Array Signal Processing Algorithms for Beamforming and Direction Finding

This thesis is submitted in partial fulfilment of the requirements for Doctor of Philosophy (Ph.D.)

Lei Wang
Communications Research Group
Department of Electronics
University of York

December 2009

ABSTRACT

Array processing is an area of study devoted to processing the signals received from an antenna array and extracting information of interest. It has played an important role in widespread applications like radar, sonar, and wireless communications. Numerous adaptive array processing algorithms have been reported in the literature in the last several decades. These algorithms, in a general view, exhibit a trade-off between performance and required computational complexity.

In this thesis, we focus on the development of array processing algorithms in the application of beamforming and direction of arrival (DOA) estimation. In the beamformer design, we employ the constrained minimum variance (CMV) and the constrained constant modulus (CCM) criteria to propose full-rank and reduced-rank adaptive algorithms. Specifically, for the full-rank algorithms, we present two low-complexity adaptive step size mechanisms with the CCM criterion for the step size adaptation of the stochastic gradient (SG) algorithms. The convergence and steady-state properties are analysed. Then, the full-rank constrained conjugate gradient (CG) adaptive filtering algorithms are proposed according to the CMV and CCM criteria. We introduce a CG based weight vector to incorporate the constraint in the design criteria for solving the system of equations that arises from each design problem. The proposed algorithms avoid the covariance matrix inversion and provide a trade-off between the complexity and performance.

In reduced-rank array processing, we present CMV and CCM reduced-rank schemes based on joint iterative optimization (JIO) of adaptive filters. This scheme consists a bank of full-rank adaptive filters that forms the transformation matrix, and an adaptive reduced-rank filter that operates at the output of the bank of filters. The transformation matrix and the reduced-rank weight vector are jointly optimized according to the CMV or CCM criteria. For the application of beamforming, we describe the JIO scheme for both the direct-form processor (DFP) and the generalized sidelobe canceller (GSC) structures. For each structure, we derive SG and recursive least squares (RLS) type algorithms to iteratively compute the transformation matrix and the reduced-rank weight vector for the reduced-rank scheme. An auxiliary vector filtering (AVF) algorithm based on the CCM design for robust beamforming is presented. The proposed beamformer decomposes the

adaptive filter into a constrained (reference vector filter) and an unconstrained (auxiliary vector filter) component. The weight vector is iterated by subtracting the scaling auxiliary vector from the reference vector.

For the DOA estimation, the reduced-rank scheme with the minimum variance (MV) power spectral evaluation is introduced. A spatial smoothing (SS) technique is employed in the proposed method to improve the resolution. The proposed DOA estimation algorithms are suitable for large arrays and to deal with direction finding for a small number of snapshots, a large number of users, and without the exact information of the number of sources.

CONTENTS

Acknowledgements

I would like to express my most sincere gratitude to my supervisor, Dr. Rodrigo C. de Lamare, for his help, valuable supervision and useful advice for my research, without which much of this work would not have been possible.

Further thanks go to all members of the Communications Research Group, for their help and support throughout my first year of research.

Finally, my deep gratitude goes to my parents and my wife for their unconditional support, end-less love and encouragement.

Declaration

Some of the research presented in this thesis has resulted in some publications. These publications are listed at the end of Chapter 1.

All work presented in this thesis as original is so, to the best knowledge of the author. References and acknowledgements to other researchers have been given as appropriate.

Glossary

AP	Alternating Projection
ASS	Adaptive Step Size
AVF	Auxiliary Vector Filtering
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CCG	Conventional Conjugate Gradient
CCM	Constrained Constant Modulus
CG	Conjugate Gradient
CM	Constant Modulus
CMV	Constrained Minimum Variance
CRB	Cramér-Rao bound
CS	Cross-Spectral
DFP	Director Form Processor
DOA	Direction of Arrival
EMSE	Excess Mean Square Error
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Technique
FIR	Finite Impulse Response
FR	Full-Rank
FSS	Fixed Step Size
GS	Gram-Schmidt
GSC	Generalized Sildelobe Canceller
JIO	Joint Iterative Optimization
LS	Least Squares
MAI	Multiple Access Interference
MASS	Modified Adaptive Step Size
MCG	Modified Conjugate Gradient
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MSWF	Multistage Wiener Filter
MUSIC	MUltiple SIgnal Classification
MV	Minimum Variance
PC	Principle Component
RGS	Recursive Gram-Schmidt
RLS	Recursive Least Squares
RMSE	Root Mean Square Error
RR	Reduced-Rank
SDMA	Space Division Multiple Access

SG	Stochastic Gradient
SINR	Signal-plus-Interference-to-Noise Ratio
SNR	Signal-to-Noise Ratio
SOI	Signal of Interest
SS	Spatial Smoothing
SVD	Singular Value Decomposition
TAASS	Time Averaging Adaptive Step Size
ULA	Uniform Linear Array

List of Symbols

∇	Gradient
$O(\cdot)$	Complexity
$\mathbb{E}[\cdot]$	Expectation
∞	Infinity
$\mathfrak{R}(\cdot)$	Real part
$\mathfrak{C}(\cdot)$	Imaginary part
Π	Product
$\sum$	Sum
$\|\cdot\|$	Euclidean norm
$\|\cdot\|$	Absolute
trace$(\cdot)$	Trace of a matrix
$\boldsymbol{I}$	Identity matrix
span$(\cdot)$	Span a space
T_k	Chebyshev polynomial of degree k
proj$(\cdot)$	Projection operator
$\mathbb{C}^{m\times q}$	space of complex valued matrices of size m by q
$\mathbb{R}^{m\times q}$	space of real valued matrices of size m by q

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Contents

1.1 Overview

Array processing is an area of signal processing that has powerful tools for extracting information from signals collected using an array of sensors. The information of interest in the signal corresponds to either the content of the signal itself as often found in communications or the location of the source or reflection that produces the signal in radar and sonar systems [1]. These signals propagate spatially through a medium and the wavefront is captured by the sensor array. The sensor array data is processed to extract useful information. Some statistical and adaptive signal processing techniques, including parameter estimation and adaptive filtering (most related topics in the thesis), are extended to sensor array applications.

Array processing finds numerous applications in wireless communications [2,3], radar [4], and sonar [5], and is a promising topic for emerging technologies such as wireless sensor networks [6]. Other applications include seismology, biomedicine, and imaging [7].

1.2 Prior Work

Amongst the most interesting topics of array processing techniques are beamforming and the estimation of the direction of arrival (DOA) of signals, which are widely used in areas that include radar, sonar, acoustics, astronomy, seismology, and communications [8, 9]. Here, we focus on their developments in communications, more specifically, wireless communications, for attenuating interference, improving estimation accuracy, and locating the positions of the sources. To simplify the discussion, we concentrate on uniform linear array (ULA), which composes of a number of identical elements arranged in a single line with uniform spacing. The extension of this material to other array configurations is fairly straightforward in most cases [1] and will be considered as a topic for future investigation.

1.2.1 Beamforming

Generally, an array captures spatially propagating signals arriving from a certain direction and processes them to obtain useful information. To this end, we intend to linearly combine the signals from all the sensors with coefficients in a manner, so as to estimate transmitted data radiating from a specific direction. This operation is known as beamforming [10, 11] since the weighting process emphasizes signals from a particular direction while attenuating those from other directions, which can be regarded as casting or forming a beam. In beamforming, an array processor steers a beam to a certain direction by computing a properly weighted sum of the individual sensor signals just as an finite impulse response (FIR) filter generates an output (at a frequency of interest) that is the weighted sum of time samples. It is convenient to view a beamformer as a frequency-selective filter. Thus, for the beamformer design, filtering techniques can be extended to sensor array applications [12].

According to weighting values are fixed or not, beamformers can be divided into conventional beamformers and adaptive beamformers [12]. Conventional beamformers employ a fixed set of weightings and time-delays to combine the signals from the sensors in the array, primarily using only information about the location of the signal of interest (SOI) relative to the sensor array. However, an array must contend with undesired signals arriving from other directions, which may prevent it from successfully extracting the SOI for which is was designed. Under this condition, the array needs to adjust its response of the received signals for rejecting unwanted signals from other directions. The resulting array is an adaptive array and the corresponding adaptive beamformer updates the weight-

ings by optimizing a certain criterion of performance (subject to various constraints). The adaptive beamformers have better resolution and interference rejection capability than the conventional one [13]. Much effort has been devoted over the past decades to devise adaptive beamformers [14]- [17].

