms have been compared and discussed. Simulation results have shown that the proposed algorithms have robustness against different choices of user-defined parameters and environmental effects, and achieved excellent output SINR performance especially in medium-high input SNR values.

Chapter 5

Distributed Beamforming and Relay Selection Techniques

Contents

5.1 Introduction

Distributed beamforming has been widely investigated in wireless communications and array processing in recent years [66–68]. It is key for situations in which the channels between the sources and the destination have poor quality so that devices cannot communicate directly and the destination relies on relays that receive and forward the signals [67].

5.1.1 Prior and Related Work

The work in [68] formulates an optimization problem that maximizes the output signal-to-interference-plus-noise ratio (SINR) under individual relay power constraints. The work in [69, 72] focuses on the optimization of weights of all relays to increase the SINR in relay networks. Another related work [80] derives a reference signal based scheme that only uses local channel state information (CSI). The approach in [88] proposes an MMSE consensus cooperative relay networking scheme to exchange data among all the relays under a total power constraint, which limits the total power of all relays regardless of the power allocation. While local communications among the relays are enabled, the ability to mitigate fading effects in wireless channels of the network can be improved [73]. Further earlier works in [89] and [90] explored local communications, while avoiding network centralized processing, which is not desirable and always comes along with the use of total power constraints [88].

However, in most scenarios relays are either not ideally distributed in terms of locations or the channels involved with some of the relays have poor quality. Possible solutions can be categorized in two approaches. One is to adaptively adjust the power of each relay according to the qualities of its associated channels, known as adaptive power control or power allocation. Some power control methods based on channel magnitude and relative analysis has been studied in [70, 91]. An alternative solution is to use relay selection, which selects a number of relays according to a criterion of interest while discarding the remaining relays. In [73, 74, 81], several optimum single-relay selection schemes and a multi-relay selection scheme using relay ordering based on maximizing the output SNR under individual relay power constraints are developed and discussed, but the beamforming weights are not optimized iteratively and synchronously to enhance the SINR maximization. The work in [75, 76] proposed a low-cost greedy search method for the uplink of cooperative direct sequence code-division multiple access systems, which approaches the performance of an exhaustive search. In [82], a combined cooperative beamforming and relay selection scheme that only selects two relays is proposed for physical layer security.

5.1.2 Contributions

In this chapter, we propose a joint MSINR distributed beamforming and restricted greedy search relay selection (RGSRS) algorithm with a total relay transmit power constraint which iteratively optimizes both the beamforming weights at the relay nodes, maximizing the ouput SINR at the destination, provided that the CSI is perfectly known. Specifically, we devise a relay selection scheme based on a greedy search and compare it to other schemes like restricted random relay selection (RRRS) and restricted exhaustive search relay selection (RESRS). The RRRS scheme selects a fixed number of relays randomly from all relays. The RESRS scheme employs the exhaustive search method that runs every single possible combination among all relays aiming to obtain the set with the best SINR performance. The proposed RGSRS scheme is developed from a greedy search method with a specific optimization problem that works in iterations and requires SINR feedback from the destination. The proposed relay selection methods are compared with the scenario without relay selection and the results show significant improvements in terms of SINR and bit-error-rate (BER) performances of the proposed algorithm. The computational cost of all algorithms are analyzed.

This chapter is organized as follows: Section 5.2 presents the system model. Section 5.3 introduces the joint MSINR beamforming and relay selection approach. Section 5.4 derives the joint MSINR and RGSRS algorithm. Section 5.5 presents the simulations and Section 5.8 gives the summary.

5.2 System Model

We consider a wireless communication network consisting of K signal sources (one desired signal with the others as interferers), M distributed single-antenna relays and a destination. It is assumed that the quality of the channels between the signal sources and the destination is very poor so that direct communications is not possible and their links are negligible. The M relays receive information transmitted by the signal sources and then retransmit to the destination as a beamforming procedure, in which a two-step amplify-and-forward (AF) protocol (as shown in Fig. 5.1) is considered as required for cooperative

communications.
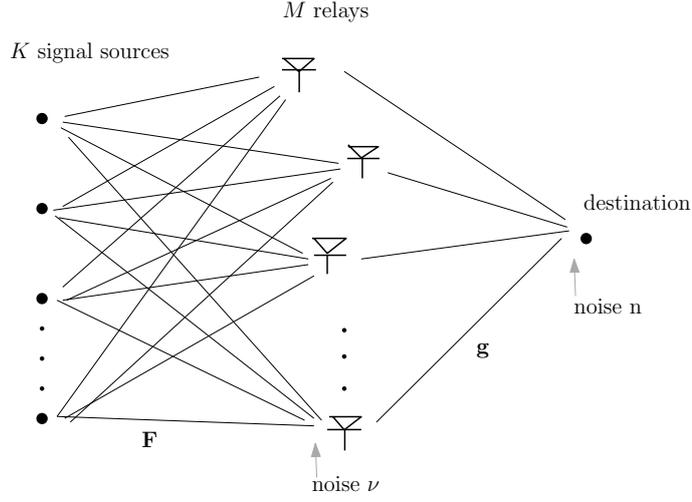
Figure 5.1: System model.

In the first step, the sources transmit the signals to the relays as

$$\mathbf{x} = \mathbf{F}\mathbf{s} + \boldsymbol{\nu}, \tag{5.1}$$

where $\mathbf{s} = [s_1, s_2, \cdots, s_K] \in \mathbb{C}^{1 \times K}$ are signal sources with zero mean, $[.]^T$ denotes the transpose, $s_k = \sqrt{P_k}\xi_k$, $E[|\xi_k|^2] = 1$, P_k is the transmit power of the kth signal source, ξ_k $(k = 1, 2, \cdots, K)$ is the information symbol. Without loss of generality we can assume s_1 as the desired signal while the others are treated as interferers. $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K] \in \mathbb{C}^{M \times K}$ is the channel matrix between the signal sources and the relays, $\mathbf{f}_k = [f_{1,k}, f_{2,k}, \cdots, f_{M,k}]^T \in \mathbb{C}^{M \times 1}$, $f_{m,k}$ denotes the channel between the mth relay and the kth source $(m = 1, 2, \cdots, M, k = 1, 2, \cdots, K)$. $\boldsymbol{\nu} = [\nu_1, \nu_2, \cdots, \nu_M]^T \in \mathbb{C}^{M \times 1}$ is the complex Gaussian noise vector at the relays and σ_ν^2 is the noise variance at each relay $(\nu_m \tilde{} \mathcal{CN}(0, \sigma_\nu^2))$. The vector $\mathbf{x} \in \mathbb{C}^{M \times 1}$ represents the received data at the relays. In the second step, the relays transmit $\mathbf{y} \in \mathbb{C}^{M \times 1}$ which is an amplified and phase-steered version of $\mathbf{x}$, which can be written as

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \tag{5.2}$$

where $\mathbf{W} = \text{diag}[w_1, w_2, \cdots, w_M] \in \mathbb{C}^{M \times M}$ is a diagonal matrix whose diagonal entries denote the beamforming weights. The signal received at the destination is given by

$$z = \mathbf{g}^T \mathbf{y} + n, \tag{5.3}$$

where z is a scalar, $\mathbf{g} = [g_1, g_2, \cdots, g_M]^T \in \mathbb{C}^{M \times 1}$ is the complex Gaussian channel vector between the relays and the destination, n $(n \tilde{} \mathcal{CN}(0, \sigma_n^2), \sigma_n^2 = \sigma_\nu^2)$ is the noise at

the destination and z is the received signal at the destination. Note that we consider both $\mathbf{F}$ and $\mathbf{g}$ are modeled in Rayleigh distribution [77, 78].

5.3 Proposed Joint MSINR Beamforming and Relay Selection

In many cases of relay networking, some of the relays are quite far away from either the signal sources or the destinations, which means they may contribute to degraded network performance due to their poor performance for receiving and transmitting signals. The aim of joint maximum SINR beamforming and relay selection is to compute the beamforming weights according to the maximum SINR criterion and optimize the relay system by discarding the relays with poor performance and making the best use of the relays with good channels in order to improve the overall system performance.

A joint SINR maximization problem with relay selection using a total relay transmit power constraint encountering interferers can be generally described as

$$\mathcal{S}_{\mathrm{opt}} = \arg \max_{\boldsymbol{\alpha}, \mathbf{w}} \; SINR(\mathcal{S}, \mathcal{H}, \mathbf{P}_s, \mathbf{P}_r, P_T, \boldsymbol{\alpha}, \mathbf{w})$$

$$\text{subject to} \sum_{m=1}^{M} \alpha_m^2 P_{r,m} \leq P_T, \tag{5.4}$$

$$\alpha_m \in \{0, 1\}, m = 1, 2, \cdots, M$$

where $\mathcal{S}_{\mathrm{opt}}$ is the optimum relay set of size M_{opt} ($1 \leq M_{\mathrm{opt}} \leq M$) and $SINR$ is a function of $\mathcal{S}, \mathcal{H}, \mathbf{P}_s, \mathbf{P}_r$ and P_T, where $\mathcal{S}$ is the original relay set of size M, $\mathcal{H}$ is the set containing parameters of the CSI (i.e., $\mathcal{H} = \{\mathbf{F}, \mathbf{g}, \sigma_\nu^2\}$), $\mathbf{P}_s = [P_{s,1}, P_{s,2}, \cdots, P_{s,K}] \in \mathbb{R}^{1 \times K}$, $k = 1, 2, \cdots, K$, $\mathbf{P}_r = [P_{r,1}, P_{r,2}, \cdots, P_{r,M}]^T \in \mathbb{R}^{M \times 1}$, $m = 1, 2, \cdots, M$, $P_{r,m}$ refers to the transmit power of the mth relay (Note that before selection we have $\sum_{m=1}^{M} P_{r,m} \leq P_T$ and we consider that each relay cooperates with its full power as long as it is selected), P_T is the maximum allowable total transmit power of all relays, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_M]^T$, α_m ($m = 1, \cdots, M$) is the relay cooperation parameter which determines whether the mth relay cooperates or not. The received signal at the mth relay is:

$$x_m = \sum_{k=1}^{K} \sqrt{P_{s,k}} s f_{m,k} + \nu_m, \tag{5.5}$$

then the transmitted signal at the mth relay can be written as:

$$y_m = \alpha_m w_m x_m. \tag{5.6}$$

Note that we can express the transmit power at the mth relay $P_{r,m}$ as $E[|y_m|^2]$ so that the total relay transmit power can be written as $\sum_{m=1}^{M} E[|y_m|^2] = \sum_{m=1}^{M} E[|\alpha_m w_m x_m|^2]$ or in matrix form as $(\boldsymbol{\alpha}^H \odot \mathbf{w}^H)\mathbf{D}(\boldsymbol{\alpha} \odot \mathbf{w})$ where $\mathbf{D} = \mathrm{diag}(\boldsymbol{\alpha} \odot (\sum_{k=1}^{K} P_{s,k}[E[|f_{1,k}|^2], E[f_{2,k}|^2], \cdots, E[f_{M,k}|^2]]) + \sigma_n^2)$ is a full-rank matrix, where $\odot$ denotes the Schur-Hadamard product which computes element-wise multiplications. The signal received at the destination can be expanded by substituting (5.5) and (5.6) in (5.3), which gives

$$z = \underbrace{\sum_{m=1}^{M} \alpha_m w_m g_m \sqrt{P_{s,1}} f_{m,1} s}_{\text{desired signal}} + \underbrace{\sum_{m=1}^{M} \alpha_m w_m g_m \sum_{k=2}^{K} \sqrt{P_{s,k}} f_{m,k} s}_{\text{interferers}} + \underbrace{\sum_{m=1}^{M} \alpha_m w_m g_m \nu_m + n}_{\text{noise}}. \tag{5.7}$$

By taking expectations of the components of (5.7), we can compute the desired signal power $P_{z,1}$, the interference power $P_{z,i}$ and the noise power $P_{z,n}$ at the destination as follows:

$$P_{z,1} = E[\sum_{m=1}^{M} (\alpha_m w_m g_m \sqrt{P_{s,1}} f_{m,1} s)^2] = P_{s,1} \sum_{m=1}^{M} \alpha_m^2 E[w_m^*(f_{m,1} g_m)(f_{m,1} g_m)^* w_m], \tag{5.8}$$

$$P_{z,i} = E[(\sum_{m=1}^{M} (\alpha_m w_m g_m \sum_{k=2}^{K} \sqrt{P_{s,k}} f_{m,k} s)^2] = \sigma_n^2 (1 + \alpha_m^2 \sum_{m=1}^{M} E[w_m^* g_m g_m^* w_m]), \tag{5.9}$$

$$P_{z,n} = E[\sum_{m=1}^{M} (\alpha_m w_m g_m \nu_m + n)^2] = \sum_{k=2}^{K} P_{s,k} \sum_{m=1}^{M} \alpha_m^2 E[w_m^*(f_{m,k} g_m)(f_{m,k} g_m)^* w_m], \tag{5.10}$$

where $*$ denotes complex conjugation. The SINR is computed as:

$$SINR = \frac{P_{z,1}}{P_{z,i} + P_{z,n}} = \frac{(\boldsymbol{\alpha}^H \odot \mathbf{w}^H)\mathbf{R}_1(\boldsymbol{\alpha} \odot \mathbf{w})}{\sigma_n^2 + (\boldsymbol{\alpha}^H \odot \mathbf{w}^H)(\mathbf{Q} + \sum_{k=2}^{K} \mathbf{R}_k)(\boldsymbol{\alpha} \odot \mathbf{w})}. \tag{5.11}$$

5.3.1 Computation of Weights

By defining $\boldsymbol{\alpha} \odot \mathbf{w} = \tilde{\mathbf{w}}$, the original problem in (5.4) can be cast in terms of solving for $\tilde{\mathbf{w}}$ as:

$$\max_{\tilde{\mathbf{w}}} \quad \frac{\tilde{\mathbf{w}}^H \mathbf{R}_1 \tilde{\mathbf{w}}}{\sigma_n^2 + \tilde{\mathbf{w}}^H (\mathbf{Q} + \sum_{k=2}^{K} \mathbf{R}_k) \tilde{\mathbf{w}}}$$

$$s.t. \quad \tilde{\mathbf{w}}^H \mathbf{D} \tilde{\mathbf{w}} \leq P_T, \tag{5.12}$$

$$\text{Rank}(\tilde{\mathbf{w}} \tilde{\mathbf{w}}^H) = \text{Rank}(\boldsymbol{\alpha} \boldsymbol{\alpha}^H),$$

$$\alpha_m \in \{0, 1\}, m = 1, 2, \cdots, M,$$

where $\mathbf{R}_1$, $\mathbf{Q}$ and $\mathbf{R}_k$ are covariance matrices that are associated with the desired signal, the noise at the relays and the kth interferer and defined by $P_{s,1} E[(\boldsymbol{\alpha} \odot \mathbf{f}_1 \odot \mathbf{g})(\boldsymbol{\alpha} \odot \mathbf{f}_1 \odot \mathbf{g})^H] \in \mathbb{C}^{M \times M}$, $\sigma_n^2 E[(\boldsymbol{\alpha} \odot \mathbf{g})(\boldsymbol{\alpha} \odot \mathbf{g})^H] \in \mathbb{C}^{M \times M}$, $P_{s,k} E[(\boldsymbol{\alpha} \odot \mathbf{f}_k \odot \mathbf{g})(\boldsymbol{\alpha} \odot \mathbf{f}_k \odot \mathbf{g})^H] \in \mathbb{C}^{M \times M}$, respectively. They are such defined so that their ranks are equal to the number of non-zero elements of $\boldsymbol{\alpha}$. The second constraint indicates and ensures $\tilde{\mathbf{w}}$ has the same number of zero elements as $\boldsymbol{\alpha}$ and Rank denotes the rank operator. At this point, we use an alternating optimization strategy to obtain the solutions for both $\mathbf{w}$ and $\boldsymbol{\alpha}$, i.e., we fix the vector $\mathbf{w}$ and optimize $\boldsymbol{\alpha}$ and vice-versa in an alternating fashion. The number of iterations of this alternating optimization depends on both the minimum required number of relays, which is a user defined parameter, and if the maximum SINR is achieved, which is determined by the system feedback. The problem in (5.12) can be solved with respect to $\mathbf{w}$ in a closed-form solution as in the total power constraint SNR maximization problem similarly to [69], with the assumption that the second-order statistics of the CSI (i.e., $\mathcal{H}$) is perfectly known. Then, a closed-form solution for $\tilde{\mathbf{w}}$ is obtained by

$$\tilde{\mathbf{w}} = \sqrt{P_T} \mathbf{D}^{-\frac{1}{2}} \mathcal{P}\{\mathbf{E}\}, \tag{5.13}$$

and the corresponding SINR is

$$SINR = P_T \lambda_{max}\{\mathbf{E}\}, \tag{5.14}$$

where $\mathcal{P}\{.\}$ denotes the principal eigenvector operator, $\lambda_{max}\{.\}$ denotes the largest eigenvalue of the argument, $\mathbf{E} = (\sigma_n^2 \mathbf{I} + P_T \mathbf{D}^{-\frac{1}{2}} (\mathbf{Q} + \sum_{k=2}^{K} \mathbf{R}_k) \mathbf{D}^{-\frac{1}{2}})^{-1} \mathbf{D}^{-\frac{1}{2}} \mathbf{R}_1 \mathbf{D}^{-\frac{1}{2}}$ has the same rank as $\mathbf{R}_1$. It is easy to observe that once we know $\boldsymbol{\alpha}$, we can compute the optimum weights and SINR from (5.13) and (5.14), respectively, by using only the currently selected relay nodes and their weights. The weight optimization steps are detailed in Table. 5.1.