The weighting values in adaptive beamforming are calculated according to optimization of certain criteria [8]. The most promising criteria employed are the constrained minimum variance (CMV) [14] and the constrained constant modulus (CCM) [8] due to their simplicity and effectiveness. The CMV criterion determines the weights by minimizing the beamformer output power, subject to the constraint that the response should be unity in the direction of the SOI. The CCM criterion is a positive measure of the deviation of the beamformer output from a constant modulus value subject to a constraint on the array response of the SOI. The CCM beamformer minimizes the square modulus of the deviation while retaining the gain along the look direction to be constant.

Numerous adaptive filtering algorithms have been employed in beamforming to realize the beamformer design [14]- [18]. Among existing algorithms, the stochastic gradient (SG) is a low-complexity algorithm that employs instantaneous gradient values for iteratively computing weighting values. Its performance is acceptable in many applications. However, its convergence and steady-state behavior depend on the step size and eigenvalue spread of the input covariance matrix. When the eigenvalues are widely spread or/and inappropriate step size values are selected, convergence may be slow and other adaptive algorithms with better convergence characteristics should be considered, e.g., the recursive least squares (RLS) [18]. The RLS algorithm has fast convergence and is independent of the eigenvalue spread of the covariance matrix for stationary inputs. The key problems of this algorithm are high complexity and numerical instability. An alternative method is the CG [19]. It generates iteration (weight) vectors to approximate the optimum, residual vectors corresponding to the iterates, and direction vectors used for updating the iterates and residuals. The algorithm obtains the solution in several iterations. The CG algorithm represents a tradeoff between SG and RLS since it has a faster convergence rate than SG and usually requires lower computational cost when compared with RLS [20].

The adaptive algorithms reviewed above belong to a class that can be called full-rank array processing algorithms. Another class of algorithms that are attracting significant interest is that of reduced-rank algorithms [21]- [25]. For the application of beamforming, reduced-rank schemes project the data received from the sensor array onto a lower dimensional subspace, and calculate the reduced-rank weight vector by minimizing a certain criterion (subject to various constraints) within this subspace for estimating the transmitted

signals. The conventional reduced-rank algorithms include the principle component (PC) and cross-spectral (CS) methods [21]- [24] that rely on an estimate of the signal subspace achieved by the eigen-decomposition of the input data covariance matrix. One of the well-known reduced-rank schemes is the multistage Wiener filter (MSWF) [24, 25]. The process observed by the Wiener filter is first decomposed by a sequence of orthogonal blocking matrices forming an analysis filterbank, whose output is shown to be a process which is characterized by a tridiagonal covariance matrix. The corresponding error-synthesis filterbank is realized by means of a nested chain of scalar Wiener filters. These Wiener filters can be interpreted as well to be a Gram-Schmidt (GS) orthogonalization which results in an error sequence for the successive stages of the decomposed Wiener filter. The MSWF was derived based on the minimum mean squared error (MMSE) criterion [26], and its extensions that utilize the CMV and CCM criteria were reported in [27, 28]. Another technique that resembles the MSWF is the auxiliary-vector filtering (AVF) [29, 30], which utilizes an iterative procedure to compute the weighting values. Its extension to adaptive filtering has been studied in [31].

1.2.2 DOA Estimation

Another important use of array processing techniques is for DOA estimation, that is, given a spatially propagating signal, the determination of its angle of arrival at the array. For the beamformer design, the assumption taken was that the angle of the desired signal relative to the antenna array was known exactly by the beamformer. Commonly, this information is employed in the constraint to make the beamformer steer in this direction. In practice, the actual angle from which the signal arrives is not precisely known. The purpose of the DOA estimation is to determine this angle. There are many DOA estimation algorithms found in the literature, and some of them are described in [1,9,32]. Three main kinds of algorithms are reviewed here, namely, conventional [9,33], subspace-based [34]-[37], and maximum likelihood (ML) [38] methods.

The conventional DOA estimation algorithms steer beams in all possible directions and look for peaks in the output power. The implementation is reasonably simple but always requires a large number of sensor elements to achieve high resolution. The Capon's method [33] is one of the conventional algorithms. It minimizes the output power of the undesired interference while maintaining a constant gain along the look direction according to the minimum variance (MV) criterion subject to a constraint on the array response. By computing and plotting the Capon's spectrum over the possible directions, the DOAs can be estimated by finding the peaks. The estimation accuracy of the Capon's method depends on the number of snapshots and the array size.

Compared with the Capon's algorithm, the subspace-based methods [34, 35] achieve better resolution by exploiting the signal subspace of the input covariance matrix. The well-known methods include the multiple signal classification (MUSIC) [34], the estimation of signal parameters via rotational invariance technique (ESPRIT) [35] reported in the 1980s, the auxiliary vector (AV) [36], and the conjugate gradient (CG) [37] algorithms proposed more recently. MUSIC and ESPRIT algorithms consider the eigen-structure of the input covariance matrix to decompose the observation space into a signal subspace and a corresponding orthogonal noise subspace. MUSIC scans the possible angle directions, plots the power spectrum by making use of this orthogonality, and locates the peaks that correspond to the angles of sources. ESPRIT could reduce the computational requirement without an exhaustive search through all possible directions and achieve a better resolution. It derives its advantages by requiring that the sensor array has a structure that can be decomposed into two equal-sized identical subarrays with the corresponding elements of the two subarrays displaced from each other by a fixed distance. Both MUSIC and ESPRIT suffer from correlated sources. The AV method is developed based on the orthogonality of an extended non-eigenvector signal subspace with the true signal subspace and the scanning vector itself. As the scanning vector drops in the signal subspace, the DOAs are determined by finding the collapse in the extended signal subspace. The CG method can be considered as an extended version of the AV method since it applies the residual vectors of the CG algorithm in place of the AV basis. The AV and CG algorithms exhibit high resolution of uncorrelated and correlated sources with a small number of snapshots and at low signal-to-noise ratio (SNR).

The performance of the ML method [9] is superior to the Capon's and subspace-based methods, especially in low SNR conditions or with a small number of snapshots. It performs well in correlated signal conditions. An attractive approach to simplify the computational complexity is based on an iterative technique referred to as "alternating projection" (AP) [38], that transforms the multivariate nonlinear maximization problem into a sequence of much simpler one-dimensional maximization problems.

1.3 Contributions

In this thesis, I focus on the development of the array processing algorithms for the applications of beamforming and DOA estimation. Note that the applications here assume a scenario with an antenna array at the receiver that is usually found in uplink channels in wireless communications.

In this thesis, we introduce a low-complexity beamformer design based on the SG algorithm that is equipped with variable step size mechanisms. We employ two adaptive step size mechanisms to adapt the beamformer weight vector for the estimation of the transmitted data. The beamformer design is based on the CCM criterion. The characteristics of the proposed adaptive step size SG algorithms are investigated. Another kind of novel algorithms is derived based on the CG technique. The proposed CG-based algorithms enforce the constraint of the CMV and CCM criteria in the system of equations without the matrix inversion and exhibit fast convergence with low-complexity.

We often have to deal with requirements that imply large arrays. However, most full-rank array processing algorithms require a large amount of samples to reach the steady-state when the number of elements in the filter is large. In dynamic scenarios, filters with many elements usually fail or provide poor performance in tracking signals embedded in interference and noise. Reduced-rank techniques were originally motivated to provide a way out of this dilemma. We introduce reduced-rank schemes based on joint iterative optimization (JIO) of filters with the CMV and CCM criteria and compare them with existing reduced-rank methods to show their improved performance in the studied scenarios. Besides, we present a CCM-based AVF algorithm for robust adaptive beamforming. Note that the adaptive beamforming algorithms described here are DOA-based and constrained, which means that the DOA of the SOI is known beforehand by the receiver and the constraint in the design criterion is related to the corresponding array response.

For the DOA estimation, we present a new reduced-rank algorithm based on the MV power spectral evaluation. It is suitable for DOA estimation with large arrays and can be applied to arbitrary array geometries. This algorithm is efficient for problems of direction finding with a large number of sources, and/or without exact information of the number of sources, and does not require the singular value decomposition (SVD).

Specifically, the contributions of this thesis are as follows:

- Two low-complexity SG algorithms with adaptive step size mechanisms are proposed. The algorithms employ the CCM criterion for the beamformer design. A complexity comparison is provided to show their advantages over existing methods. The condition on the step size for the convergence of the mean weight vector is established. The mean and mean-squared values of the step size, in the steady-state condition, are calculated for computation of the excess mean squared error (EMSE) and tracking analysis. The EMSE here considers the effects of additive noise and multiple access interference (MAI) when multiple users are introduced in the system. The energy conservation relation developed in [40]

is exploited in the analysis to simplify the derivation of the recursion for the weight error energy. Simulation experiments are carried out for the stationary and non-stationary scenarios, highlighting the improved performance achieved by the proposed mechanisms in comparison with the fixed step size (FSS) [18] and adaptive step size (ASS) [42, 47] ones.

- Constrained CG adaptive filtering algorithms are proposed for beamforming and provide an attractive tradeoff between the complexity and the performance. The proposed algorithms are derived according to the CMV and CCM criteria. A CG-based weight vector strategy is created for enforcing the constraint and computing the weight expressions. The devised algorithms avoid the matrix inversion and exhibit fast convergence with low complexity. The complexity and the convexity properties of the CCM algorithms are studied, and the convergence analysis of the CG-based weight vector is derived.

- A robust reduced-rank scheme based on joint iterative optimization (JIO) of adaptive filters is presented for the beamformer design. This scheme is designed according to the MV criterion subject to the constraint on the array response of the SOI. It consists of a bank of full-rank adaptive filters that forms the transformation matrix, and an adaptive reduced-rank filter that operates at the output of the bank of filters to estimate the desired signal. The proposed reduced-rank scheme provides an iterative exchange of information between the transformation matrix and the reduced-rank weight vector. We derive SG and RLS type algorithms to compute the transformation matrix and the reduced-rank weight vector. An automatic rank selection technique according to the MV criterion is developed to determine the most adequate rank of the proposed methods. An analysis of the stability and the convergence properties is presented and semi-analytical expressions are given for predicting their performance.

- The JIO reduced-rank scheme is applied to the DOA estimation based on the MV power spectral evaluation. It is specific for the large arrays' condition and can be extended to arbitrary array geometries. We present a constrained RLS algorithm to compute the transformation matrix and the reduced-rank weight vector for calculating the output power of each scanning angle. A spatial smoothing (SS) technique is employed in the LS based method to improve the probability of resolution with highly correlated sources.The proposed algorithms exhibit high resolution for dealing with direction finding with a large number of users and/or without the exact

information of the number of sources. Simulation are conducted to show their performance in different scenarios.

- Considering the fact that the CCM-based beamformer achieves better output performance than the CMV-based one for constant modulus constellations, we present a CCM-based JIO reduced-rank scheme. The transformation matrix and the reduced-rank weight vector are jointly optimized according to the CM criterion subject to different constraints. We describe the proposed scheme for both the direct-form processor (DFP) and the generalized sidelobe canceller (GSC) structures. For each structure, we derive the SG and RLS algorithms. The Gram-Schmidt (GS) technique is applied to reformulate the transformation matrix and improve performance. An automatic rank selection technique is developed according to the CM criterion. The complexity and convexity analyses of the proposed methods are carried out.

- An auxiliary vector filtering algorithm with the CCM criterion is introduced for beamforming. This algorithm utilizes an iterative way to compute the weight vector for estimating the transmitted signal. It provides an efficient way to deal with filters with a large number of elements and shows superior performance under severe scenarios, e.g., steering vector mismatch.

1.4 Thesis Outline

The structure of the thesis is as follows:

- In Chapter 2, a system model and design criteria are introduced for beamforming and DOA estimation. Several assumptions are given to facilitate the system model for the development of the proposed algorithms. Based on this system model, low-complexity adaptive step size mechanisms are introduced for the update of the step size in the SG algorithm. The beamformer design according to the CCM criterion is detailed in this chapter. Characteristics of the proposed algorithms are given and analytical expressions are developed to predict their performance.

- In Chapter 3, the conventional CG algorithm is reviewed. We develop modified versions of the CG algorithm for beamforming according to the CMV and CCM

criteria. The convergence properties are analyzed.

- In Chapter 4, a reduced-rank scheme based on joint iterative optimization of filters with the CMV criterion is proposed for beamforming. The SG and RLS type algorithms are described. The properties of the proposed methods are analyzed. Besides, a DOA estimation algorithm based on the proposed reduced-rank scheme is developed. An SS technique is employed to increase the probability of resolution.

- In Chapter 5, a CCM-based reduced-rank scheme is introduced. Two structures are investigated for the realization of the reduced-rank scheme. The GS and automatic rank selection techniques are combined in the proposed method for further enhancing the performance. A CCM-based auxiliary vector filtering algorithm is described. It provides an iterative way to compute the weight vector and exhibits a good performance in the studied scenarios.

- In Chapter 6, conclusions and a discussion on possibilities for the future work are presented.

1.5 Notation

In this thesis, small and capital boldface letters are used to denotes vectors and matrices, e.g., $\boldsymbol{x}$ and $\boldsymbol{R}$, respectively. Elements of the vector and matrix are denoted as x_j and $R_{k,l}$. With no specific explanation, all vectors are column vectors. The symbol $\boldsymbol{I}$ denotes the identity matrix of appropriate dimensions, and the boldface $\mathbf{0}$ denotes either a zero vector or a zero matrix. The notation $\|\boldsymbol{x}\|$ denotes the Euclidean norm of a vector. The variable i is used as a time index, e.g., $\boldsymbol{x}(i)$ is the vector $\boldsymbol{x}$ at time instant i. $\mathfrak{R}(\cdot)$ and $\mathfrak{J}(\cdot)$ denote the real and imaginary components of a complex number, respectively. The symbol $*$ denotes complex conjugate (for scalars), $(\cdot)^T$ denotes transpose, and $(\cdot)^H$ Hermitian transpose. The symbol $\mathbb{E}[\cdot]$ denotes the statistical expectation operator.

1.6 List of Publications

Some of the research presented in this thesis has been published, submitted, or will be submitted for publication at the time of submission of this thesis.

Journal Papers

1. L Wang, R. C. de Lamare, and Y. L. Cai, "Low-complexity adaptive step size constrained constant modulus SG algorithms for adaptive beamforming," *Elsevier Signal Processing*, vol. 89, pp. 2503-2513, Dec. 2009.

2. L Wang and R. C. de Lamare, "Constrained adaptive filtering algorithms based on conjugate gradient techniques for beamforming," submitted to *IET Signal Processing*, 2009.

3. R. C. de Lamare, L. Wang, and R. Fa, "Adaptive reduced-rank LCMV beamforming algorithms based on joint iterative optimization of filters: design and analysis," (accepted) *Elsevier Signal Processing*, 2009.

4. L. Wang, R. C. de Lamare, and M. Yukawa, "Adaptive reduced-rank constrained constant modulus algorithms based on joint iterative optimization of filters for beamforming," (under review in the 2nd round) *IEEE Trans. Signal Processing*, 2009.

5. L. Wang and R. C. de Lamare, "Robust auxiliary vector filtering algorithm for adaptive beamforming," submitted to *IEEE Trans. Signal Processing*.

6. L. Wang, R. C. de Lamare, and M. Haardt, "Reduced-rank DOA estimation algorithms based on joint iterative optimization," (under preparation) for *IEEE Trans. Signal Processing*.

Conference Papers

1. L. Wang and R. C. de Lamare, "Constrained constant modulus RLS-based blind adaptive beamforming algorithm for smart antennas," *4th International Symposium on Wireless Communication Systems 2007*, pp. 657-661, Oct. 2007.

2. L. Wang and R. C. de Lamare, "A novel constrained adaptive algorithm using the conjugate gradient method for smart antennas," *41st Asilomar Conference on Signals, Systems and Computers*, pp. 2243-2247, Nov. 2007.

3. L. Wang, Y. L. Cai, and R. C. de Lamare, "Low-complexity adaptive step size constrained constant modulus sg-based algorithms for blind adaptive beamforming," *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008*, pp. 2593-2596, Mar. 2008.