104

5.3.2 Relay Selection

In order to solve the problem in (5.12) with respect to $\boldsymbol{\alpha}$, we consider

$$\max_{\boldsymbol{\alpha}} \ SINR$$

$$\text{subject to} \quad \sum_{m=1}^{M} \alpha_m^2 P_{r,m} \leq P_T, \tag{5.15}$$

$$\alpha_m \in \{0,1\}, m = 1, 2, \cdots, M$$

that can be solved with algorithms like greedy search and exhaustive search, which can be determined by the designer. Note that $\boldsymbol{\alpha}$ is obtained before $\mathbf{w}$ is computed in each recursion. An alternative way that computes $\mathbf{w}$ before obtaining $\boldsymbol{\alpha}$ also works but the above equations will be different. This joint MSINR beamforming and relay selection method requires output SINR comparisons and feedback from the destination to the relay nodes as a form of information exchange, which is similar to [73], but weight optimization is neglected in their work.

5.4 Proposed Joint MSINR and RGSRS Algorithm

The joint MSINR and RGSRS algorithm employs alternating optimization [83–86] iterations. We consider a user-defined parameter M_{min} as a restriction to the minimum number of relays that must be used to allow a higher flexibility for the users to control the number of relays. Before the first iteration all relays are considered (i.e., $\mathcal{S}(0) = \mathcal{S}$). Consequently, we solve the following problem once for each iteration in order to cancel the relay with worst performance from the set $\mathcal{S}(i-1)$ and evaluate $SINR(i)$:

$$\mathcal{S}(i) = \arg \max_{\boldsymbol{\alpha}(i)} \ SINR(i)$$

$$\text{subject to} \sum_{m=1}^{M} \alpha_m^2(i) P_{r,m}(i) \leq P_T,$$

$$\alpha_m(i) \in \{0,1\}, \tag{5.16}$$

$$||\boldsymbol{\alpha}(i)||_1 = M - i,$$

$$||\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}(i-1)||_1 = 1,$$

$$M - i \geq M_{min}, \lrcorner$$

Table 5.1: Beamforming weight vector optimization

1) Choose $\boldsymbol{\alpha}$.

With $\boldsymbol{\alpha}$ and the CSI, compute the following quantities

using the selected relay nodes in each iteration:

The desired signal related covariance matrix :

2) $\mathbf{R}_1 = P_{s,1} E[(\boldsymbol{\alpha} \odot \mathbf{f}_1 \odot \mathbf{g})(\boldsymbol{\alpha} \odot \mathbf{f}_1 \odot \mathbf{g})^H]$

The interferers related covariance matrices for $k = 2, \cdots, K$:

3) $\mathbf{R}_k = P_{s,k} E[(\boldsymbol{\alpha} \odot \mathbf{f}_k \odot \mathbf{g})(\boldsymbol{\alpha} \odot \mathbf{f}_k \odot \mathbf{g})^H] \in \mathbb{C}^{M \times M}$

The noise related covariance matrix:

4) $\mathbf{Q} = \sigma_n^2 E[(\boldsymbol{\alpha} \odot \mathbf{g})(\boldsymbol{\alpha} \odot \mathbf{g})^H]$

The transmit power related full-rank matrix $\mathbf{D}$:

5) $\mathbf{D} = \mathrm{diag}(\boldsymbol{\alpha} \odot (\sum_{k=1}^{K} P_{s,k}$

$[E[|f_{1,k}|^2], E[f_{2,k}|^2], \cdots, E[f_{M,k}|^2]]) + \sigma_n^2)$

The defined matrix $\mathbf{E}$:

6) $\mathbf{E} = (\sigma_n^2 \mathbf{I} + P_T \mathbf{D}^{-\frac{1}{2}} (\mathbf{Q} + \sum_{k=2}^{K} \mathbf{R}_k) \mathbf{D}^{-\frac{1}{2}})^{-1} \mathbf{D}^{-\frac{1}{2}} \mathbf{R}_1 \mathbf{D}^{-\frac{1}{2}}$

Optimize and obtain the beamforming weight vector $\tilde{\mathbf{w}}$:

7) $\tilde{\mathbf{w}} = \sqrt{P_T} \mathbf{D}^{-\frac{1}{2}} \mathcal{P}\{\mathbf{E}\}$

Compute the output SINR at the destination:

8) $SINR = P_T \lambda_{max}\{\mathbf{E}\}$

Table 5.2: Joint MSINR and RGSRS Algorithm

---

step 1: Initialize $\mathcal{S}_{opt} = \mathcal{S}(0)$, $\boldsymbol{\alpha}(0) = \mathbf{1}$ and obtain

$SINR_{opt} = SINR(0)$ using Table. 5.1.

step 2:

for $i = 1, \cdots, M - M_{min}$

    solve the optimization problem (5.16) to obtain $\boldsymbol{\alpha}(i)$,

    $\mathcal{S}(i)$ and compute $SINR(i)$ using Table. 5.1.

    compare $SINR(i)$ to $SINR(i-1)$,

    if $SINR(i) > SINR(i-1)$

        update $\mathcal{S}_{opt} = \mathcal{S}(i)$ and $SINR_{opt} = SINR(i)$.

    else

        keep $\mathcal{S}_{opt} = \mathcal{S}(i-1)$ and $SINR_{opt} = SINR(i-1)$.

        break.

    end if.

end for.

---

where $SINR(i) = SINR(\mathcal{S}(i-1), \mathcal{H}, \mathbf{P}_s, \mathbf{P}_r(i-1), P_T)$ and can be computed by (5.14). If the SINR in the current iteration is higher than that in the previous iteration (i.e. $SINR(i) > SINR(i-1)$), then the selection process continues; if $SINR(i) \leq SINR(i-1)$, we cancel the selection of the current iteration and remain the relay set $\mathcal{S}(i-1)$ and $SINR(i-1)$. The joint MSINR and RGSRS algorithm can be implemented as in Table. 5.2.

At this point, we analyze the computational complexity required by the relay selection algorithms. The MSINR based method for SINR driven beamforming weights optimization has a cost of $\mathcal{O}(M^3)$ since matrix inversions and eigen-decompositions are required. However, M is usually not large so that attentions should be paid to the computational cost caused by the number of iterations required in these relay selection algorithms. As can be seen in table. 5.3, for the joint MSINR and RRRS algorithm, there is no weight vector or relay selection vector optimization required, which means there is only one iteration and the complexity is simply $\mathcal{O}(M^3)$. The joint MSINR and RESRS algorithm has the highest computational cost due to the fact it almost searches for all possible combinations of the relays even though an extra restriction of the minimum number of relays

Table 5.3: Complexity Comparison

Algorithms	Computational Cost
Joint MSINR and RRRS	$\mathcal{O}(M^3)$
Joint MSINR and RESRS	$\sum_{c=M_{min}}^{M} \frac{M!}{(M-c)!c!} \, \mathcal{O}(M^3)$
Joint MSINR and RGSRS	$\leq \frac{(2M-i+1)i}{2} \, \mathcal{O}(M^3)$

required is added in our case. With a restriction of that at least M_{min} relays must be selected, the number of iterations is $\sum_{c=M_{min}}^{M} \frac{M!}{(M-c)!c!}$. In the joint MSINR and RGSRS algorithm, (5.16) is solved once per iteration, which can be done by disabling only one relay while enabling all the others and computing and comparing their output SINRs. The total number of iterations is no greater than $\frac{(2M-i+1)i}{2}$). The proposed joint MSINR and RGSRS algorithm has much lower complexity compared to the joint MSINR and RESRS algorithm when the value of M is large.

5.5   Simulations

For the MSINR relay selection algorithms, we compare the joint MSINR and relay selection algorithms to the scenario without relay selection in terms of their SINR and bit error rate (BER) performances. The parameters used for all scenarios include: number of signal sources $K = 3$, the path loss exponent $\rho = 2$, the power path loss from signals to the destination $L = 10$dB, shadowing spread $\sigma_s = 3$dB, $P_T = 1$dBW. Fig. 5.2-a illustrates the SINR versus SNR (from 0dB to 20dB) performance of the compared algorithms, in which the total number of relays and interference-to-noise ratio (INR) are fixed at $M = 8$ and INR=10dB, respectively. Fig. 5.2-b illustrates how the SINR varies when the total number of relays in the network increases, in which the input SNR=10dB and INR=10dB are fixed. In this case, a minimum total number of relays observed is chosen as $M = 3$, whereas the maximum is at $M = 10$. For each of the above two scenarios, 500 repetitions are carried out for each algorithm. In Fig. 5.3, we evaluate the BER versus SNR performance of all algorithms using binary phase shift keying (BPSK) for the system and test all algorithms with 100000 bits, while keeping INR=10dB. For all the above scenarios, we fix the number of randomly selected relays at 3 for the joint MSINR and random relay selection algorithm, the minimum required selected relays also at 3 for the

other algorithms. As observed, the joint MSINR and RESRS and the joint MSINR and RGSRS algorithms have the best performance.
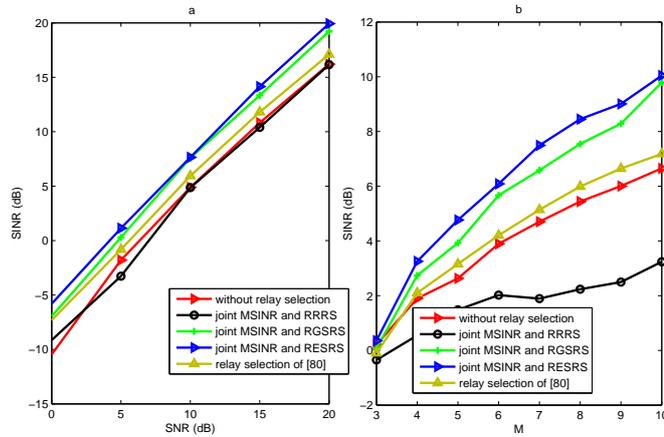
Figure 5.2: SINR versus SNR and M.
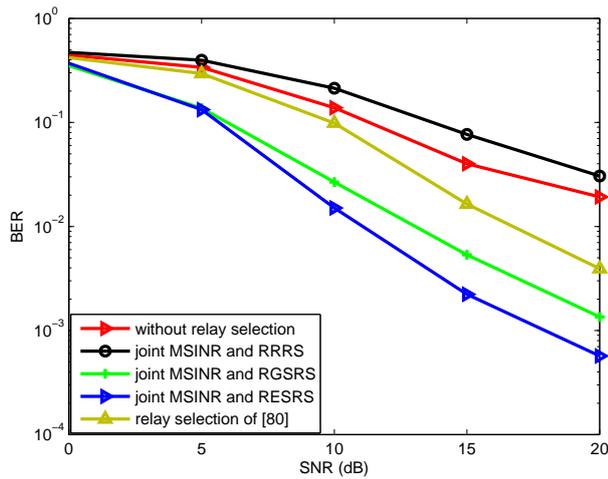
Figure 5.3: BER versus SNR.

5.6 Summary

We have proposed relay selection approaches and developed efficient algorithms for distributed beamforming. We have proposed a joint MSINR and RGSRS algorithm for distributed beamforming which is derived based on a greedy search relay selection scheme.

The computational costs of the proposed algorithms hava been analyzed and compared to prior work that employ RRRS and RESRS schemes. The results have shown excellent SINR and BER performances of the proposed algorithm which are very close to the joint MSINR and RESRS algorithm.

Chapter 6

Robust Distributed Beamforming Techniques

Contents

6.1 Introduction

In most scenarios encountered and even when the relays are ideally distributed in terms of locations, the channels observed by the relays may suffer quality degradation because of inevitable measurement, estimation and quantization errors in CSI [97] as well as propagation effects. These impairments result in imperfect CSI that can affect most distributed beamforming methods, which either fail to work properly or cannot provide satisfactory performance. In this context, robust distributed beamforming (RDB) techniques are hence

in demand to mitigate the channel errors or uncertainties and preserve the relay system performance.

6.1.1 Prior and Related Work

The studies in [94, 96, 97, 108] minimize the total relay transmit power under an overall quality of service (QoS) constraint, using a convex semi-definite programme (SDP) relaxation method. The works in [94, 97] consider the channel errors as Gaussian random vectors with known statistical distributions between the source to the relay nodes and the relay nodes to the destination, whereas [96] models the channel errors on their covariance matrices as a type of matrix perturbation. The work in [98, 103, 104] presents a robust design, which ensures that the SNR constraint is satisfied in the presence of imperfect CSI by adopting a worst-case design and formulates the problem as a convex optimization problem that can be solved efficiently. Similar approaches that use worst-case method can also be found in conventional beamforming as in [7, 109]. The study in [99] discusses multicell coordinated beamforming in the presence of CSI errors, where base stations (BSs) collaboratively mitigate their intercell interference (ICI), in which an optimization problem that minimizes the overall transmission power subject to multiple QoS constraints is considered and solved using semi-definite relaxation (SDR) and the S-Lemma. The work in [107] provides a study of systematic analytical framework for the convergence of a general set of adaptive schemes and their tracking capability with stochastic stability.

6.1.2 Contributions

In this work, unlike most of the existing RDB approaches, we aim to maximize the system output SINR subject to a total relay transmit power constraint using an approach that exploits the cross-correlation between the beamforming weight vector and the system output and then projects the obtained cross-correlation vector onto a subspace computed from the statistics of second-order imperfect channels, namely, the cross-correlation and subspace projection (CCSP) RDB technique. Unlike our previous work on centralized beamform-

ing [100], the CCSP RDB technique is distributed and has marked differences in the way the subspace processing is carried out. In the CCSP RDB method, the covariance matrices of the channel errors are modeled by a certain type of additive matrix perturbation methods [93], which ensures that the covariance matrices are always positive-definite. We consider multiple signal sources and assume that there is no direct link between them and the destination. We also consider that the errors only exist between the signal sources and the relays. The channel error is decomposed and estimated for each signal originating from a source at each time instant separately. The proposed CCSP RDB technique shows outstanding SINR performance as compared to the existing distributed beamforming techniques, which focus on transmit power minimization over a wide range of system input SNR values.