4. L. Wang and R. C. de Lamare, "A new approach to reduced-rank DOA estimation based on joint iterative subspace optimization and grid search," *16th International Conference on Digital Signal Processing*, pp. 1-6, July 2009.

5. L. Wang and R. C. de Lamare, "Adaptive reduced-rank constrained constant modulus beamforming algorithms based on joint iterative optimization of filters," (will have been presented) *2009 IEEE Workshop on Statistical Signal Processing*, 2009.

6. L. Wang and R. C. de Lamare, "Robust auxiliary vector filtering algorithm based on constrained constant modulus design for adaptive beamforming," submitted to *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2010.

7. L. Wang, R. C. de Lamare, and M. Haardt, "Reduced-rank DOA estimation based on joint iterative subspace recursive optimization and grid search," submitted to *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2010.

2. ADAPTIVE STEP SIZE CCM SG ALGORITHMS FOR ADAPTIVE BEAMFORMING

Contents

2.1 Introduction

We categorize this chapter into two parts. The first part presents a system model of the antenna array at the receiver. We will see that the system structures applied to beamformer design and DOA estimation are similar. The main difference is that, for adaptive beamforming, weighting values are adapted with the received data.

Based on the system model, we introduce the adaptive algorithms for beamforming in the second part. Adaptive beamforming is employed widely in communication systems, such as spatial-division multiple access (SDMA) systems [11, 44]. Numerous array processing algorithms haven been developed for the beamformer design [8, 45, 46]. The SG is a low-complexity method for iteratively computing the weight vector to generate the beamformer output. Its performance is sensitive to the step size, which has to be adjusted to make a compromise between fast convergence and small misadjustment. Adaptive step size (ASS) mechanisms [42], [47] were employed to circumvent these problems. However, they cannot yield both fast tracking as well as small misadjustment with simple implementation.

We propose two low-complexity adaptive step size mechanisms for use with SG algorithms designed according to the CCM criterion. The origin of these mechanisms can be traced back to the works [48] and [49], where the algorithms were proposed with the MMSE criterion. The algorithms according to the CMV criterion were reported in [50]. We extend these mechanisms to the CCM criterion in consideration of its superior performance over the CMV for constant modulus constellations.

The rest of this chapter is organized as follows: the array structure is introduced in Section 2.2, and the system model is described in Section 2.3. The adaptive SG algorithm based on the CCM beamformer design is introduced in Section 2.4. Section 2.5 presents the developed mechanisms for the SG method. Section 2.6 is dedicated to the convergence, steady-state and tracking analyses of the proposed algorithms. Simulation results are provided and discussed in Section 2.7, and conclusions are drawn in Section 2.8.

2.2 Array Structure

2.2.1 Adaptive Array Structure

Adaptive array is an efficient antenna structure to realize beamforming, as shown in Fig. 2.1. This is an uniform linear array (ULA) that consists of m sensor elements. The sample received at each element is x_j, where $j = 1, \ldots, m$, and the corresponding weighting value is w_j. The array processor processes the received samples to generate the output that is an estimate of the transmitted desired signal. The output signal passes through a feedback structure to get a reference signal, which is sent back to the adaptive processor for the weights' update. The weight adaptation is performed by optimizing a cost function, which is also followed by the generation of the reference signal. Commonly, this reference is a deviation between the estimated desired signal and an assistant information, e.g., a constant modulus condition in CM criterion. It is important to let the adaptive array processing algorithm control the weights following the change of the received data and maximize the quality of the estimated desired signal. In working with antenna array, it is very convenient to make use of vector notation, i.e., $\boldsymbol{x} = [x_1, \ldots, x_m]^T$ and $\boldsymbol{w} = [w_1, \ldots, w_m]^T$ denote the received data vector and weight vector, respectively.

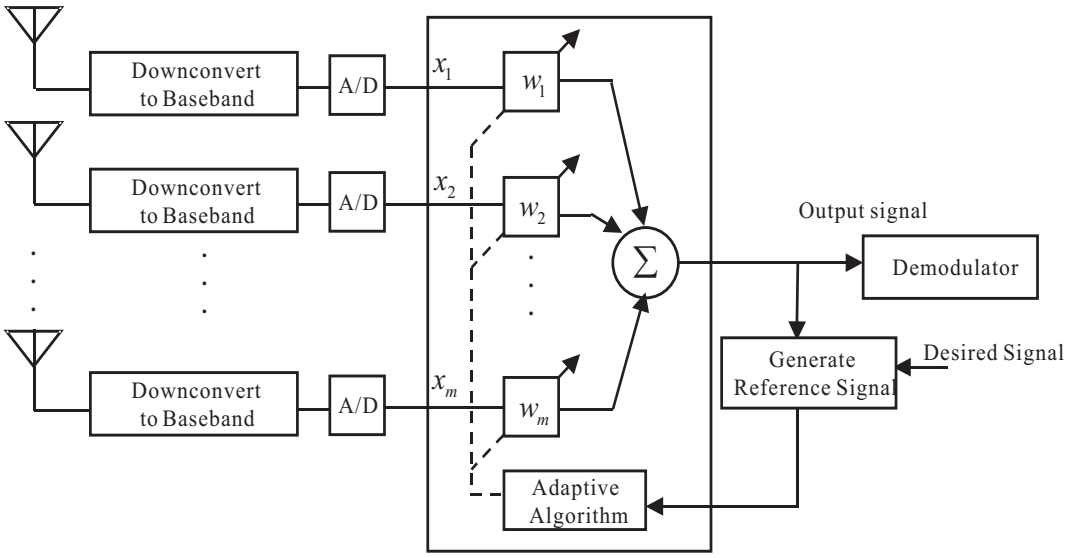

Fig. 2.1: An adaptive array structure.

2.2.2 Antenna Array and Direction of Arrival

The direction of arrival (DOA) of the source signal of the antenna array can be determined by two angles, horizontal and azimuthal, which is illustrated in Fig.2.2. It is clear that the direction of the incident wave can be located by the horizonal angle θ and the azimuthal angle ϕ together. Thus, the DOA is a function related to them, i.e., $f(\theta, \phi)$. For simplicity, unless otherwise noted, it is assumed that the incident wave arrives at the array in the horizontal plane, $\phi = \pi/2$, so that the azimuthal direction θ completely specifies the DOA.

2.3 System Model

In Fig. 2.3, we describe a generic system model applied to beamforming and the DOA estimation. It is essential to make two assumptions [9]:

- The propagating signals are assumed to be produced by point sources; that is, the size of the source is small with respect to the distance between the source and the

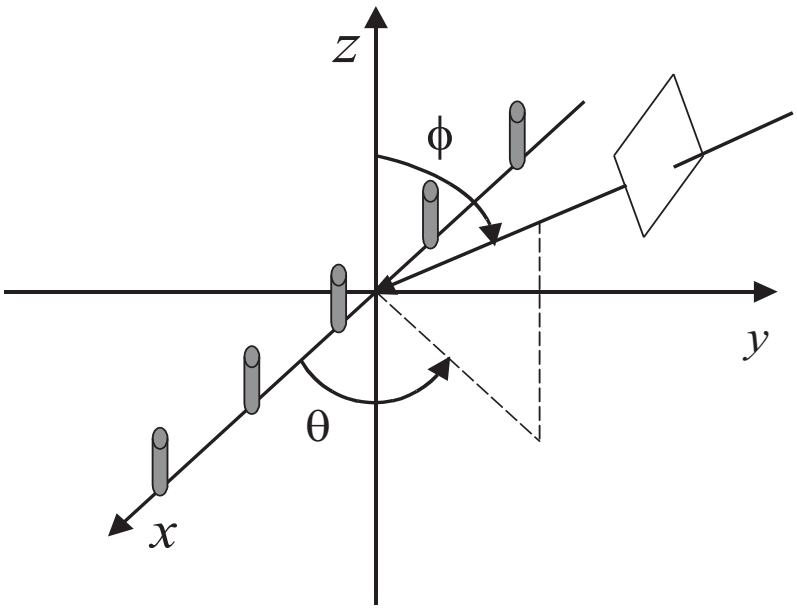

Fig. 2.2: A linear equally spaced array oriented along the axis receives a plane wave from direction (θ, ϕ).

array at the receiver end.

- The sources are assumed to be in the far field, so that the spherically propagating wave can be reasonably approximated with a plane wave.

Now, let us suppose that q narrowband signals impinge on a ULA of $m(m \geq q)$ sensor elements. The DOAs of the sources are $\theta_0, \ldots, \theta_{q-1}$. The received vector $\boldsymbol{x}(i) \in \mathbb{C}^{m\times 1}$ at the ith snapshot (time instant) can be modeled as

$$\boldsymbol{x}(i) = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{s}(i) + \boldsymbol{n}(i), \quad i = 1, \ldots, N, \tag{2.1}$$

where $\boldsymbol{\theta} = [\theta_0, \ldots, \theta_{q-1}]^T \in \mathbb{R}^{q\times 1}$ contains the DOAs of the sources, $\boldsymbol{A}(\boldsymbol{\theta}) = [\boldsymbol{a}(\theta_0), \ldots, \boldsymbol{a}(\theta_{q-1})] \in \mathbb{C}^{m\times q}$ comprises the steering vectors $\boldsymbol{a}(\theta_k) = [1, e^{-2\pi j \frac{\iota}{\lambda_c} cos\theta_k}, \ldots, e^{-2\pi j(m-1)\frac{\iota}{\lambda_c} cos\theta_k}]^T \in \mathbb{C}^{m\times 1}, (k = 0, \ldots, q-1)$, where λ_c is the wavelength and ι ($\iota = \lambda_c/2$ in general) is the inter-element distance of the ULA. In order to avoid mathematical ambiguities, the steering vectors $\boldsymbol{a}(\theta_k)$ are assumed to be linearly independent, $\boldsymbol{s}(i) \in \mathbb{C}^{q\times 1}$ contains the source data, $\boldsymbol{n}(i) \in \mathbb{C}^{m\times 1}$ is temporally and spatially white sensor noise, which is assumed to be a zero-mean and Gaussian process, and N is the observation size given in snapshots.

The output is given by

$$y(i) = \boldsymbol{w}^H(i)\boldsymbol{x}(i), \tag{2.2}$$

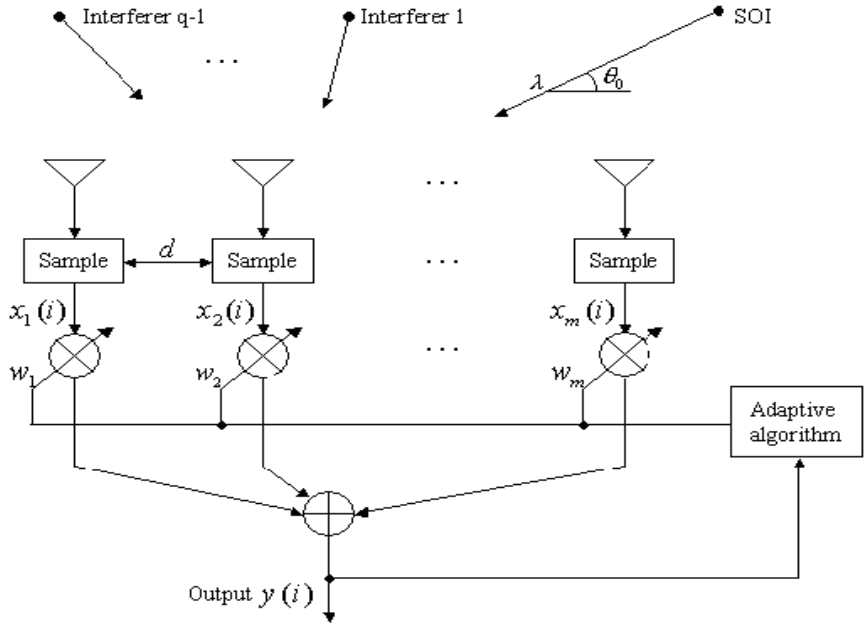

Fig. 2.3: Adaptive ULA structure.

where $\boldsymbol{w}(i) = [w_1(i), \ldots, w_m(i)]^T \in \mathbb{C}^{m\times 1}$ is the complex weight vector.

The feedback structure is controlled by the adaptive array processing algorithm derived according to the design criterion. It updates the weight vector to generate the output $y(i)$. For adaptive beamforming, this part is responsible for rapidly decreasing the deviation of the beamformer output from the desired signal and improving the accuracy of estimation.

Considering the importance of the design criteria in the development of the array processing algorithms, we interpret the CMV and CCM criteria for convenience of the presentation of the current and following chapters. The CMV criterion updates $\boldsymbol{w}(i)$ by solving the following optimization problem [32]:

$$\min \quad J_{\text{mv}}(\boldsymbol{w}(i)) = \boldsymbol{w}(i)^H \boldsymbol{R}\boldsymbol{w}(i), \quad \text{subject to} \quad \boldsymbol{w}^H(i)\boldsymbol{a}(\theta_0) = \gamma, \tag{2.3}$$

where $\boldsymbol{R} = E[\boldsymbol{x}(i)\boldsymbol{x}^H(i)] \in \mathbb{C}^{m\times m}$ is the input covariance matrix, γ is a constant value, and $\boldsymbol{a}(\theta_0)$ is the steering vector (array response) of the SOI, which can be obtained by employing DOA estimation algorithms.

The CCM criterion determines $\boldsymbol{w}(i)$ by solving the following optimization problem:

$$\min \quad J_{\text{cm}}(\boldsymbol{w}(i)) = \mathbb{E}\big[(|y(i)|^p - \nu)^2\big], \quad \text{subject to} \quad \boldsymbol{w}^H(i)\boldsymbol{a}(\theta_0) = \gamma, \tag{2.4}$$

where ν is suitably chosen to guarantee that the weight solution is close to the global minimum and γ is set to ensure the convexity of (2.4) [28]. In general, $p = 2$ [8] is selected to consider the cost function as the expected deviation of the squared modulus of the beamformer output to a constant, say $\nu = 1$. This deviation provides information for the parameter estimation in the beamformer design. The beamformer minimizes the contribution of interference and noise while maintaining the gain along the look direction to be constant.

2.4 Adaptive SG Algorithm

Considering the system model and the CCM design criterion introduced in Section 2.3, the constraint in (2.4) can be incorporated by the method of Lagrange multipliers [18] in the form

$$J_{\text{CCM}}\big(\boldsymbol{w}(i)\big) = \mathbb{E}[(|y(i)|^2 - 1)^2] + 2\,\Re\big\{\lambda[\boldsymbol{w}^H(i)\boldsymbol{a}(\theta_0) - 1]\big\}, \tag{2.5}$$

where λ is a scalar Lagrange multiplier and $\gamma = 1$. The solution can be obtained by taking the instantaneous gradient of (2.5) of $\boldsymbol{w}^*(i)$, setting it equal to a null vector, and using the constraint. Thus, the weight update of the adaptive SG algorithm is given by

$$\boldsymbol{w}(i+1) = \boldsymbol{w}(i) - \mu(i)\nabla\boldsymbol{w}(i), \tag{2.6}$$

where $\nabla\boldsymbol{w}(i) = e(i)y^*(i)[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)]\boldsymbol{x}(i)$, and $e(i) = |y(i)|^2 - 1$. The coefficient $\mu(i)$ is the step size, which is constant for the fixed step size (FSS) [52] and a variable value for the ASS. Note that $e(i)$ depends on $y(i)$, which is a function of current weight vector $\boldsymbol{w}(i)$. The solution is obtained by initializing $\boldsymbol{w}(i)$ and estimating a priori $y(i)$ to start the iteration. For the FSS algorithm, μ is predetermined to make a compromise between fast convergence rate and small misadjustment. Its performance deteriorates in time-varying channels. The ASS methods cannot achieve both fast tracking and small misadjustment, and their complexity is proportional to the number of sensor elements m, which makes a high additional computational load for large arrays.

2.5 Proposed Adaptive Step Size Mechanisms

In this section, two adaptive step size mechanisms according to the CCM criterion are introduced for adjusting the step size. The developed SG algorithms can effectively

adjust step size values with low-complexity while maintaining fast tracking as well as small misadjustment.