In summary, the main contributions of our work are:

- The proposed CCSP RDB technique.

- A performance analysis of the proposed and existing RDB related techniques.

- A simulation study of the proposed and existing RDB algorithms in several scenarios of interest.

This chapter is organized as follows. Section 6.2 presents the system model. In Section 6.3, the proposed CCSP RDB method is introduced. Section 6.4 present the performance analysis. Section 6.5 presents and discusses the simulation results. Section 6.6 gives the summary.

6.2   System Model

We consider the same relay system model as in Fig. 5.1. The received signal at the mth relay can be expressed as:

$$x_m = \sum_{k=1}^{K} \sqrt{P_{s,k}} s f_{m,k} + \nu_m,$$

(6.1)

then the transmitted signal at the mth relay is given by

$$y_m = w_m x_m. \tag{6.2}$$

The transmit power at the mth relay is equivalent to $E[|y_m|^2]$ so that can be written as $\sum_{m=1}^{M} E[|y_m|^2] = \sum_{m=1}^{M} E[|w_m x_m|^2]$ or in matrix form as $\mathbf{w}^H \mathbf{D} \mathbf{w}$ where $\mathbf{D} = \text{diag}(\sum_{k=1}^{K} P_{s,k}[E[|f_{1,k}|^2], E[f_{2,k}|^2], \cdots, E[f_{M,k}|^2]] + P_n)$ is a full-rank matrix. The signal received at the destination can be expanded by substituting (6.1) and (6.2) in (5.3), which yields

$$z = \underbrace{\sum_{m=1}^{M} w_m g_m \sqrt{P_{s,1}} f_{m,1} s}_{\text{desired signal}} + \underbrace{\sum_{m=1}^{M} w_m g_m \sum_{k=2}^{K} \sqrt{P_{s,k}} f_{m,k} s}_{\text{interferers}} + \underbrace{\sum_{m=1}^{M} w_m g_m \nu_m + n}_{\text{noise}}. \tag{6.3}$$

By taking expectation of the components of (6.3), we can compute the desired signal power $P_{z,1}$, the interference power $P_{z,i}$ and the noise power $P_{z,n}$ at the destination as follows:

$$P_{z,1} = E\Big[\sum_{m=1}^{M} (w_m g_m \sqrt{P_{s,1}} f_{m,1} s)^2 \Big] = P_{s,1} \sum_{m=1}^{M} E\Big[w_m^* (f_{m,1} g_m)(f_{m,1} g_m)^* w_m \Big], \tag{6.4}$$

$$P_{z,i} = E\Big[\sum_{m=1}^{M} (w_m g_m \sum_{k=2}^{K} \sqrt{P_{s,k}} f_{m,k} s)^2 \Big] = \sum_{k=2}^{K} P_{s,k} \sum_{m=1}^{M} E\Big[w_m^* (f_{m,k} g_m)(f_{m,k} g_m)^* w_m \Big], \tag{6.5}$$

$$P_{z,n} = E\Big[\sum_{m=1}^{M} (w_m g_m \nu_m + n)^2 \Big] = P_n(1 + \sum_{m=1}^{M} E\Big[w_m^* g_m g_m^* w_m \Big]), \tag{6.6}$$

where $*$ denotes complex conjugation. By defining

$$\mathbf{R}_k \triangleq P_{s,k} E[(\mathbf{f}_k \odot \mathbf{g})(\mathbf{f}_k \odot \mathbf{g})^H]$$

for $k = 1, 2, \cdots, K$ and

$$\mathbf{Q} \triangleq P_n E[\mathbf{g}\mathbf{g}^H]$$

, the SINR is computed as:

$$SINR = \frac{P_{z,1}}{P_{z,i} + P_{z,n}} = \frac{\mathbf{w}^H \mathbf{R}_1 \mathbf{w}}{P_n + \mathbf{w}^H (\mathbf{Q} + \sum_{k=2}^{K} \mathbf{R}_k) \mathbf{w}}. \tag{6.7}$$

In order to introduce uncertainties or errors $\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_K] \in \mathbb{C}^{M \times K}$ to the channels, we emphasize that only channel $\mathbf{F}$ is considered whereas $\mathbf{g}$ is not affected, in which case we have

$$\hat{\mathbf{f}}_k = \mathbf{f}_k + \mathbf{e}_k, k = 1, 2, \cdots, K, \tag{6.8}$$

where $\hat{\mathbf{f}}_k$ is the kth mismatched channel component of $\mathbf{F}$, $\mathbf{e}_k$ for any $k = 1, \cdots, K$ follows a Gaussian distribution so that its covariance matrix $\mathbf{R}_{\mathbf{e}_k} = \mathbf{e}_k \mathbf{e}_k^H$ is diagonal. In fact, the second-order statistics of any Gaussian distributed vector is diagonal and so as we can directly impose the effects of the uncertainties to all the matrices associated with $\mathbf{f}_k$ in (6.7). By using an additive Frobenius norm matrix perturbation method as introduced in [93], thus we can have the following

$$\hat{\mathbf{R}}_k = \mathbf{R}_k + \mathbf{R}_{\mathbf{e}_k} = \mathbf{R}_k + \epsilon ||\mathbf{R}_k||_F \mathbf{I}_M, k = 1, \cdots, K, \qquad (6.9)$$

$$\hat{\mathbf{D}} = \mathbf{D} + \epsilon ||\mathbf{D}||_F \mathbf{I}_M, \qquad (6.10)$$

where $\hat{\mathbf{R}}_k$ and $\hat{\mathbf{D}}$ are the matrices perturbed after the channel mismatch effects are taken into account, ϵ is the perturbation parameter uniformly distributed within $(0, \epsilon_{max}]$ where ϵ_{max} is a predefined constant which describes the mismatch level. The matrix $\mathbf{I}_M$ represents the identity matrix of dimension M and it is clear that $\hat{\mathbf{R}}_k$ and $\hat{\mathbf{D}}$ are positive definite, i.e. $\hat{\mathbf{R}}_k \succ \mathbf{0} (k = 1, \cdots, K)$ and $\hat{\mathbf{D}} \succ \mathbf{0}$. At this point, according to (6.7), the robust optimization problem that aims to maximize the output SINR with a total relay transmit power constraint can be written as

$$\max_{\mathbf{w}} \quad \frac{\mathbf{w}^H \hat{\mathbf{R}}_1 \mathbf{w}}{P_n + \mathbf{w}^H (\mathbf{Q} + \sum_{k=2}^{K} \hat{\mathbf{R}}_k) \mathbf{w}} \qquad (6.11)$$
$$\text{subject to} \quad \mathbf{w}^H \hat{\mathbf{D}} \mathbf{w} \leq P_T.$$

The optimization problem (6.11) shares a similar form to the optimization problem discussed in [69] and hence can be solved in a closed form using an eigen-decomposition method that only requires quantities or parameters with known second-order statistics.

6.3 Proposed CCSP RDB Algorithm

In this section, the proposed CCSP RDB algorithm is introduced. The algorithm works iteratively and is based on the exploitation of cross-correction vector between the relay received data and the system output, as well as the construction of an eigen-subspace. By projecting the so obtained cross-correlation vector onto the subspace, the channel errors can be efficiently eliminated at its best and the result leads to an precise estimate of the

mismatched channels. To do this, the iteration index i is introduced and the sample cross-correlation vector (SCV) $\hat{\mathbf{q}}(i)$ in the ith iteration can be estimated by

$$\hat{\mathbf{q}}(i) = \frac{1}{i} \sum_{j=1}^{i} \mathbf{x}(j) z^*(j). \tag{6.12}$$

Then, we break down the mismatched channel matrix $\hat{\mathbf{F}}(i)$ to K components as $\hat{\mathbf{F}}(i) = [\hat{\mathbf{f}}_1(i), \hat{\mathbf{f}}_2(i), \cdots, \hat{\mathbf{f}}_K(i)]$ and for each of them we construct a separate projection matrix. For the kth ($1 \leq k \leq K$) component, we firstly compute the covariance matrix for $\hat{\mathbf{f}}_k(i)$ directly and take it as an estimate of the true channel covariance matrix instead of the mismatched channel covariance matrix as

$$\hat{\mathbf{R}}_{\hat{\mathbf{f}}_k}(i) = \frac{1}{i} \sum_{j=1}^{i} \hat{\mathbf{f}}_k(j) \hat{\mathbf{f}}_k^H(j). \tag{6.13}$$

Here we take an approximation for the time-averaged estimate of the covariance matrix so that we have $\frac{1}{i} \sum_{j=1}^{i} \mathbf{f}_k(j) \mathbf{f}_k^H(j) \approx \frac{1}{i} \sum_{j=1}^{i} \hat{\mathbf{f}}_k(j) \hat{\mathbf{f}}_k^H(j)$ and $\mathbf{R}_{\mathbf{f}_k}(i) = \frac{1}{i} \sum_{j=1}^{i} \hat{\mathbf{f}}_k(j) \hat{\mathbf{f}}_k^H(j)$. Then the error covariance matrix $\mathbf{R}_{\mathbf{e}_k}(i)$ can be computed as

$$\mathbf{R}_{\mathbf{e}_k}(i) = \epsilon ||\mathbf{R}_{\mathbf{f}_k}(i)||_F \mathbf{I}_M. \tag{6.14}$$

In order to eliminate or reduce the errors $\mathbf{e}_k(i)$ from $\hat{\mathbf{f}}_k(i)$, the SCV obtained in (6.12) can be projected onto a subspace given by

$$\mathbf{P}_k(i) = [\mathbf{c}_{1,k}(i), \mathbf{c}_{2,k}(i), \cdots, \mathbf{c}_{N,k}(i)][\mathbf{c}_{1,k}(i), \mathbf{c}_{2,k}(i), \cdots, \mathbf{c}_{N,k}(i)]^H, \tag{6.15}$$

where $\mathbf{c}_{1,k}(i), \mathbf{c}_{2,k}(i), \cdots, \mathbf{c}_{N,k}(i)$ are the N principal eigenvectors of the error spectrum matrix $\mathbf{C}_k(i)$, which is defined by

$$\mathbf{C}_k(i) \triangleq \int_{\epsilon \to 0^+}^{\epsilon_{max}} E[\hat{\mathbf{f}}_k(i) \hat{\mathbf{f}}_k^H(i)] d\epsilon = \int_{\epsilon \to 0^+}^{\epsilon_{max}} E[(\mathbf{f}_k(i) + \mathbf{e}_k(i))(\mathbf{f}_k(i) + \mathbf{e}_k(i))^H] d\epsilon. \tag{6.16}$$

Assuming $\mathbf{e}_k(i)$ is uncorrelated with $\mathbf{f}_k(i)$ and ϵ follows a uniform distribution over the sector $(0, \epsilon_{max}]$ and approximating $\mathbf{f}_k(i) \mathbf{f}_k^H(i) \approx \mathbf{R}_{\mathbf{f}_k}(i)$ and $\mathbf{e}_k(i) \mathbf{e}_k^H(i) \approx \mathbf{R}_{\mathbf{e}_k}(i)$, then (6.16) can be simplified as

$$\mathbf{C}_k(i) = \int_{\epsilon \to 0^+}^{\epsilon_{max}} (\mathbf{R}_{\mathbf{f}_k}(i) + \mathbf{R}_{\mathbf{e}_k}(i)) d\epsilon = \epsilon_{max} \mathbf{R}_{\mathbf{f}_k}(i) + \frac{\epsilon_{max}^2}{2} ||\mathbf{R}_{\mathbf{f}_k}(i)||_F \mathbf{I}_M. \tag{6.17}$$

Then the mismatched channel component is then estimated by

$$\hat{\mathbf{f}}_k(i) = \frac{\mathbf{P}_k(i) \hat{\mathbf{q}}(i)}{||\mathbf{P}_k(i) \hat{\mathbf{q}}(i)||_2}. \tag{6.18}$$

To this point, all the K channel components of $\hat{\mathbf{f}}_k(i)$ are obtained so that we have $\hat{\mathbf{F}}_k(i) = [\hat{\mathbf{f}}_1(i), \hat{\mathbf{f}}_2(i), \cdots, \hat{\mathbf{f}}_K(i)]$. In the next step, we use the so obtained channel components to provide estimates for the matrix quantities $\hat{\mathbf{R}}_k(i)$ $(k = 1, \cdots, K)$ and $\hat{\mathbf{D}}(i)$ in (6.11) as follows:

$$\hat{\mathbf{R}}_k(i) = P_{s,k} E[(\hat{\mathbf{f}}_k(i) \odot \mathbf{g}(i))(\hat{\mathbf{f}}_k(i) \odot \mathbf{g}(i))^H], \tag{6.19}$$

$$\hat{\mathbf{D}}(i) = \mathrm{diag}\Big(\sum_{k=1}^{K} P_{s,k}[E[|\hat{f}_{1,k}(i)|^2], \cdots, E[|\hat{f}_{M,k}(i)|^2]] + P_n \Big). \tag{6.20}$$

To proceed further, we define $\hat{\mathbf{U}}(i) = \mathbf{Q}(i) + \sum_{k=2}^{K} \hat{\mathbf{R}}_k(i)$ (where $\mathbf{Q}(i) = P_n E[\mathbf{g}(i)\mathbf{g}^H(i)]$), so that (6.11) can be written as

$$\max_{\mathbf{w}(i)} \frac{\mathbf{w}^H(i)\hat{\mathbf{R}}_1(i)\mathbf{w}(i)}{P_n + \mathbf{w}^H(i)\hat{\mathbf{U}}(i)\mathbf{w}(i)} \tag{6.21}$$
$$\text{subject to} \quad \mathbf{w}^H(i)\hat{\mathbf{D}}(i)\mathbf{w}(i) \leq P_T.$$