2.5.1 Modified Adaptive Step Size (MASS)

Motivated by the algorithm proposed in [48], the first developed algorithm adjusts the step size that is controlled by the square of the prediction error, i.e.,

$$\mu(i+1) = \alpha\mu(i) + \tau e^2(i), \tag{2.7}$$

where $e(i) = |y(i)|^2 - 1$, $0 < \alpha < 1$, and $\tau > 0$ is an independent variable for controlling the prediction error. The rationale for this modified adaptive step size (MASS) is that a large prediction error will increase the step size to provide faster convergence rate while a small prediction error will result in a decrease in the step size to yield smaller misadjustment. Note that the step size $\mu(i+1)$ should be restricted in a range as follows

$$\mu(i+1) = \begin{cases} \mu_{\max} & \text{if } \mu(i+1) > \mu_{\max} \\ \mu_{\min} & \text{if } \mu(i+1) < \mu_{\min} \\ \mu(i+1) & \text{otherwise} \end{cases}, \tag{2.8}$$

where $0 < \mu_{\min} < \mu_{\max}$. The constant $\mu_{\min}$ is chosen as a compromise between the desired level of steady-state misadjustment and the required minimum level of tracking ability while $\mu_{\max}$ is normally selected close to the point of instability of the algorithm for providing the maximum convergence speed. Typical values of $\mu_{\min}$ and $\mu_{\max}$ that were found to work well in simulations are around 10^{-6} and 10^{-3}, respectively. They are nearly invariant under different scenarios.

2.5.2 Time Averaging Adaptive Step Size (TAASS)

Motivated by the robust variable step size algorithm proposed in [49], the second developed algorithm employs a time average estimate of the correlation of $e(i)$ and $e(i-1)$, which is given by

$$v(i) = \beta v(i-1) + (1-\beta)e(i)e(i-1), \tag{2.9}$$

and the update rule is

$$\mu(i+1) = \alpha\mu(i) + \tau v^2(i), \tag{2.10}$$

where the limits on $\mu(i+1)$, α and τ are similar to those of the MASS mechanism. The exponential weighting parameter $\beta(0 < \beta < 1)$ governs the averaging time constant.

In stationary environments, β should be close to 1 for saving information contained in previous samples that is relevant to determining a measure of adaptation state. For non-stationary environments, the time averaging window should be small for forgetting past data and leaving space for the current statistics adaptation, so, $\beta < 1$. Hereafter this is called as time averaging adaptive step size (TAASS). At the early stage, the error correlation estimate $v^2(i)$ is large and so $\mu(i)$ is large to increase the convergence rate. As it approaches the optimum, $v^2(i)$ is very small, resulting in a small step size for keeping low misadjustment near optimum.

The origin of (2.10) can be traced back to the work in [49] for the MMSE criterion and developed in [50] for the CMV criterion. Here, we extend it to the SG algorithms with the CCM criterion since they achieve better performance than the CMV-based methods for constant modulus constellations. The update rule in (2.10) is in accordance with the CCM criterion since it estimates the time average of the autocorrelation between $e(i)$ and $e(i-1)$, which is the error between the square of the array output and the unit modulus. The MASS mechanism uses the instantaneous square error to control the step size and make a tradeoff between the slow convergence and misadjustment. However, the step size adaptation and the misadjustment are affected by the noise [49]. The advantage of the TAASS mechanism is that it employs the error autocorrelation to estimate the time average of adjacent error terms, which include more information about adaptation. In this way, the algorithm can effectively maintain a reasonable immunity to uncorrelated additive noise. The objective is to operate in a large $\mu(i)$ when the algorithm works far from the optimum with $\mu(i)$ reducing as it approaches the optimum.

2.5.3 Computational Complexity

The complexity considered here is the additional complexity, which includes additions and multiplications required for the step size adaptation. The CCM-SG algorithm [51] works without additional complexity since there is no step size adaptation. The complexity of the ASS algorithm in [42] is a linear monotonic increasing function of the number of sensor elements m, i.e., $5m-1$ for additions and $4m+3$ for multiplications. It increases the computational load if the array size is large. The proposed MASS and TAASS mechanisms only require a small fixed number of operations for adjusting the step size. Specifically, the complexity of the MASS is 1 addition and 3 multiplications, and of the TAASS is 2 additions and 6 multiplications.

2.6 Analysis of the Proposed Algorithms

In this section, we investigate characteristics of the proposed algorithms. Note that the analysis here is derived according to the MMSE criterion, i.e., $\mathbb{E}[|e(i)|^2] = \mathbb{E}[|d(i) - y(i)|^2]$, where $d(i)$ is the desired response of the beamformer [18]. The analysis is suitable to the proposed algorithms with the CCM criterion since the global minimum of the CCM algorithm roughly correspond to the MMSE. This fact was conjectured in [53] and proved in [54]. A sufficient condition for the convergence of the mean weight vector is studied. Then, the steady-state mean and the mean-square expressions of the step size are derived. On the basis of these step size expressions, the steady-state and tracking analyses are provided by employing the energy conservation relation originally proposed in [40]. The effects of multiple access interference (MAI) and additive noise are considered in the analysis. To facilitate the analysis, the following approximations and assumptions are taken into account.

Approximation:

i) $\lim_{i\to\infty} \mathbb{E}\big[|e(i)|^2\big] = \xi_{\text{min}} + \xi_{\text{ex}}(\infty)$. Here, $\xi_{\text{min}} = |d(i)|^2 - \boldsymbol{p}^H \boldsymbol{R}^{-1} \boldsymbol{p}$ [18] is the MMSE and $\xi_{\text{ex}}(\infty)$ is the excess mean square error (EMSE) associated with the optimization problem. The term $\boldsymbol{p} = \mathbb{E}\big[\boldsymbol{x}(i)d^*(i)\big]$ is the cross correlation between the received vector $\boldsymbol{x}(i)$ and the desired response $d(i)$, and $\boldsymbol{R} = \mathbb{E}\big[\boldsymbol{x}(i)\boldsymbol{x}^H(i)\big]$ is the covariance matrix of the received vector.

Assumptions:

i) The variance of $\mu(i)$ is very small. This assumption is approximately true if τ is small and α close to one.

ii) $(\xi_{\text{min}}+\xi_{\text{ex}}(\infty)) \approx \xi_{\text{min}}$ and $(\xi_{\text{min}}+\xi_{\text{ex}}(\infty))^2 \approx \xi_{\text{min}}^2$ if $\xi_{\text{min}} \gg \xi_{\text{ex}}(\infty)$. This assumption is clear since the term $\xi_{\text{ex}}(\infty)$ can be ignored in the steady-state environment.

iii) The additive noise $n(i)$ is independent and identically distributed (i.i.d.) and statically independent of the received vector $\boldsymbol{x}(i)$. This is a realistic assumption that was made in [55].

2.6.1 Convergence Analysis

Sufficient Condition for the Convergence of the Mean Weight Vector

In view of (2.6), the convergence of the weight vector is determined by two factors, namely, the step size parameter $\mu(i)$ and the received vector $\boldsymbol{x}(i)$. Taking expectation of $\mu(i)\nabla\boldsymbol{w}(i)$ in (2.6) and applying assumption i), we have

$$\mathbb{E}\big\{\mu(i)e(i)y^*(i)\big[\boldsymbol{I}-\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{x}(i)\big\} = \mathbb{E}\big[\mu(i)\big]\mathbb{E}\big\{e(i)y^*(i)\big[\boldsymbol{I}-\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{x}(i)\big\}, \tag{2.11}$$

and

$$\mathbb{E}\big[\mu(i)e(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\big]\boldsymbol{w}(i) = \mathbb{E}\big[\mu(i)\big]\boldsymbol{R}_{\text{CCM}}\boldsymbol{w}(i), \tag{2.12}$$

where $\boldsymbol{R}_{\text{CCM}} = \mathbb{E}\big[e(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\big] \in \mathbb{C}^{m\times m}$.

Equation (2.6) can be written in an alternative way as

$$\boldsymbol{w}(i+1) = \big[\boldsymbol{I} - \mu(i)e(i)\boldsymbol{u}(i)\boldsymbol{x}^H(i)\big]\boldsymbol{w}(i), \tag{2.13}$$

where $\boldsymbol{u}(i) = \big[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{x}(i) \in \mathbb{C}^{m\times 1}$.