To solve the optimization problem in (6.21), the weight vector is rewritten as

$$\mathbf{w}(i) = \sqrt{p}\mathbf{D}^{-1/2}(i)\tilde{\mathbf{w}}(i), \tag{6.22}$$

where $\tilde{\mathbf{w}}(i)$ satisfies $\tilde{\mathbf{w}}^H(i)\tilde{\mathbf{w}}(i) = 1$. Then (6.21) can be rewritten as

$$\max_{p, \tilde{\mathbf{w}}(i)} \frac{p\tilde{\mathbf{w}}^H(i)\tilde{\mathbf{R}}_1(i)\tilde{\mathbf{w}}(i)}{p\tilde{\mathbf{w}}^H(i)\tilde{\mathbf{U}}(i)\tilde{\mathbf{w}}(i) + P_n} \tag{6.23}$$
$$\text{subject to} \quad ||\tilde{\mathbf{w}}(i)||^2 = 1, p \leq P_T,$$

where $\tilde{\mathbf{R}}_1(i) = \hat{\mathbf{D}}^{-1/2}(i)\hat{\mathbf{R}}_1(i)\mathbf{D}^{-1/2}(i)$ and $\tilde{\mathbf{U}}(i) = \hat{\mathbf{D}}^{-1/2}(i)\hat{\mathbf{U}}(i)\hat{\mathbf{D}}^{-1/2}(i)$. As the objective function in (6.23) increases monotonically with p regardless of $\tilde{\mathbf{w}}(i)$, which means the objective function is maximized when $p = P_T$, hence (6.23) can be simplified to

$$\max_{\tilde{\mathbf{w}}(i)} \frac{P_T\tilde{\mathbf{w}}^H(i)\tilde{\mathbf{R}}_1(i)\tilde{\mathbf{w}}(i)}{P_T\tilde{\mathbf{w}}^H(i)\tilde{\mathbf{U}}(i)\tilde{\mathbf{w}}(i) + P_n} \tag{6.24}$$
$$\text{subject to} \quad ||\tilde{\mathbf{w}}(i)||^2 = 1,$$

or equivalently as

$$\max_{\tilde{\mathbf{w}}(i)} \frac{P_T\tilde{\mathbf{w}}^H(i)\tilde{\mathbf{R}}_1(i)\tilde{\mathbf{w}}(i)}{\tilde{\mathbf{w}}^H(i)(P_n\mathbf{I}_M + P_T\tilde{\mathbf{U}}(i))\tilde{\mathbf{w}}(i)} \tag{6.25}$$
$$\text{subject to} \quad ||\tilde{\mathbf{w}}(i)||^2 = 1,$$

in which the objective function is maximized when $\tilde{\mathbf{w}}(i)$ is chosen as the principal eigenvector of $(P_n\mathbf{I}_M + P_T\tilde{\mathbf{U}}(i))^{-1}\tilde{\mathbf{R}}_1(i)$ [69], which leads to the solution for the weight vector

of the distributed beamformer with channel errors given by

$$\mathbf{w}(i) = \sqrt{P_T}\hat{\mathbf{D}}^{-1/2}(i)\mathcal{P}\{(P_n\mathbf{I}_M + \hat{\mathbf{D}}^{-1/2}(i)\hat{\mathbf{U}}(i)\hat{\mathbf{D}}^{-1/2}(i))^{-1}\hat{\mathbf{D}}^{-1/2}(i)\hat{\mathbf{R}}_1(i)\hat{\mathbf{D}}^{-1/2}(i)\},$$

(6.26)

where $\mathcal{P}\{.\}$ denotes the principal eigenvector corresponding to the largest eigenvalue. Then the maximum achievable SINR of the system in the presence of channel errors is given by

$$\text{SINR}_{max} = P_T\lambda_{max}\{(P_n\mathbf{I}_M + \hat{\mathbf{D}}^{-1/2}(i)\hat{\mathbf{U}}(i)\hat{\mathbf{D}}^{-1/2}(i))^{-1}\hat{\mathbf{D}}^{-1/2}(i)\hat{\mathbf{R}}_1(i)\hat{\mathbf{D}}^{-1/2}(i)\},$$

(6.27)

where λ_{max} is the maximum eigenvalue. The steps of the proposed CCSP RDB algorithm are detailed in Table 6.1.

6.4 Analysis

This section presents a performance analysis of the proposed CCSP RDB algorithm in terms of the mean square error (MSE) for the channels. In the MSE analysis, we make assumptions that the channel components $\mathbf{f}_k$, $k = 1, \cdots, K$, the error vectors $\mathbf{e}_k$, $k = 1, \cdots, K$, and the noise ν, n are all uncorrelated with each other. We then investigate the MSE using two different approaches, one obtains a general pair of upper and lower bounds that are based on the spread of the channel covariance matrix for the channel error model adopted, whereas the other approach focuses on the procedure of subspace projection that involves the SCV and leads to a problem related to the study of principal component analysis (PCA), which has been a popular research topic in computer science, statistics and theoretical mathematics.

6.4.1 MSE Analysis

In this section, we carry out a general MSE analysis of the channel errors associated with the distributed beamforming problem. In particular, we firstly aim to obtain a pair of upper and lower bounds for all methods that model the channel error covariance matrix as an addictive perturbation based on the Frobenius norm of the true channel covariance

Table 6.1: Proposed CCSP RDB Algorithm

Initialization:

$\mathbf{w}(0) = \mathbf{1}$; $\hat{\mathbf{q}}(0) = \mathbf{1}$; $\hat{\mathbf{R}}_k(0) = \mathbf{I}_M$ for $k = 1, \cdots, K$; ϵ_{max}; N; P_T. For iteration $i = 1, 2, \cdots$:

Compute the SCV as:

$\quad \hat{\mathbf{q}}(i) = ((i-1) \cdot \hat{\mathbf{q}}(i-1) + \mathbf{x}(i) z^*(i))/i$

For $k = 1, \cdots, K$:

$\quad$ Approximate the covariance matrix for the kth channel component as:

$\quad \mathbf{R}_{\mathbf{f}_k}(i) \approx ((i-1) \cdot \hat{\mathbf{R}}_{\hat{\mathbf{f}}_k}(i-1) + \hat{\mathbf{f}}_k(i) \hat{\mathbf{f}}_k^H(i))/i$

$\quad$ Compute the error spectrum matrix for $\hat{\mathbf{f}}_k(i)$:

$\quad \mathbf{C}_k(i) = \epsilon_{max} \mathbf{R}_{\mathbf{f}_k}(i) + \frac{\epsilon_{max}^2}{2} ||\mathbf{R}_{\mathbf{f}_k}(i)||_F \mathbf{I}_M$

$\quad$ Compute N principal eigenvectors of $\mathbf{C}$

$\quad$ and obtain $[\mathbf{c}_{1,k}, \mathbf{c}_{2,k}, \cdots, \mathbf{c}_{N,k}]$

$\quad$ Compute the projection matrix for $\hat{\mathbf{f}}_k(i)$:

$\quad \mathbf{P}_k(i) = [\mathbf{c}_{1,k}, \mathbf{c}_{2,k}, \cdots, \mathbf{c}_{N,k}][\mathbf{c}_{1,k}, \mathbf{c}_{2,k}, \cdots, \mathbf{c}_{N,k}]^H$

$\quad$ Estimate $\hat{\mathbf{f}}_k(i)$ by subspace projection:

$\quad \hat{\mathbf{f}}_k(i) = \frac{\mathbf{P}_k(i) \hat{\mathbf{q}}(i)}{||\mathbf{P}_k(i) \hat{\mathbf{q}}(i)||_2}$

$\quad$ Compute $\hat{\mathbf{R}}_k(i)$:

$\quad \hat{\mathbf{R}}_k(i) = P_{s,k} E[(\hat{\mathbf{f}}_k(i) \odot \mathbf{g}(i))(\hat{\mathbf{f}}_k(i) \odot \mathbf{g}(i))^H]$

End of k.

Compute quantities $\hat{\mathbf{D}}(i)$, $\mathbf{Q}(i)$ and $\hat{\mathbf{U}}(i)$:

$\hat{\mathbf{D}}(i) = \mathrm{diag}(\sum_{k=1}^K P_{s,k}[E[|\hat{f}_{1,k}(i)|^2], E[|\hat{f}_{2,k}(i)|^2], \cdots, E[\hat{f}_{M,k}(i)|^2]] + P_n)$

$\mathbf{Q}(i) = P_n E[\mathbf{g}(i) \mathbf{g}^H(i)]$

$\hat{\mathbf{U}}(i) = \mathbf{Q}(i) + \sum_{k=2}^K \hat{\mathbf{R}}_k(i)$

Obtain the beamformer weight vector:

$\mathbf{w}(i) = \sqrt{P_T} \hat{\mathbf{D}}^{-1/2}(i) \mathcal{P}\{(P_n \mathbf{I}_M + \hat{\mathbf{D}}^{-1/2}(i) \hat{\mathbf{U}}(i) \hat{\mathbf{D}}^{-1/2}(i))^{-1} \hat{\mathbf{D}}^{-1/2}(i) \hat{\mathbf{R}}_1(i) \hat{\mathbf{D}}^{-1/2}(i)\}$

Compute the system output SINR:

$\mathrm{SINR}_{max} = P_T \lambda_{max}\{(P_n \mathbf{I}_M + \hat{\mathbf{D}}^{-1/2}(i) \hat{\mathbf{U}}(i) \hat{\mathbf{D}}^{-1/2}(i))^{-1} \hat{\mathbf{D}}^{-1/2}(i) \hat{\mathbf{R}}_1(i) \hat{\mathbf{D}}^{-1/2}(i)\}$

End of i.

---

matrix and a matrix perturbation parameter ϵ as in (6.14). According to the definition, the MSE of $\hat{\mathbf{f}}_k$ is given by

$$\text{MSE}\{\hat{\mathbf{f}}_k\}_1 \triangleq \text{tr}(E[(\hat{\mathbf{f}}_k - \mathbf{f}_k)(\hat{\mathbf{f}}_k - \mathbf{f}_k)^H])$$

$$= \text{tr}(E[\mathbf{e}_k\mathbf{e}_k^H]) = \text{tr}(E[\mathbf{R}_{\mathbf{e}_k}]) = \text{tr}(\frac{\epsilon_{max}}{2}E[||\mathbf{R}_{\mathbf{f}_k}||_F]\mathbf{I}_M) = \frac{\epsilon_{max}M}{2}E[||\mathbf{R}_{\mathbf{f}_k}||_F]. \quad (6.28)$$

Furthermore, the Frobenius norm of any positive definite matrix can be expressed as the square root of the sum of its squared eigenvalues, which results in

$$||\mathbf{R}_{\mathbf{f}_k}||_F = \sqrt{\sum_{m=1}^{M} \lambda_{m,k}^2}, \quad (6.29)$$

where $\lambda_{m,k}$ refers to the mth eigenvalue of matrix $\mathbf{R}_{\mathbf{f}_k}$.

Let us now denote the eigenvalue spread of the matrix $\mathbf{R}_{\mathbf{f}_k}$ as $\sigma_{\lambda,k}$, which is defined by $|\lambda_{max,k} - \lambda_{min,k}|$, where $\lambda_{max,k}$ and $\lambda_{max,k}$ refer to the maximum eigenvalue and the minimum eigenvalue of $\mathbf{R}_{\mathbf{f}_k}$, respectively. Then we can obtain a lower bound for $\min\{\sum_{m=1}^{M} \lambda_{m,k}^2\}$, by assuming $\lambda_{1,k}, \lambda_{2,k}, \cdots, \lambda_{m,k}, \cdots, \lambda_{M,k}$ ($\lambda_{m,k} \neq \lambda_{max,k}$) $\rightarrow 0^+$, which yields the following relations for the lower bound on the MSE of $\hat{\mathbf{f}}_k$:

$$\min\{\sum_{m=1}^{M} \lambda_{m,k}^2\} > (M-1)\lambda_{min,k}^2 + \lambda_{max,k}^2$$

$$= (M-1)(\lambda_{max,k} - \sigma_{\lambda,k})^2 + \lambda_{max,k}^2 = M\lambda_{max,k}^2 - 2(M-1)\sigma_{\lambda,k}\lambda_{max,k} + (M-1)\sigma_{\lambda,k}^2. \quad (6.30)$$

On the other hand, we can also obtain an upper bound for $\max\{\sum_{m=1}^{M} \lambda_{m,k}^2\}$, by assuming $\lambda_{1,k}, \lambda_{2,k}, \cdots, \lambda_{m,k}, \cdots, \lambda_{M,k}$ ($\lambda_{m,k} \neq \lambda_{max,k}$) $\rightarrow \lambda_{max,k}^-$, which yields the following relations for the upper bound on the MSE of $\hat{\mathbf{f}}_k$:

$$\max\{\sum_{m=1}^{M} \lambda_{m,k}^2\} < (M-1)\lambda_{max,k}^2 + \lambda_{min,k}^2$$

$$= (M-1)\lambda_{max,k}^2 + (\lambda_{max,k} - \sigma_{\lambda,k})^2 = M\lambda_{max,k}^2 - 2\sigma_{\lambda,k}\lambda_{max,k} + \sigma_{\lambda,k}^2. \quad (6.31)$$

By substituting (6.29) to (6.30) and (6.31), we have

$$\min\{||\mathbf{R}_{\mathbf{f}_k}||_F\} > \sqrt{M\lambda_{max,k}^2 - 2(M-1)\sigma_{\lambda,k}\lambda_{max,k} + (M-1)\sigma_{\lambda,k}^2}, \quad (6.32)$$

$$\max\{||\mathbf{R}_{\mathbf{f}_k}||_F\} < \sqrt{M\lambda_{max,k}^2 - 2\sigma_{\lambda,k}\lambda_{max,k} + \sigma_{\lambda,k}^2}. \quad (6.33)$$

Since we have $\min\{||\mathbf{R_{f_k}}||_F\} \leq E[||\mathbf{R_{f_k}}||_F] \leq \max\{||\mathbf{R_{f_k}}||_F\}$, then we can obtain the upper and lower bounds for $E[||\mathbf{R_{f_k}}||_F]$ by substituting the relations in (6.30) and (6.31) in (6.32) and (6.33), resulting in

$$\sqrt{M\lambda_{max,k}^2 - 2(M-1)\sigma_{\lambda,k}\lambda_{max,k} + (M-1)\sigma_{\lambda,k}^2} < E[||\mathbf{R_{f_k}}||_F] <$$
$$\sqrt{M\lambda_{max,k}^2 - 2\sigma_{\lambda,k}\lambda_{max,k} + \sigma_{\lambda,k}^2}, \quad (6.34)$$

which is then substituted in (6.28) and yields the bounds for the MSE:

$$\frac{\epsilon_{max}M}{2}\sqrt{M\lambda_{max,k}^2 - 2(M-1)\sigma_{\lambda,k}\lambda_{max,k} + (M-1)\sigma_{\lambda,k}^2} < \text{MSE}\{\hat{\mathbf{f}}_k\}_1 <$$
$$\frac{\epsilon_{max}M}{2}\sqrt{M\lambda_{max,k}^2 - 2\sigma_{\lambda,k}\lambda_{max,k} + \sigma_{\lambda,k}^2}. \quad (6.35)$$

The bounds described in (6.35) give a basic idea about how the Frobenius norm of the kth component channel matrix $\mathbf{R_{f_k}}$ is constrained with respect to its maximum (principal) eigenvalue $\lambda_{max,k}$ and the eigenvalue spread $\sigma_{\lambda,k}$ of $||\mathbf{R_{f_k}}||_F$. If we take the lower bounds as the Minimum MSE (MMSE) of the channel components and compute the overall system MMSE, then we can obtain the system MMSE with respect to $\hat{\mathbf{F}}$ as following:

$$\text{MMSE}\{\hat{\mathbf{F}}\} = \sum_{k=1}^{K}\text{MMSE}\{\hat{\mathbf{f}}_k\}$$
$$= \frac{\epsilon_{max}M}{2}\sum_{k=1}^{K}\sqrt{M\lambda_{max,k}^2 - 2(M-1)\sigma_{\lambda,k}\lambda_{max,k} + (M-1)\sigma_{\lambda,k}^2}. \quad (6.36)$$

At this point, we can compute the MMSE for the system output by directly substituting (6.36) in (5.1), (5.2), (5.3), and obtain

$$\text{MMSE}\{z\} = \mathbf{g}^T\mathbf{W}(\hat{\mathbf{F}}\mathbf{s} + \boldsymbol{\nu}) + n. \quad (6.37)$$

Here we simply replace $\hat{\mathbf{F}}$ by a diagonal matrix $\text{MMSE}\{\hat{\mathbf{F}}\}\mathbf{I}_M$ which is characterized by its MMSE, and we obtain

$$\text{MMSE}\{z\} = \mathbf{g}^T\mathbf{W}(\frac{\epsilon_{max}M}{2}\sum_{k=1}^{K}(M\lambda_{max,k}^2 - 2(M-1)\sigma_{\lambda,k}\lambda_{max,k} + (M-1)\sigma_{\lambda,k}^2)^{-1/2}\mathbf{s} + \boldsymbol{\nu}) + n. \quad (6.38)$$