Now, we define the weight error vector $\tilde{\boldsymbol{w}}(i)$ and substitute (2.13) into the expression, which becomes

$$\begin{aligned}\tilde{\boldsymbol{w}}(i+1) &= \boldsymbol{w}_{\text{opt}} - \boldsymbol{w}(i+1)\\ &= \big[\boldsymbol{I} - \mu(i)e(i)\boldsymbol{u}(i)\boldsymbol{x}^H(i)\big]\tilde{\boldsymbol{w}}(i) + \mu(i)e(i)\boldsymbol{u}(i)\boldsymbol{x}^H(i)\boldsymbol{w}_{\text{opt}},\end{aligned} \tag{2.14}$$

where $\boldsymbol{w}_{\text{opt}}$ denotes the optimal weight solution. By employing assumption i) and taking expectations on both sides of (2.14), we have

$$\mathbb{E}\big[\tilde{\boldsymbol{w}}(i+1)\big] = \big\{\boldsymbol{I} - \mathbb{E}\big[\mu(i)\big]\boldsymbol{R}_{\text{ux}}\big\}\mathbb{E}\big[\tilde{\boldsymbol{w}}(i)\big], \tag{2.15}$$

where $\boldsymbol{R}_{\text{ux}} = \mathbb{E}\big[e(i)\boldsymbol{u}(i)\boldsymbol{x}^H(i)\big] = \big[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{R}_{\text{CCM}}$ and $\boldsymbol{R}_{\text{ux}}\boldsymbol{w}_{\text{opt}} = \mathbb{E}[e(i)(\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0))d^*(i)\boldsymbol{x}(i)] = \boldsymbol{0}$ [50]. Therefore, $\mathbb{E}\big[\boldsymbol{w}(i)\big] \rightarrow \boldsymbol{w}_{\text{opt}}$ or equivalently, $\lim_{i\rightarrow\infty}\mathbb{E}[\tilde{\boldsymbol{w}}(i)] = \boldsymbol{0}$ represents the stable condition if and only if $\prod_{i=0}^{\infty}\big\{\boldsymbol{I} - \mathbb{E}\big[\mu(i)\big]\boldsymbol{R}_{\text{ux}}\big\} \rightarrow \boldsymbol{0}$. The sufficient condition for the convergence of the mean weight vector to hold implies that

$$0 \leq \mathbb{E}\big[\mu(\infty)\big] \leq \frac{2}{\lambda_{\text{max}}^{\text{ux}}}, \tag{2.16}$$

where $\lambda_{\text{max}}^{\text{ux}}$ is the maximum eigenvalue of $\boldsymbol{R}_{\text{ux}}$. This condition ensures the convergence of the mean weight vector for both the MASS and TAASS algorithms.

Steady-State Step Size Value for MASS

Employing assumption i) and taking expectations on both sides of (2.7), the first-moment steady-state step size for the MASS algorithm becomes

$$\mathbb{E}\big[\mu(i+1)\big] = \alpha\mathbb{E}\big[\mu(i)\big] + \tau\mathbb{E}\big[e^2(i)\big]. \tag{2.17}$$

Also, by squaring (2.7) and taking expectations, the second-moment steady-state step size can be written as

$$\begin{aligned}\mathbb{E}\big[\mu^2(i+1)\big] &= \alpha^2\mathbb{E}\big[\mu^2(i)\big] + 2\alpha\tau\mathbb{E}\big[\mu(i)\big]\mathbb{E}\big[e^2(i)\big] + \tau^2\mathbb{E}\big[e^4(i)\big] \\ &\approx \alpha^2\mathbb{E}\big[\mu^2(i)\big] + 2\alpha\tau\mathbb{E}\big[\mu(i)\big]\mathbb{E}\big[e^2(i)\big],\end{aligned} \tag{2.18}$$

where the term $\tau^2\mathbb{E}\big[e^4(i)\big]$ in (2.18) is negligible as compared to the other terms since τ^2 and $\mathbb{E}\big[e^4(i)\big]$ are relatively small in the steady-state environment.

In the steady-state, the relations $\lim_{i\to\infty}\mathbb{E}[\mu(i)] = \lim_{i\to\infty}\mathbb{E}[\mu(i+1)] = \mathbb{E}[\mu(\infty)]$ and $\lim_{i\to\infty}\mathbb{E}[\mu^2(i)] = \lim_{i\to\infty}\mathbb{E}[\mu^2(i+1)] = \mathbb{E}[\mu^2(\infty)]$ hold. Applying approximation i) to (2.17) and (2.18), we obtain

$$\mathbb{E}\big[\mu(\infty)\big] = \frac{\tau\big[\xi_{\text{min}} + \xi_{\text{ex}}(\infty)\big]}{1-\alpha}, \tag{2.19}$$

$$\mathbb{E}\big[\mu^2(\infty)\big] = \frac{2\alpha\tau^2\big[\xi_{\text{min}} + \xi_{\text{ex}}(\infty)\big]^2}{(1-\alpha)^2(1+\alpha)}. \tag{2.20}$$

The expressions (2.19) and (2.20) can be further simplified if we consider assumption ii), yielding

$$\mathbb{E}\big[\mu(\infty)\big] \approx \frac{\tau\xi_{\text{min}}}{1-\alpha}, \tag{2.21}$$

$$\mathbb{E}\big[\mu^2(\infty)\big] \approx \frac{2\alpha\tau^2\xi_{\text{min}}^2}{(1-\alpha)^2(1+\alpha)}. \tag{2.22}$$

Note that (2.21) and (2.22) are employed in the analysis of the EMSE for the proposed MASS algorithm.

Steady-State Step Size Value for TAASS

Applying assumption i) to (2.10) and taking expectations on both sides, the first-order steady-state step size for the TAASS algorithm is expressed

$$\mathbb{E}\big[\mu(i+1)\big] = \alpha\mathbb{E}\big[\mu(i)\big] + \tau\mathbb{E}\big[v^2(i)\big]. \tag{2.23}$$

Also, by squaring (2.10) and taking expectations, the second-order steady-state step size is

$$\begin{aligned}\mathbb{E}\big[\mu^2(i+1)\big] &= \alpha^2\mathbb{E}\big[\mu^2(i)\big] + 2\alpha\tau\mathbb{E}\big[\mu(i)\big]\mathbb{E}\big[v^2(i)\big] + \tau^2\mathbb{E}\big[v^4(i)\big] \\ &\approx \alpha^2\mathbb{E}\big[\mu^2(i)\big] + 2\alpha\tau\mathbb{E}\big[\mu(i)\big]\mathbb{E}\big[v^2(i)\big],\end{aligned} \tag{2.24}$$

where the term $\tau^2\mathbb{E}\big[e^4(i)\big]$ in (2.24) is negligible.

According to (2.9), the time average estimate $v(i)$ can be written in a recursive form as

$$v(i) = (1-\beta)\sum_{n=0}^{i-1}\beta^n e(i-n)e(i-n-1), \tag{2.25}$$

and

$$v^2(i) = (1-\beta)^2\sum_{n=0}^{i-1}\sum_{j=0}^{i-1}\beta^n\beta^j e(i-n)e(i-n-1)e^*(i-j)e^*(i-j-1). \tag{2.26}$$

In the steady-state condition, we assume that the proposed algorithm has converged. In this case, we have $\mathbb{E}\big[e(i-n)e^*(i-j)\big] = 0\ \forall n \neq j$. Hence, expectation of (2.26) can be simplified

$$\mathbb{E}\big[v^2(i)\big] = (1-\beta)^2\sum_{n=0}^{i-1}\beta^{2n}\mathbb{E}\big[|e(i-n)|^2\big]\mathbb{E}\big[|e(i-n-1)|^2\big]. \tag{2.27}$$

Applying approximation i) and assumption ii) to (2.27) and employing $\lim_{i\to\infty}\mathbb{E}[v^2(i)] = \mathbb{E}[v^2(\infty)]$, it brings

$$\mathbb{E}\big[v^2(i)\big] = \frac{(1-\beta)\big[\xi_{\text{min}} + \xi_{\text{ex}}(\infty)\big]^2}{1+\beta} \approx \frac{(1-\beta)\xi_{\text{min}}^2}{1+\beta}, \tag{2.28}$$

where the derivation is given in Appendix A.