Furthermore, from [102] we know that the MMSE of the system SINR can be associated with its actual value, provided that the channels and input data have Gaussian distribution, which is given by

$$\text{SINR} = \frac{1}{\text{MMSE}\{z\}} - 1. \quad (6.39)$$

By substituting (6.38) in (6.39) we obtain

$$\text{SINR} = 1/\Big(\mathbf{g}^T\mathbf{W}\big(\frac{\epsilon_{max}M}{2}\sum_{k=1}^{K}(M\lambda_{max,k}^2$$

$$- 2(M-1)\sigma_{\lambda,k}\lambda_{max,k} + (M-1)\sigma_{\lambda,k}^2)^{-1/2}\mathbf{s} + \boldsymbol{\nu}\big) + n\Big) - 1. \quad (6.40)$$

Equation (6.40) presents a relation between the system output SINR and the maximum eigenvalue $\lambda_{max,k}$ of matrix $\mathbf{R}_{\mathbf{f}_k}$ and its spread $\sigma_{\lambda,k}$. In order to study how the MSE bounds vary with the values of the matrix spread and the maximum eigenvalue, we set total number of relays and signal sources $M = 8$, $\epsilon_{max} = 0.2$. Then we test two cases with $\sigma_{\lambda,k} = 0.9\lambda_{max,k}$ and $\sigma_{\lambda,k} = 0.5\lambda_{max,k}$ and illustrate the variations of those bounds in Fig. 6.1 and Fig. 6.2, respectively. Because we use linear relations between $\sigma_{\lambda,k}$ and $\lambda_{max,k}$, proportional relations between the MSE bounds and $\lambda_{max,k}$ are reflected as can be seen in Fig. 6.1 with Fig. 6.2. In addition, we generate the sensor array data and compute the actual MSE values of the proposed CCSP RDB algorithm, according to the above conditions and compare the results to the analytical bounds in those two figures as well. The practical results are obtained by taking the average MSE result from $k = 1, \cdots, K$ and the same system parameters used for obtaining Fig. 6.6 in the Simulations section, except that the system input SNR is set to 10dB. The sets of matrix eigenvalues are captured and selected to be as close as possible to the analytical conditions assumed for ease of comparison. Also, by comparing the values and variations of the MSE bounds in these two figures, we can see that there is not obvious difference between the upper bounds. However, with a smaller eigenvalue spread $\sigma_{\lambda,k}$, the lower bound gets closer to the upper bound. The practical results obtained by using generated sensor array data indicate that only with smaller $\lambda_{max,k}$, the MSE gets closer to the upper bound.

6.4.2 Cross-Correlation and Subspace Projection Analysis

In this section, we present the performance analysis of the proposed CCSP RDB technique. In particular, this analysis is specific to the cross-correlation and subspace projection method used. At first we aim to exploit the properties of the cross-correlation vector $\mathbf{q}(i)$ estimated in (6.12). For convenience purposes, we omit the time index i in the
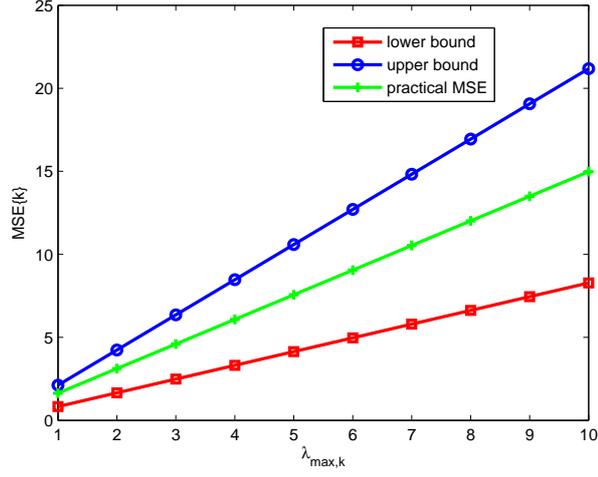
Figure 6.1: MSE bounds versus $\lambda_{max,k}$, $\sigma_{\lambda,k} = 0.9\lambda_{max,k}$
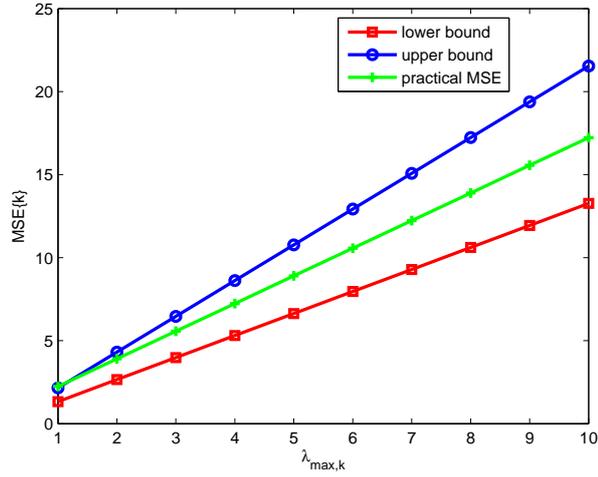
Figure 6.2: MSE bounds versus $\lambda_{max,k}$, $\sigma_{\lambda,k} = 0.5\lambda_{max,k}$

following analysis. By definition, we have

$$\mathbf{q} \triangleq E[z^*\mathbf{x}] = E[(\mathbf{g}^T\mathbf{W}\mathbf{x} + n)^*\mathbf{x}] = E[(\mathbf{g}^H\mathbf{W}^*\mathbf{x}^* + n^*)(\hat{\mathbf{F}}\mathbf{s} + \nu)]. \qquad (6.41)$$

Since $\mathbf{W}$ is diagonal, we have $\mathbf{W}^* = \mathbf{W}^H$. With the assumption that the noise n is uncorrelated with $\mathbf{s}$ and ν, the terms $E[n^*\hat{\mathbf{F}}\mathbf{s}]$ and $E[n^*\nu]$ are equal to zero and hence can

be discarded. Then from (6.41) we have

$$\mathbf{q} = E[\mathbf{g}^H \mathbf{W}^* \mathbf{x}^* \hat{\mathbf{F}} \mathbf{s} + \mathbf{g}^H \mathbf{W}^* \mathbf{x}^* \nu]$$

$$= E[\mathbf{g}^H \mathbf{W}^* (\hat{\mathbf{F}}^* \mathbf{s}^* + \nu^*) \hat{\mathbf{F}} \mathbf{s} + \mathbf{g}^H \mathbf{W}^* \mathbf{x}^* \nu]$$

$$= E[\hat{\mathbf{F}} \mathbf{s} \mathbf{s}^H \hat{\mathbf{F}}^H \mathbf{W} \mathbf{g}] + E[\nu \nu^H \mathbf{W} \mathbf{g}], \quad (6.42)$$

where $\hat{\mathbf{F}} = [\hat{\mathbf{f}}_1, \cdots, \hat{\mathbf{f}}_K]$ is the mismatched channel matrix and $\mathbf{s} = [s_1, \cdots, s_K]^T$. Now we expand the expressions for both $\hat{\mathbf{F}}$ and $\mathbf{s}$ in (6.42), assuming the signal sources are uncorrelated with each other. Then we obtain

$$\mathbf{q} = E[(\sum_{k=1}^{K} \hat{\mathbf{f}}_k s_k)(\sum_{k=1}^{K} \hat{\mathbf{f}}_k s_k)^H \mathbf{W} \mathbf{g}] + E[\nu \nu^H \mathbf{W} \mathbf{g}] = E[\sum_{k=1}^{K} s_k s_k^* \hat{\mathbf{f}}_k \hat{\mathbf{f}}_k^H \mathbf{W} \mathbf{g}] + E[\nu \nu^H \mathbf{W} \mathbf{g}].$$

$$(6.43)$$

At this stage, we substitute $\hat{\mathbf{f}}_k = \mathbf{f}_k + \mathbf{e}_k$ in (6.43). It should be noticed that $E[\mathbf{W} \mathbf{g}]$ is a deterministic quantity and $E[s_k s_k^*]$ and $E[\nu \nu^H]$ can be replaced by $P_{s,k}$ and P_n, respectively. Furthermore, with the assumption that $\mathbf{f}_k$ is uncorrelated with $\mathbf{e}_k$, (6.43) can be simplified to:

$$\mathbf{q} = E[\sum_{k=1}^{K} P_{s,k} \mathbf{f}_k \mathbf{f}_k^H + \mathbf{e}_k \mathbf{e}_k^H \mathbf{W} \mathbf{g}] + P_n \mathbf{W} \mathbf{g} = (\sum_{k=1}^{K} P_{s,k} E[(\mathbf{R}_{\mathbf{f}_k} + \mathbf{R}_{\mathbf{e}_k})] + P_n) \mathbf{W} \mathbf{g}. \quad (6.44)$$

Here we define the kth cross-correlation vector component as

$$\mathbf{q}_k \triangleq (P_{s,k} E[(\mathbf{R}_{\mathbf{f}_k} + \mathbf{R}_{\mathbf{e}_k})] + P_n) \mathbf{W} \mathbf{g}, \quad (6.45)$$

$$\mathbf{q} \triangleq \sum_{k=1}^{K} \mathbf{q}_k. \quad (6.46)$$

In order to introduce the cross-correlation vector and subspace projection approach, we substitute (6.46) in $\hat{\mathbf{f}}_k = \mathbf{P}_k \mathbf{q}$ (assuming it is already normalized as in (6.18)) and obtain

$$\hat{\mathbf{f}}_k = \mathbf{P}_k \sum_{k=1}^{K} \mathbf{q}_k. \quad (6.47)$$

If we assume that there is no error extracted by projecting any cross-correlation vector component $\mathbf{q}_l$ generated from the channel components $\hat{\mathbf{f}}_l$ ($1 \leq l \neq k \leq K$) onto the subspace projection matrix $\mathbf{P}_k$ so that $\mathbf{P}_k \mathbf{q}_l = 0$, then (6.47) can be simplified to

$$\hat{\mathbf{f}}_k = \mathbf{P}_k \mathbf{q}_k = \mathbf{P}_k (P_{s,k} E[(\mathbf{R}_{\mathbf{f}_k} + \mathbf{R}_{\mathbf{e}_k})] + P_n) \mathbf{W} \mathbf{g}. \quad (6.48)$$

From the MSE definition in (6.28), we have $\mathrm{MSE}\{\hat{\mathbf{f}}_k\}_2 = \mathrm{tr}(E[(\hat{\mathbf{f}}_k - \mathbf{f}_k)(\hat{\mathbf{f}}_k - \mathbf{f}_k)^H]) = E[(\hat{\mathbf{f}}_k - \mathbf{f}_k)^H(\hat{\mathbf{f}}_k - \mathbf{f}_k)]$. After substituting (6.48) in (6.28) we have

$$\mathrm{MSE}\{\hat{\mathbf{f}}_k\}_2 = E[(\mathbf{P}_k (P_{s,k} E[(\mathbf{R}_{\mathbf{f}_k} + \mathbf{R}_{\mathbf{e}_k})] + P_n) \mathbf{W} \mathbf{g} - \mathbf{f}_k)^H$$

$$(\mathbf{P}_k (P_{s,k} E[(\mathbf{R}_{\mathbf{f}_k} + \mathbf{R}_{\mathbf{e}_k})] + P_n) \mathbf{W} \mathbf{g} - \mathbf{f}_k)]. \quad (6.49)$$

124

After expanding (6.49), we obtain

$$\text{MSE}\{\hat{\mathbf{f}}_k\}_2 = E[\mathbf{g}^H\mathbf{W}(P_{s,k}\mathbf{R}_{\mathbf{f}_k} + P_{s,k}\mathbf{R}_{\mathbf{e}_k} + P_n)\mathbf{P}_k^H\mathbf{P}_k(P_{s,k}\mathbf{R}_{\mathbf{f}_k} + P_{s,k}\mathbf{R}_{\mathbf{e}_k} + P_n)\mathbf{W}\mathbf{g}]$$
$$- 2E[\mathbf{f}_k^H\mathbf{P}_k(P_{s,k}\mathbf{R}_{\mathbf{f}_k} + P_{s,k}\mathbf{R}_{\mathbf{e}_k} + P_n)\mathbf{W}\mathbf{g}] + E[\mathbf{f}_k^H\mathbf{f}_k]. \quad (6.50)$$

It should be noticed that $\mathbf{P}_k^H\mathbf{P}_k = \mathbf{P}_k = \mathbf{P}_k^H$ as the projection of a subspace projection matrix onto itself results in the same projection matrix. In addition, we have $\mathbf{f}_k^H\mathbf{P}_k = \mathbf{f}_k^H$ in the second term of (6.50), which can be simply verified. Since we have $\mathbf{f}_k = E[\mathbf{P}_k\mathbf{q}_k]$ by pre-multiplying both sides by $\mathbf{P}_k^H$ then we have $\mathbf{P}_k^H\mathbf{f}_k = E[\mathbf{P}_k^H\mathbf{P}_k\mathbf{q}_k] = E[\mathbf{P}_k\mathbf{q}_k] = \mathbf{f}_k$. Then by taking the Hermitian transpose on both sides gives $\mathbf{f}_k^H\mathbf{P}_k = \mathbf{f}_k^H$. Therefore, (6.50) can be rewritten as

$$\text{MSE}\{\hat{\mathbf{f}}_k\}_2 = E[\mathbf{g}^H\mathbf{W}(P_{s,k}\mathbf{R}_{\mathbf{f}_k} + P_{s,k}\mathbf{R}_{\mathbf{e}_k} + P_n)\mathbf{P}_k(P_{s,k}\mathbf{R}_{\mathbf{f}_k} + P_{s,k}\mathbf{R}_{\mathbf{e}_k} + P_n)\mathbf{W}\mathbf{g}]$$
$$- 2E[\mathbf{f}_k^H(P_{s,k}\mathbf{R}_{\mathbf{f}_k} + P_{s,k}\mathbf{R}_{\mathbf{e}_k} + P_n)\mathbf{W}\mathbf{g}] + E[\mathbf{f}_k^H\mathbf{f}_k]. \quad (6.51)$$

After expanding the internal multiplications and eliminating the uncorrelated ones with the assumption that $\mathbf{g}$ is uncorrelated with $\mathbf{f}_k$ and taking into account that $\mathbf{f}_k$ is normalized, i.e. $E[\mathbf{f}_k^H\mathbf{f}_k] = 1$, we obtain

$$\text{MSE}\{\hat{\mathbf{f}}_k\}_2 = E[\mathbf{g}^H\mathbf{W}(P_{s,k}^2\mathbf{R}_{\mathbf{e}_k}\mathbf{P}_k\mathbf{R}_{\mathbf{e}_k} + P_nP_{s,k}\mathbf{P}_k\mathbf{R}_{\mathbf{e}_k} + P_n^2\mathbf{P}_k)\mathbf{W}\mathbf{g} + 1]. \quad (6.52)$$

With (6.14), further simplifications can be made so we have

$$\text{MSE}\{\hat{\mathbf{f}}_k\}_2 = E[(P_{s,k}^2\epsilon^2||\mathbf{R}_{\mathbf{f}_k}||_F^2 + P_nP_{s,k}\epsilon||\mathbf{R}_{\mathbf{f}_k}||_F + P_n^2)\mathbf{g}^H\mathbf{W}\mathbf{P}_k\mathbf{W}\mathbf{g} + \mathbf{f}_k^H\mathbf{f}_k]$$
$$= (\frac{1}{3}P_{s,k}^2\epsilon_{max}^2E[||\mathbf{R}_{\mathbf{f}_k}||_F]^2 + \frac{1}{2}P_nP_{s,k}\epsilon_{max}E[||\mathbf{R}_{\mathbf{f}_k}||_F] + P_n^2)\mathbf{g}^H\mathbf{W}E[\mathbf{P}_k]\mathbf{W}\mathbf{g} + 1$$
$$\quad (6.53)$$

which monotonically increases with respect to $E[||\mathbf{R}_{\mathbf{f}_k}||_F]$. However, the results of the analysis can become more interesting if we compare the MSE obtained in the above two approaches. Let us denote them as $\text{MSE}\{\hat{\mathbf{f}}_k\}_1$ (described in (6.28)) and $\text{MSE}\{\hat{\mathbf{f}}_k\}_2$ (described in (6.53)), respectively, and $\mathbf{g}^H\mathbf{W}E[\mathbf{P}_k]\mathbf{W}\mathbf{g}$ as τ. If we compute their difference we have

$$\text{MSE}\{\hat{\mathbf{f}}_k\}_2 - \text{MSE}\{\hat{\mathbf{f}}_k\}_1 = (\frac{1}{3}P_{s,k}^2\epsilon_{max}^2E[||\mathbf{R}_{\mathbf{f}_k}||_F]^2 + \frac{1}{2}P_nP_{s,k}\epsilon_{max}E[||\mathbf{R}_{\mathbf{f}_k}||_F] + P_n^2)\tau$$
$$+ 1 - \frac{M}{2}\epsilon_{max}E[||\mathbf{R}_{\mathbf{f}_k}||_F]. \quad (6.54)$$

If we take the partial derivative of (6.54) with respect to τ, then we have $\frac{\partial\{\text{MSE}\{\hat{\mathbf{f}}_k\}_2 - \text{MSE}\{\hat{\mathbf{f}}_k\}_1\}}{\partial\tau} > 0$ which implies that $\text{MSE}\{\hat{\mathbf{f}}_k\}$ is proportionally and monotonically increasing with respect to τ. From (6.33) we have $\max\{||\mathbf{R}_{\mathbf{f}_k}||_F\} < \sqrt{M\lambda_{max,k}^2 - 2\sigma_{\lambda,k}\lambda_{max,k} + \sigma_{\lambda,k}^2} < \sqrt{M}\lambda_{max,k}$, which yields

$$\text{MSE}\{\hat{\mathbf{f}}_k\}_2 - \text{MSE}\{\hat{\mathbf{f}}_k\}_1 < (\frac{1}{3}P_{s,k}^2\epsilon_{max}^2 M\lambda_{max,k}^2 + \frac{1}{2}P_n P_{s,k}\epsilon_{max}\sqrt{M}\lambda_{max,k} + P_n^2)\tau$$
$$+ 1 - \frac{M}{2}\epsilon_{max}\sqrt{M}\lambda_{max,k}. \quad (6.55)$$

In other words, if the right-hand side of (6.55) is less than 0 or τ satisfies

$$\tau < \frac{\frac{M}{2}\epsilon_{max}\sqrt{M}\lambda_{max,k} - 1}{\frac{1}{3}P_{s,k}^2\epsilon_{max}^2 M\lambda_{max,k}^2 + \frac{1}{2}P_n P_{s,k}\epsilon_{max}\sqrt{M}\lambda_{max,k} + P_n^2}, \quad (6.56)$$

then $\text{MSE}\{\hat{\mathbf{f}}_k\}_2 - \text{MSE}\{\hat{\mathbf{f}}_k\}_1 < 0$ is true for all possible values of $E[||\mathbf{R}_{\mathbf{f}_k}||_F]$, which indicates a smaller MSE result from approach 2 ($\text{MSE}\{\hat{\mathbf{f}}_k\}_2$) as compared to approach 1 ($\text{MSE}\{\hat{\mathbf{f}}_k\}_1$). Interestingly, this indicates that using prior knowledge about the mismatch in the form of cross-correlation and subspace processing can result in smaller values of MSE. However, the only term of τ that has to be determined is the subspace projection matrix $\mathbf{P}_k$, which is dependent on its subspace properties and can be further exploited using PCA methods that can be found in the literature [101].

6.5 Simulations

In the simulations, we compare the proposed CCSP RDB algorithm for the case where the perfect CSI is known, the case where no robust method is used when the CSI is imperfect and for the cases where the CSI is imperfect and the several existing robust approaches [7, 80, 96–99, 103, 104, 107, 108] (i.e. worst-case SDP online programming) are used. The simulation metrics considered include the system output SINR versus input SNR as well as the maximum allowable total transmit power P_T. We also examine the incoherent scenarios, where some of the interferers are strong enough as compared to the desired signal and the noise. In all simulations, the system input SNR is known and can be controlled by adjusting only the noise power. Both of the channels $\mathbf{F}$ and $\mathbf{g}$ follow the Rayleigh distributed whereas the mismatch is only considered for $\mathbf{F}$. The shadowing and path loss effects are taken into account where the path loss exponent $\rho = 2$, the

source-to-destination power path loss $L = 10$dB and the shadowing spread $\sigma_s = 3$dB. The total number of relays and signal sources are set as $M = 8$ and $K = 3$, respectively. The system interference-to-noise ratio (INR) is fixed at 10dB or otherwise specified. The number of principal components is manually selected to optimize the performance for the proposed CCSP RDB algorithm. A total number of 100 snapshots is considered.

At first, we examine the SINR performance in terms of a variation of input SNRs (i.e. -10dB to 20dB), while limiting the maximum allowable transmit power to $P_T = 1$dBW, for all the compared cases. The powers of interferers are equally spared by he interferers at this moment. We illustrate their SINR versus SNR performances with different values of the mismatch parameter ϵ_{max}, i.e. $\epsilon_{max} = 0.2$, $\epsilon_{max} = 0.5$ and $\epsilon_{max} = 1.0$ in Fig. 6.3, Fig. 6.4 and Fig. 6.5, respectively. The worst-case SDP method is adopted from [96], in which the values of ϵ_{max} are set to be consistent for all the mismatched matrix quantities. The results show that the proposed CCSP RDB method perfectly preserves the robustness against the increase of channel error level and behave close to the case of perfect CSI, whereas the worst-case SDP method suffers performance degradations against the increase of channel error levels.
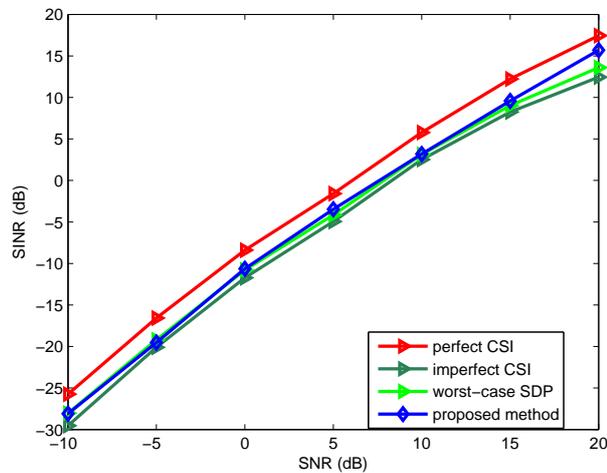
Figure 6.3: SINR versus SNR, $P_T = 1$dBW, $\epsilon_{max} = 0.2$, INR=10dB

Secondly, we examine the SINR performance with respect to a variation of maximum allowable total transmit power P_T (i.e. 1dBW to 5dBW) by fixing the SNR at a certain level (i.e. SNR=10dB). We still consider the same INR and all interferers have the same power. This time, we fix the perturbation parameter at $\epsilon_{max} = 0.5$ for all compared
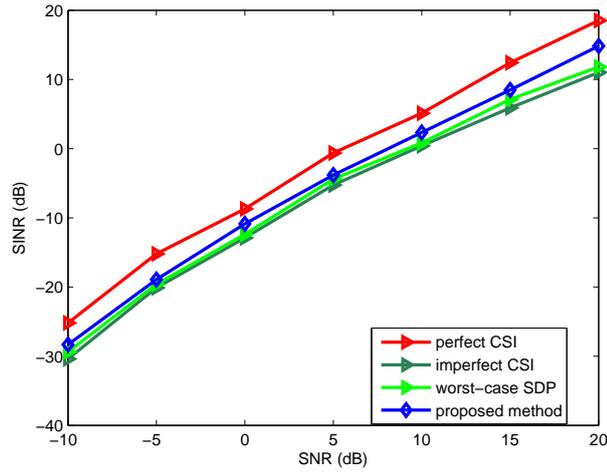
Figure 6.4: SINR versus SNR, $P_T = 1$dBW, $\epsilon_{max} = 0.5$, INR=10dB
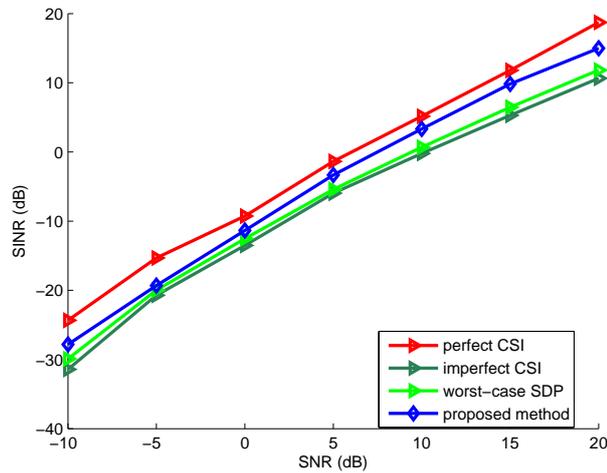
Figure 6.5: SINR versus SNR, $P_T = 1$dBW, $\epsilon_{max} = 1.0$, INR=10dB

algorithms. In Fig. 6.6, it shows the output SINR increases as we lift up the limit for the maximum allowable transmit power and it makes a substantial difference when a robust approach is used. The proposed CCSP RDB method still outperforms the worst-case SDP algorithm and perform close to the case where we have a perfect CSI.

In the last example, we increase the system INR from 10dB to 20dB. We still consider $K = 3$ users (which means there are two interferers in total) but rearrange the powers of the interferers so that one of them is much stronger than the other. Specifically, we examine the compared algorithms in an incoherent scenario and set the power ratio of
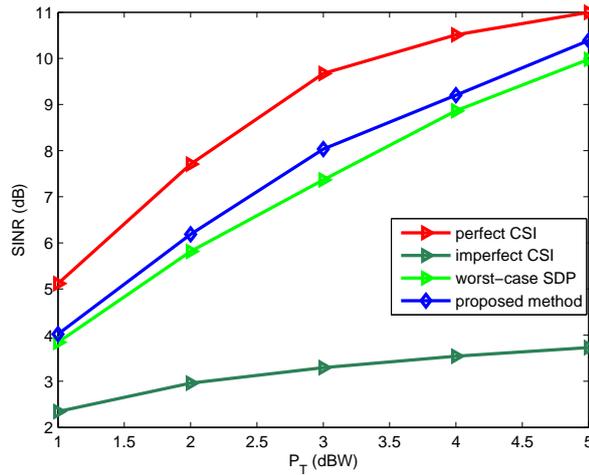
Figure 6.6: SINR versus P_T, SNR=10dB, $\epsilon_{max} = 0.5$, INR=10dB

the stronger interferer over the weaker one to 10. The maximum allowable total transmit power P_T and the perturbation parameter ϵ_{max} are fixed at 1dBW and 0.2, respectively. We observe the SINR performance versus SNR for these algorithms and illustrate the results in Fig. 6.7. Then we fix the system SNR at 10dB and observe the output SINR performance versus snapshots as in Fig. 6.8. It can be seen that all the algorithms have performance degradations due to the adoption of strong interferers as well as their power distribution. However, the proposed CCSP RDB algorithm has excellent robustness in terms of the system output SINR against the presences of strong interferers with unbalanced power distribution. Especially with relative high system SNRs, the CCSP RDB algorithm is able to perform extremely close to the case of perfect CSI.

6.6 Summary

We have devised a novel RDB approach based on the exploitation of the cross-correlation between the received data from the relays at the destination and the system output as well as a subspace projection method to estimate the channel errors. In the proposed CCSP RDB method, a total relay transmit power constraint has been applied to the system with the objective of maximizing the output SINR. A performance analysis of the CCSP RDB technique has been carried out. The proposed CCSP RDB method does not require any
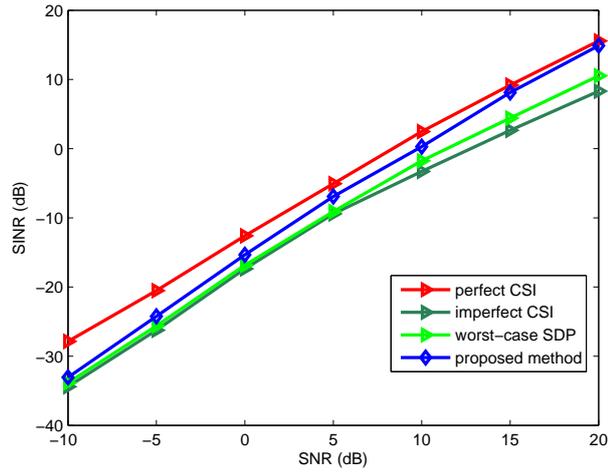
Figure 6.7: SINR versus SNR, $P_T = 1\text{dBW}$, $\epsilon_{max} = 0.2$, INR=20dB
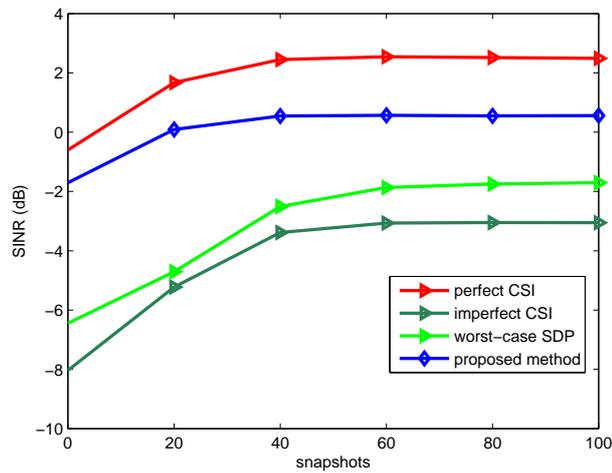
Figure 6.8: SINR versus snapshots, $P_T = 1\text{dBW}$, $\epsilon_{max} = 0.2$, SNR=10dB, INR=20dB

online optimization procedure and the simulation results show an excellent performance as compared to the existing approaches.

Chapter 7

Conclusions and Future Work

Contents

7.1 Conclusions

In this thesis, sensor array beamforming algortihms as well as relay selection techniques have been investigated for applications of conventional beamformers and distributed beamforming using relay networks. For conventional beamforming, low-complexity RAB techniques that are based on shrinkage methods, eigen subspace projection and Krylov subspace projection techniques have been proposed. For distributed beamforming, cooperative relay networks with relay selections and output SINR maximization techniques and robust distributed beamforming techniques have been proposed. The corresponding system models have been described and the techniques have been introduced in an algorithmic manner. Moreover, computational complexity and performance analysis and practical simulations have also been presented for those techniques.

In Chapter 3, low-complexity RAB techniques based on shrinkage methods have been proposed. We have firstly derived a LOCMSE batch algorithm to estimate the desired signal steering vector mismatch, in which the INC matrix is also estimated with a recursive

matrix shrinkage method. Then we have developed low-complexity adaptive recursive versions of SG and CG to update the beamforming weights from a series of modified optimization problems, resulting in low-cost LOCSME-SG, LOCSME-CCG and LOCSME-MCG robust adaptive algorithms. An analysis of the effect of shrinkage on the estimation procedure has been developed to have a deep investigation on how it affects the estimation accuracy and shown to provide better performance. A computational complexity comparison has also been presented and discussed for the proposed and existing algorithms. In the simulations, a diverse set of scenarios have been considered and the performance of the proposed algorithms has been illustrated and discussed.

In Chapter 4, cost-effective low-rank RAB techniques have been proposed. The techniques include a batch OKSPME method and novel RAB algorithms that are based on the exploitation of the cross-correlation between the array observation data and the output of the beamformer, while using a Krylov subspace. In OKSPME, we have constructed a general linear equation considered in large dimensions whose solution helps to update the steering vector mismatch. Then, an FOM method that aims to form an orthogonal Krylov subspace has been introduced to iteratively estimate the steering vector mismatch in a reduced-dimensional subspace, so that the system complexity can be significantly reduced and controlled when dealing with high dimension subspaces or large sensor arrays. Adaptive algorithms based on SG and CG techniques have been derived based on reformulated optimization problems to update the beamforming weights, for the purpose of further reducing the complexity in weight vector computations. In the simulations, we have considered multiple scenarios including both time-invariant and time-varying cases and the results have shown excellent performance in terms of the output SINR of the proposed RAB algorithms among all the compared RAB methods.

In Chapter 5, relay selection methods for distributed beamforming have been proposed. Specifically, we have proposed joint MMSE consensus relay and selection schemes with a total power constraint and local communications among the relays for a network with cooperating sensors, then we have devised greedy relay selection algorithms named LMMSEC-G and SMMSEC-G, based on the MMSE consensus approach that optimize the network performance. Moreover, we have also proposed the MSINR relay selection algorithms for distributed beamforming, with a total relay transmit power constraint that iteratively optimizes both the beamforming weights at the relays nodes, maximizing the

SINR at the destination, which are named as RGSRS relay selection algorithm. The proposed greedy relay selection scheme has also been compared to other schemes like RRRS and RESRS. A complexity analysis has been provided and simulation results have shown that the proposed algorithms achieve excellent BER and SINR performances.

In Chapter 6, we have proposed a novel CCSP RDB technique, which does not require any online optimization procedure as compared to previously reported RDB algorithms. The CCSP RDB approach relies on a total relay transmit power constraint and aims to maximize the output SINR. The channel errors are modeled using an additive matrix perturbation method, which results in degradation of the system performance. A performance analysis of the proposed CCSP robust technique has also been provided.

7.2 Future Work

Many of the methods and algorithms introduced in this thesis have potential applications in other systems and scenarios outside the scope of this thesis, and there is further work and analysis that could be considered to extend the work that has been covered. RAB and RDB techniques are mainly designed for the preservation of performance of beamforming and relay systems. However, the error or mismatch estimation methods can be used in many other areas like biomedical signal processing, applied mathematics, estimation and detection techniques.

The LOCSME and OKSPME RAB techniques in Chapter 3 and 4 can be also combined with widely-linear signal processing, where noncircular signals or data models and their associated augmented statistics must be considered for optimal performance. Specifically, the linear shrinkage method from LOCSME can be extended to take non-linear parameters or coefficients into account so as to fit non-linear data model or processes, in which case the complexity may have some increase. Since the steering vector mismatch estimation is independent from the estimation of the covariance matrix, the matrix shrinkage method in LOCSME can be employed in the covariance matrix estimation of OKSPME, which can be also implemented in an adaptive mechanism.

In the applications of compressive sensing, where sparse signal models and systems

are considered, in which case it is still possible to derive sparse versions of signal esti-
mation and reconstruction algorithms based on LOCSME or OKSPME. Interestingly, if
the exploitation of the cross-correlation between the sensor array observation data and the
system output and the subspace projection approaches work well for sparse systems, then
we can make full use of the information from a limited number of available sensors to im-
prove the accuracy of signal estimation and the quality of reconstruction without requiring
much computational complexity. In other words, to preserve or enhance the system per-
formance using the output SNR or SINR metrics. Similar MSE or MMSE analysis will
also be possible to evaluate the system performance.

Besides the greedy-like search based relay selection algorithms in this thesis, it is also
possible to combine the cooperative MMSE consensus method with many other search
algorithms like genetic algorithms. A comprehensive comparison considering multiple
aspects including computational complexity, system capacity, compatibility and perfor-
mance of interest among all the typical relay selection search algorithms (greedy search,
exhaustive search, random search etc.) should be carried out. In addition, the proposed
CCSP RDB technique can be associated with adaptive algorithms like SG or CG to reduce
the complexity and make it implementable and practical in most applications.

Glossary

AF	amplify-and-forward
BER	bit-error-rate
BPSK	binary phase shift keying
CCG	conventional conjugate gradient
CCSP	cross-correlation and subspace projection
CF	compress-and-forward
CG	conjugate gradient
LCMV	linear constrained minimum variance
CSI	channel state information
DF	decode-and-forward
DL	diagonal loading
DoA	direction of arrival
FOM	full orthogonalization method
INC	interference-plus-noise covariance
INR	interferences-to-noise ratio
JIO	joint iterative optimization
KA	knowledge-aided
LCWC	low-complexity worst-case
LMS	least mean squares
LOCME	low-complexity mismatch estimation
LOCSME	low-complexity shrinkage-based mismatch estimation
MAO	modified array observation
MCG	modified conjugate gradient
MJIO	modified joint iterative optimization

ML	maximum likelihood
MMSE	minimum mean squared error
MSE	mean squared error
MSINR	maximum signal-to-interference-plus-noise ratio
MVDR	minimum variance distortionless response
OAS	oracle approximating shrinkage
OKSPME	orthogonal Krylov subspace projection mismatch estimation
PCA	principal component analysis
QoS	quality of service
RAB	robust adaptive beamforming
RCB	robust Capon beamformer
RDB	robust distributed beamforming
RESRS	restricted exhaustive search relay selection
RGSRS	restricted greedy search relay selection
RLS	recursive least squares
RRRS	restricted random relay selection
SCM	sample covariance matrix
SCV	sample correlation vector
SDP	semi-definite programme
SDR	semi-definite relaxation
SG	stochastic gradient
SINR	signal-to-interference-plus-noise ratio
SIR	signal-to-interference ratio
SMI	sampled matrix inversion
SNR	signal-to-noise ratio
SOCP	second order cone programme
SoI	signal of interest
SQP	sequential quadratic programme
UCA	uniform circular array
ULA	uniform linear array

Bibliography

[1] H. L. Van Trees, *Optimum Array Processing*, New York: Wiley, pp.17-89, 710-826, 2002.

[2] A. H. Sayed, *Adaptive Filters*, Published by John Wiley & Sons, Inc., Hoboken, New Jersey, 2008.

[3] H. Krim and M. Viberg, "Two Decades of Singal Processing Research", *IEEE Signal Processing Magzine*, pp. 67-94, July 1996.

[4] B. V. Veen and K. M. Buckley, "Beamforming Techniques for Spatial Filtering", 2000 CRC Press LLC.

[5] I. Psaromiligkos and S. Batalama, "Interference-plus-noise covariance matrix estimation for adaptive space-time processing of DS/CDMA signals, in IEEE Fall Vehicular Technology Conference, pp. 21972204, Oct 2000.

[6] Warnick and Jensen, "Antennas and Propagation for Wireless Communications", April 5, 2011.

[7] S. A. Vorobyov, A. B. Gershman and Z. Luo, "Robust Adaptive Beamforming using Worst-Case Performance Optimization: A Solution to Signal Mismatch Problem", *IEEE Transactions on Signal Processing*, Vol. 51, No. 4, pp 313-324, Feb 2003.

[8] J. Li, P. Stoica and Z. Wang, "On Robust Capon Beamforming and Diagonal Loading", *IEEE Transactions on Signal Processing*, Vol. 57, No. 7, pp 1702-1715, July 2003.

[9] D. Astely and B. Ottersten, "The effects of Local Scattering on Direction of Arrival Estimation with Music", *IEEE Transactions on Signal Processing*, Vol. 47, No. 12, pp 3220-3234, Dec 1999.

[10] S. A. Vorobyov, "Principles of Minimum Variance Robust Adaptive Beamforming Design", *Signal Processing* (2012), http://dx.doi.org/10.1016/j.sigpro.2012.10.021

[11] A. Khabbazibasmenj, S. A. Vorobyov and A. Hassanien, "Robust Adaptive Beamforming Based on Steering Vector Estimation with as Little as Possible Prior Information", *IEEE Transactions on Signal Processing*, Vol. 60, No. 6, pp 2974-2987, June 2012.

[12] A. Hassanien, S. A. Vorobyov and K. M. Wong, "Robust Adaptive Beamforming Using Sequential Quadratic Programming: An Iterative Solution to the Mismatch Problem", *IEEE Signal Processing Letters*, Vol. 15, pp 733-736, 2008.

[13] L. Landau, R. de Lamare, M. Haardt, "Robust Adaptive Beamforming Algorithms Using Low-Complexity Mismatch Estimation", *Proc. IEEE Statistical Signal Processing Workshop*, Nice, France, 2011.

[14] Y. Gu and A. Leshem, "Robust Adaptive Beamforming Based on Jointly Estimating Covariance Matrix and Steering Vector", *Proc. International Conference on Acoustics Speech and Signal Processin (ICASSP)*, pp 2640-2643, 2011.

[15] Y. Gu and A. Leshem, "Robust Adaptive Beamforming Based on Interference Covariance Matrix Reconstruction and Steering Vector Estimation", *IEEE Transactions on Signal Processing*, Vol. 60, No. 7, pp 3881-3885, July 2012.

[16] Y. Chen, A. Wiesel and A. O. Hero III, "Shrinkage Estimation of High Dimensional Covariance Matrices", *Proc. International Conference on Acoustics Speech and Signal Processin (ICASSP)*, pp 2937-2940, 2009.

[17] O. Ledoit and M. Wolf, "A well-conditioned estimator for large dimensional covariance matrix", Journal of Multivariate Analysis 88 (2004) 365-411, Feb 2001.

[18] J. Zhuang and A. Manikas, "Interference cancellation beamforming robust to pointing errors," IET Signal Process., Vol. 7, No. 2, pp. 120-127, April 2013.

[19] A. Elnashar, "Efficient implementation of robust adaptive beamforming based on worst-case performance optimization", IET Signal Process., Vol. 2, No. 4, pp. 381-393, 2008.

[20] R. Fa, R. de Lamare and V. H. Nascimento, "Knowledge-aided STAP algorithm using convex combination of inverse covariance matrices for heterogeneous clutter," *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*, pp 2742-2745, 2010.

[21] L. Wang, "Array Signal Processing Algorithms for Beamforming and Direction Finding", Ph.D. Thesis, Department of Electronics, University of York, pp. 14-17, 40-44, Dec 2009.

[22] S. W. Varade and K. D. Kulat, "Robust Algorithms for DOA Estimation and Adaptive Beamforming for Smart Antenna Application", *Second International Conference on Emerging Trends in Engineering and Technology (ICETET)*, pp 1195-1200, 2009.

[23] L. Wang and R. de Lamare, "Adaptive Reduced-Rank Constrained Constant Modulus Algorithms Based on Joint Iterative Optimization of Filters for Beamforming", *Statistical Signal Processing (SSP)*, pp 152-156, 2009.

[24] L. Wang and R. C. de Lamare, "Constrained adaptive filtering algorithms based on conjugate gradient techniques for beamforming," IET Signal Process., Vol. 4, No. 6, pp. 686697, Feb 2010.

[25] N. H. Noordin, V. Zuniga, A. O. El-Rayis, N. Haridas, A. T. Erdogan and Tughrul Arslan, "Uniform Circular Arrays for Phased Array Antenna", *Antennas and Propagation Conference (LAPC)*, pp 1-4, Nov 2011.

[26] C. Zhang, M. Zhao and Y. Cai, "Joint 2-D DOA estimation and mutual coupling calibration for uniform rectangular arrays", *International Conference on Wireless Communications and Signal Processing (WCSP)*, 2015.

[27] P. Rocca, R. L. Haupt and A. Massa, "Interference Suppression in Uniform Linear Arrays Through a Dynamic Thinning Strategy", *IEEE Transactions on Antennas and Propagation*, Vol. 59, Issue. 12, pp 4525-4533, 2011.

[28] X. Lan, Y. Li and E. Wang, "A RARE Algorithm for 2D DOA Estimation Based on Nested Array in Massive MIMO System", *IEEE Access*, Vol. 4, pp 3806-3814, 2016.

[29] A. Tawfik, "A generic processing structure decomposing the beamforming process of 2-D and 3-D arrays of sensors into sub-sets of coherent process", *National Radio Science Conference (NRSC)*, 1999.

[30] S. Li, R. C. de Lamare and M. Haardt, "Adaptive Frequency-Domain Group-Based Shrinkage Estimators for UWB Systems", *IEEE Trans. Veh. Technol.*, Vol. 62, No. 8, pp. 3639-3652, Oct 2013.

[31] H. Ruan and R. C. de Lamare, "Robust Adaptive Beamforming Using a Low-Complexity Shrinkage-Based Mismatch Estimation Algorithm," *IEEE Sig. Proc. Letters.*, Vol. 21, No. 1, pp 60-64, Nov 2013.

[32] P. Stoica, J. Li, X. Zhu, and J. R. Guerci, "On Using a priori Knowledge in Space-Time Adaptive Processing, IEEE Transactions on Signal Processing, Vol. 56, No. 6, pp. 2598-2602, June 2008.

[33] S. E. Nai, W. Ser, Z. L. Yu and H. Chen, "Iterative Robust Minimum Variance Beamforming," *IEEE Transactions on Signal Processing*, Vol. 59, NO. 4, pp. 1601-1611, April 2011.

[34] Z. L. Yu, Z. Gu, J. Zhou, Y. Li, S. Wee, M. H. Er, "A Robust Adaptive Beamformer Based on Worst-Case Semi-Definite Programming," *IEEE Transactions on Signal Processing*, vol. 58, No. 11, pp. 5914-5919, 2010.

[35] J. P. Lie, W. Ser and C. M. S. See, "Adaptive Uncertainty Based Iterative Robust Capon Beamformer Using Steering Vector Mismatch Estimation," *IEEE Transactions on Signal Processing*, Vol. 59, No. 9, pp. 4483-4488, Sep 2011.

[36] M. Morelli, L. Sangunietti and U. Mengali, "Channel Estimation for Adaptive Frequency-Domain Equalization," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 5, pp. 53-57, Sep 2005.

[37] B. Liao, S. C. Chan, and K. M. Tsui, "Recursive Steering Vector Estimation and Adaptive Beamforming Under Uncertainties," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 489-501, Jan 2013.

[38] M. H. Gutknecht, "A Brief Introduction to Krylov Space Methods for Solving Linear Systems," *ETH Zurich, Seminar for Applied Mathematics*, [Online]

available FTP: http://www.sam.math.ethz.ch/ mhg/pub/biksm.pdf, available email: mhg@math.ethz.ch, May 2008.

[39] H. A. van der Vorst, "Iterative Krylov Methods for Large Linear Systems," Cambridge University Press, 2003.

[40] Z. Bai, "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems," *Applied Numerical Mathematics*, Department of Computer Science, University of California, Davis, CA 95616, USA, 2002.

[41] R. C. de Lamare, M. Haardt and R. Sampaio-Neto, "Blind Adaptive Constrained Reduced-Rank Parameter Estimation Based on Constant Modulus Design for CDMA Interference Suppression," *IEEE Transactions on Signal Processing*, Vol. 56, No. 6, pp 2470-2482, June 2008.

[42] H. Ge, I. P. Kirsteins and L. L. Scharf, "Data Dimension Reduction Using Krylov Subspaces Making Adaptive Beamformers Robust to Model Order-Determination," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 4, pp IV-IV, 2006.

[43] I. P. Kirsteins and H. Ge, "Performance Analysis of Krylov Space Adaptive Beamformers," *Sensor Array and Multichannel Processing, Fourth IEEE Workshop*, pp 16-20, July 2006.

[44] S. D. Somasundaram, N. H. Parsons, P. Li and R. C. de Lamare, "Data-Adaptive Reduced-Dimension Robust Capon Beamforming," *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp 4159-4163, May 2013.

[45] S. D. Somasundaram, N. H. Parsons, P. Li and R. C. de Lamare, "Reduced-Dimension Robust Capon Beamforming Using Krylov-Subspace Techniques," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 1, pp 270-289, Jan 2015.

[46] V. Druskin and V. Simoncini, "Adaptive Rational Krylov Subspaces for Large-Scale Dylymical Systems," *Systems and Control Letters*, Vol. 60, No. 8, pp 546-560, 2011.

[47] K. Carlberg and C. Farhat, "An Adaptive POD-Krylov Reduced-Order Model for Structural Optimization," *8th World Congress on Structural and Multidisciplinary Optimization*, Lisbon, Portugal, June 2009.

[48] A. Hassanien and S. A. Vorobyov, "A Robust Adaptive Dimension Reduction Technique With Application to Array Processing," *IEEE Sig. Proc. Letters.* Vol. 16, No. 1, pp 22-25, Jan 2009.

[49] M. Yukawa, R. C. de Lamare and I. Yamada, "Robust Reduced-Rank Adaptive Algorithm Based on Parallel Subgradient Projection and Krylov Subspace," *IEEE Transactions on Signal Processing*, Vol. 57, No. 12, pp 4660-4674, Dec 2009.

[50] C. Dumard and T. Zemen, "Double Krylov Subspace Approximation for Low Complexity Iterative Multi-User Decoding and Time-Variant Channel Estimation," *6th IEEE Workshop on Signal Processing Advances in Wireless Communications*, pp 328-332, June 2005.

[51] R. C. de Lamare and R. Sampaio-Neto, "Reduced-Rank Adaptive Filtering Based on Joint Iterative Optimization of Adaptive Filters," *IEEE Sig. Proc. Letters.* Vol. 14, No. 12, pp 980-983, Dec 2007.

[52] N. Song, W. U. Alokozai, R. C. de Lamare and M. Haardt, "Adaptive Widely Linear Reduced-Rank Beamforming Based on Joint Iterative Optimization," *IEEE Sig. Proc. Letters.* Vol. 21, No. 3, pp 265-269, Mar 2014.

[53] S. D. Somasundaram, "A Framework for Reduced Dimension Robust Capon Beamforming," *IEEE Statistical Signal Processing Workshop (SSP)*, pp 425-428, 2011.

[54] P. Li and R. C. de Lamare, "Low-Complexity Robust Data-Dependant Dimensionality Reduction Based on Joint Iterative Optimization of Parameters," *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp 49-52, Dec 2013.

[55] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Processing Based on Joint and Iterative Interpolation, Decimation, and Filtering," *IEEE Transactions on Signal Processing*, Vol. 57, No. 7, pp 2503-2514, July 2009.

[56] L. Wang, R. C. de Lamare and M. Haardt, "Direction finding Algorithms with Joint Iterative Subspace Optimization," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 4, pp 2541-2553, Oct 2014.

[57] C. Zhang, R. Tian, X. Tan, "A Leakage-Based Dimensionality Reduction Beamforming for MIMO Cognitive Radio Networks," *8th International Conference on Communications and Networking in China (CHINACOM), pp 12-16, 2013.*

[58] R. Fa, R. C. de Lamare and D. Zanatta-Filho, "Reduced-Rank STAP Algorithm for Adaptive Radar Based on Joint Iterative Optimization of Adaptive Filters," *42nd Asilomar Conference on Signals, Systems and Computers*, pp 533-537, Oct 2008.

[59] R. Fa, R. C. de Lamare and P. Clarke, "Reduced-Rank STAP for MIMO Radar Based on Joint Iterative Optimization of Knowledge-Aided Adaptive Filters," *Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp 496-500, Nov 2009.

[60] S.D. Somasundaram, "Reduced dimension robust Capon beamforming for large aperture passive sonar arrays," IET Radar, Sonar and Navigation, Vol. 5, No. 7, pp 707-715, 2011.

[61] Popoviciu, "Sur Les Equations Algebriques Ayant Toutes Leurs Racines Relles," Mathematica 9, pp 129-145, 1935.

[62] N. Song, R. C. de Lamare, M. Haardt and M. Wolf, "Adaptive Widely Linear Reduced-Rank Interference Suppression Based on the Multistage Wiener Filter," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4003-4016, Aug. 2012.

[63] R. C. de Lamare and R. Sampaio-Neto, "Reduced-Rank Spaceime Adaptive Interference Suppression With Joint Iterative Least Squares Algorithms for Spread-Spectrum Systems," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1217-1228, March 2010.

[64] R. Fa, R. C. de Lamare and L. Wang, "Reduced-Rank STAP Schemes for Airborne Radar Based on Switched Joint Interpolation, Decimation and Filtering Algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4182-4194, Aug. 2010.

[65] H. Ruan and R. C. de Lamare, "Robust Adaptive Beamforming Based on Low-Rank and Cross-Correlation Techniques," *23rd European Signal Processing Conference (EUSIPCO)*, pp. 854-858, 2015.

[66] R. Mudumbai, D.R. Brown III, U. Madhow, and H.V. Poor, "Distributed Transmit Beamforming Challenges and Recent Progress," *IEEE Communications Magazine*, Vol. 47, Issue. 2, pp. 102-110, 2009.

[67] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex Optimization-Based Beamforming," *IEEE Signal Processing Magazine*, Vol. 27, Issue. 3, pp. 62-75, May 2010.

[68] J. Uher, T. A. Wysocki, and B. J. Wysocki, "Review of Distributed Beamforming," *Journal of Telecommunications and Information Technology*, 2011.

[69] V. H. Nassab, S. Shahbazpanahi, A. Grami, and Z. Luo, "Distributed Beamforming for Relay Networks Based on Second-Order Statistics of the Channel State Information," *IEEE Trans. Signal Process.*, Vol. 56, No. 9, pp. 4306-4316, Sep 2008.

[70] Y. Jing and H. Jafarkhani, "Network Beamforming Using Relays With Perfect Channel Information," *IEEE Trans. Information Theory*, Vol. 55, No. 6, pp. 2499-2517, June 2009.

[71] V. Havary-Nassab, S. Shahbazpanahi, and A. Grami, "Optimal Distributed Beamforming for Two-Way Relay Networks," *IEEE Trans. Signal Process.*, Vol. 58, No. 3, pp. 1238-1250, March 2010.

[72] K. Zarifi, S. Zaidi, S. Affes, and A. Ghrayeb, "A Distributed Amplify-and-Forward Beamforming Technique in Wireless Sensor Networks," *IEEE Trans. Signal Process.*, Vol. 59, No. 8, pp. 3657-3674, Aug 2011.

[73] Y. Jing, and H. Jafarkhani, "Single and Multiple Relay Selection Schemes and their Achievable Diversity Orders," *IEEE Trans. Wireless Commun.*, Vol. 8, No. 3, pp. 1414-1423, March 2009.

[74] P. Clarke and R. C. de Lamare, "Transmit Diversity and Relay Selection Algorithms for Multirelay Cooperative MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1084-1098, March 2012.

[75] J. Gu and R. C. de Lamare, "Joint PIC and Relay Selection based on Greedy Techniques for Cooperative DS-CDMA Systems," *Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing*, pp. 2754-2758, 4-9 May 2014.

[76] J. Gu and R. C. de Lamare, "Joint Interference Cancellation and Relay Selection Algorithms Based on Greedy Techniques for Cooperative DS-CDMA Systems", *EURASIP Journal on Wireless Communications and Networking*, 2016.

[77] T. Hesketh, R, C. de Lamare and S. Wales, "Joint maximum likelihood detection and link selection for cooperative MIMO relay systems," *IET Communications*, Vol. 8, Issue 14, pp. 2489-2499, Sep 2014.

[78] T. J. Hesketh, "Detection and Resource Allocation Algorithms for Cooperative MIMO Relay Systems," *Ph.D. Thesis, University of York, Electronics, Communications Research Group*, Feb 2014.

[79] J. Xu, H. Zhang, D. Yuan, Q. Jin, and C. Wang, "Novel Multiple Relay Selection Schemes in Two-Hop Cognitive Relay Networks," *Third International Conference on Communications and Mobile Computing*, pp. 207-310, April 2011.

[80] L. Zhang, W. Liu, A. Quddus, M. Dianati, and R. Tafazolli,"Adaptive Distributed Beamforming for Relay Networks based on Local Channel State Information," *IEEE Trans. on Signal and Information Process. over Networks*, Vol. 1, Issue 2, pp. 117-128, June 2015.

[81] L. Blanco and M. Najar, "Sparse Multiple Relay Selection for Network Beamforming with Individual Power Constraints Using Semidefinite Relaxation," *IEEE Trans. Wireless Commun.*, Vol. PP, Issue 99, pp. 1, Oct 2015.

[82] J. Kim, A. Ikhlef, and R. Schober, "Combined Relay Selection and Cooperative Beamforming for Physical Layer Security," *Communications and Networks*, Vol. 14, Issue 4, pp. 364-373, Aug 2012.

[83] U. Niesen, D. Shah, and G. W. Wornell, "Adaptive alternating minimization algorithms," *IEEE Transactions on Information Theory*, vol. 55, no. 3, pp. 1423-1429, March 2009.

[84] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Processing Based on Joint and Iterative Interpolation, Decimation, and Filtering," *IEEE Transactions on Signal Processing*, Vol. 57, No. 7, July 2009.

[85] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Equalization Algorithms Based on Alternating Optimization Design Techniques for MIMO Sys-

tems," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 6, pp. 2482-2494, July 2011.

[86] T. Wang, R. C. de Lamare and A. Schmeink, "Alternating Optimization Algorithms for Power Adjustment and Receive Filter Design in Multihop Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 173-184, Jan. 2015.

[87] L. Landau, "Advanced Robust Adaptive Beamforming for Wireless Networks," *Master Thesis, Technische Universitat Ilmenau Fakultat fur Elektrotechnik und Informationstechnik*, May 2011.

[88] J. Choi, "Distributed Beamforming Using a Consensus Algorithm for Cooperative Relay Networks," *IEEE Communication Letters*, VOL. 15, Issue 4, pp. 368-370, April 2011.

[89] B. Johannson, C. M. Carretti, and M. Johansson, "On Distributed Optimization Using peer-to-peer Communications in Wireless Sensor Networks," *Proc. IEEE SECON*, pp. 497-505, 2008.

[90] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew, "eneralized Consensus Computation in Networked Systems with Erasure Links," *Proc. IEEE SPAWC*, pp. 1088-1092, 2005.

[91] S. Schedler and V. Kuehn, "Resource Allocation for Distributed Beamforming with Multiple Relays and Individual Power Constraints," *Proc. International Sympositum Wireless Communications Systems (ISWCS)*, pp 1-5, Aug 2014.

[92] O. T. Demir, T. E. Tuncer, "Optimum Discrete Distributed Beamforming for Single Group Multi Casting Relay Networks with Relay Selection," *ICASSP 2015*, pp 2524-2528, April 2015.

[93] M. Dahleh, M. A. Dahleh, G. Verghese, "Dynamic Systems and Control," *Department of Electrical Engineering and Computer Science Massachuasetts Institute of Technology*, 2011.

[94] M. A. Maleki, S. Mehrizi, M. Ahmadian, "Distributed Robust Beamforming in Multi-User Relay Network," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 904-907, 2014.

[95] J. Steinwandt and M. Haardt, "Optimal widely-linear distributed beamforming for relay networks," *Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing*, pp. 4241-4245, May 2013.

[96] B. Mahboobi, M. Ardebilipour, A. Kalantari and E. Soleimani-Nasab, "Robust Co-operative Relay Beamforming," *IEEE Wireless Communications Letters*, Vol. 2, Is-sue. 4, pp. 399-402, May 2013.

[97] D. Ponukumati, F. Gao, and C. Xing, "Robust Peer-to-Peer Relay Beamforming A Probabilistic Approach," *IEEE Communications Letters*, Vol. 17, Issue. 2, pp. 305-308, Jan 2013.

[98] P. Ubaidulla; A. Chockalingam, "Robust distributed beamforming for wireless relay networks," *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 2345-2349, Sep 2009.

[99] A. Shaverdian, M. R. Nakhai, "Robust Distributed Beamforming With Interference Coordination in Downlink Cellular Networks," *IEEE Trans. Wireless Commun.*, Vol. 62, Issue. 7, pp. 2411-2421, 2014.

[100] H. Ruan and R. C. de Lamare, "Robust Adaptive Beamforming Based on Low-Rank and Cross-Correlation Techniques," *IEEE Trans. Signal Process.*, Vol. PP, Is-sue. 99, pp. 1, April 2016.

[101] J. V. Stone, "Principal Component Analysis and Factor Analysis," *MIT Press*, Edi-tion. 1, pp. 129-135, 2004.

[102] A. G. Fabregas, A. Martinez and G. Caire, "Bit-Interleaved Coded Modulation," *Foundations and Trends in Communications and Information Theory*, Vol. 5, No. 12, pp 1-153, 2008. http://dx.doi.org/10.1561/0100000019

[103] H. Chen and L. Zhang, "Worst-Case Based Robust Distributed Beamforming for Relay Networks," *Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing*, pp. 4963-4967, May 2013.

[104] S. Salari, M. Z. Amirani, I. Kim, D. Kim and J. Yang, "Worst-Case Based Ro-bust Distributed Beamforming for Relay Networks," *IEEE Trans. Wireless Com-mun.*, Vol. 15, Issue. 6, pp. 4455-4469, March 2016.

[105] Y. Zhang, E. DallAnese and G. B. Giannakis, "Distributed Optimal Beamformers for Cognitive Radios Robust to Channel Uncertainties," *IEEE Trans. Signal Process.*, Vol. 60, Issue. 12, pp. 6495-6508, Sep 2012.

[106] Y. Cao and C. Tellambura, "Joint Distributed Beamforming and Power Allocation in Underlay Cognitive Two-Way Relay Links," *IEEE Trans. Signal Process.*, Vol. 62, Issue. 22, pp. 5950-5961, Sep 2014.

[107] C. Tseng, J. Denis and C. Lin, "On the Robust Design of Adaptive Distributed Beamforming for Wireless SensorRelay Networks," *IEEE Trans. Signal Process.*, Vol. 62, Issue. 13, pp. 3429-3441, May 2014.

[108] P. Hsieh, Y. Lin, and S. Chen, "Robust distributed beamforming design in amplify-and-forward relay systems with multiple user pairs," *23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 371-375, Sep 2015.

[109] Z. L. Yu, Z. Gu, J. Zhou, Y. Li, S. Wee, M. H. Er, "A Robust Adaptive Beamformer Based on Worst-Case Semi-Definite Programming," *IEEE Transactions on Signal Processing*, Vol. 58, No. 11, pp. 5914-5919, 2010.

[110] H. Ruan and R. C. de Lamare, "Joint MSINR and Relay Selection Algorithms for Distributed Beamforming," *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, July 2016.

[111] H. Ruan and R. C. de Lamare, "Joint MMSE Consensus and Relay Selection Algorithms for Distributed Beamforming," *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, July 2016.