Substituting (2.28) into (2.23) and (2.24), respectively, and recalling the relations $\lim_{i\to\infty}\mathbb{E}[\mu(i)] = \lim_{i\to\infty}\mathbb{E}[\mu(i+1)] = \mathbb{E}[\mu(\infty)]$ and $\lim_{i\to\infty}\mathbb{E}[\mu^2(i)] = \lim_{i\to\infty}\mathbb{E}[\mu^2(i+1)] = \mathbb{E}[\mu^2(\infty)]$, yield

$$\mathbb{E}[\mu(\infty)] \approx \frac{\tau(1-\beta)\left[\xi_{\text{min}} + \xi_{\text{ex}}(\infty)\right]^2}{(1-\alpha)(1+\beta)}, \tag{2.29}$$

$$\mathbb{E}[\mu^2(\infty)] \approx \frac{2\alpha\tau^2(1-\beta)^2\left[\xi_{\text{min}} + \xi_{\text{ex}}(\infty)\right]^4}{(1+\alpha)(1-\alpha)^2(1+\beta)^2}. \tag{2.30}$$

We can use assumption ii) to develop $\left[\xi_{\text{min}} + \xi_{\text{ex}}(\infty)\right]^4 \approx \xi_{\text{min}}^4$. Thus, from (2.29) and (2.30)

$$\mathbb{E}[\mu(\infty)] \approx \frac{\tau(1-\beta)\xi_{\text{min}}^2}{(1-\alpha)(1+\beta)}, \tag{2.31}$$

$$\mathbb{E}[\mu^2(\infty)] \approx \frac{2\alpha\tau^2(1-\beta)^2\xi_{\text{min}}^4}{(1+\alpha)(1-\alpha)^2(1+\beta)^2}. \tag{2.32}$$

It is observed that the first-moment and second-moment steady-state step size values associated with the TAASS approach are more complicated than those of the MASS method due to the presence of $v^2(i)$. Note that (2.31) and (2.32) can be employed in the analysis of the EMSE of the proposed TAASS algorithm. The details will be shown in the next part.

2.6.2 Steady-state Analysis

The classic approaches reported in [18], [55] for the steady-state and tracking analyses cannot be employed in our proposed algorithms since they have to decide the recursion for the weight error energy, which is difficult for the CCM criterion due to its inherent nonlinear updates [56], [57]. Instead, we adopt an energy flow framework [40], [58], [59] for the steady-state and tracking performance evaluation. The framework creates an energy conservation connection, relying on a fundamental error variance relation, between adjacent time instants, to avoid the difficult derivation of the recursion for the weight error energy. This approach holds not only for the CCM algorithms but for a general class of adaptive filters [60].

In the following derivation, we will give the MSE expression, which includes two

parts, i.e., the minimum mean square error (MMSE) and the excess mean square error (EMSE). By using the CCM weight expression in (2.6) to express the EMSE and employing the energy conservation approach, we could obtain the steady-state expressions for MASS and TAASS, respectively.

Following the classic MSE measure [18],

$$J_{\text{mse}} = \lim_{i\to\infty} \mathbb{E}\big[|e_c(i)|^2\big] = \lim_{i\to\infty} \mathbb{E}\big[|\tilde{\boldsymbol{w}}^H(i)\boldsymbol{x}(i) + n(i)|^2\big], \tag{2.33}$$

where $e_c(i) = d(i) - \boldsymbol{w}^H(i)\boldsymbol{x}(i)$, is the estimation error, $d(i)$ is the desired response with additive noise $n(i)$, the subscript "c" means that it is from the classic measure, and $\tilde{\boldsymbol{w}}(i) = \boldsymbol{w}_{\text{opt}} - \boldsymbol{w}(i)$ is the weight error vector.

Employing assumption iii), (2.33) becomes,

$$J_{\text{mse}} = \sigma_n^2 + \lim_{i\to\infty} \mathbb{E}\big[|\tilde{\boldsymbol{w}}^H(i)\boldsymbol{x}(i)|^2\big], \tag{2.34}$$

where σ_n^2 represents the variance of additive noise.

If we define a priori and a posteriori estimation errors, which represent

$$e_a(i) = \tilde{\boldsymbol{w}}^H(i)\boldsymbol{x}(i), \quad e_p(i) = \tilde{\boldsymbol{w}}^H(i+1)\boldsymbol{x}(i), \tag{2.35}$$

and then, J_{mse} in (2.34) can be expressed by

$$J_{\text{mse}} = \sigma_n^2 + \zeta_s, \tag{2.36}$$

where $\zeta_s = \lim_{i\to\infty} \mathbb{E}\big[|e_a(i)|^2\big]$ denotes the EMSE corresponding to the steady-state MSE measure. It is straightforward that calculating ζ_s is equivalent to finding the MSE.

Now, we consider the CCM weight expression and the energy conservation approach to express the EMSE for the steady-state analysis. In order to obtain ζ_s, we write (2.6) in a compact form

$$\boldsymbol{w}(i+1) = \boldsymbol{w}(i) + \mu(i)\boldsymbol{u}(i)F_e(i), \tag{2.37}$$

where, as mentioned before, $\boldsymbol{u}(i) = \big[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{x}(i)$, $e(i) = |y(i)|^2 - 1$, and $F_e(i) = -y^*(i)e(i)$.

Following the definition of $e_p(i)$ in (2.35), after some simple operations, we have

$$e_p(i) = e_a(i) - \mu(i)F_e^*(i)\boldsymbol{u}^H(i)\boldsymbol{x}(i) = e_a(i) - \mu(i)F_e^*(i)\|\boldsymbol{u}(i)\|^2, \tag{2.38}$$

where the second expression is obtained if we notice that $\boldsymbol{u}^H(i)\boldsymbol{x}(i) = \boldsymbol{x}^H(i)\big[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{x}(i) = \boldsymbol{u}^H(i)\boldsymbol{u}(i)$.

Substituting (2.38) into the weight error vector update equation defined in (2.14), yields

$$\tilde{\boldsymbol{w}}(i+1) = \tilde{\boldsymbol{w}}(i) - \frac{\boldsymbol{u}(i)}{\|\boldsymbol{u}(i)\|^2}\big[e_a^*(i) - e_p^*(i)\big]. \tag{2.39}$$

Rearranging (2.39) and squaring it, we obtain

$$\|\tilde{\boldsymbol{w}}(i+1)\|^2 + \bar{\mu}(i)|e_a(i)|^2 = \|\tilde{\boldsymbol{w}}(i)\|^2 + \bar{\mu}(i)|e_p(i)|^2, \tag{2.40}$$

where $\bar{\mu}(i) = 1/\|\boldsymbol{u}(i)\|^2$. This is an exact energy conservation relation that illustrates the energies of the weight error vectors between two successive time instants and which corresponds to the energies of the a priori and a posteriori estimation errors. Note that (2.40) is obtained without any assumptions. We use this expression to derive the EMSE for the steady-state analysis. In the steady-state environment, considering $\mathbb{E}\big[\|\tilde{\boldsymbol{w}}(i+1)\|^2\big] = \mathbb{E}\big[\|\tilde{\boldsymbol{w}}(i)\|^2\big]$, substituting (2.38) into (2.40), and taking expectations, then

$$\mathbb{E}[\bar{\mu}(i)|e_a(i)|^2] = \mathbb{E}[\bar{\mu}(i)|e_a(i) - \frac{\mu(i)}{\bar{\mu}(i)}F_e^*(i)|^2]. \tag{2.41}$$

Employing $e(i) = |y(i)|^2 - 1$ and $F_e(i) = -y^*(i)e(i)$ in (2.41),

$$\begin{aligned}\mathbb{E}\big\{\mu(i)e_a^*(i)y(i)\big[1 - |y(i)|^2\big]\big\} + \mathbb{E}\big\{\mu(i)e_a(i)y^*(i)\big[1 - |y(i)|^2\big]\big\} \\ = \mathbb{E}\big\{\frac{\mu^2(i)}{\bar{\mu}(i)}|y(i)|^2\big[1 - |y(i)|^2\big]^2\big\}.\end{aligned} \tag{2.42}$$

For ease of notation, we define J_{lp} and J_{rp} to represent terms in (2.42), i.e.,

$$\begin{aligned}J_{\text{lp}} &= \mathbb{E}\big\{\mu(i)e_a^*(i)y(i)\big[1 - |y(i)|^2\big]\big\} + \mathbb{E}\big\{\mu(i)e_a(i)y^*(i)\big[1 - |y(i)|^2\big]\big\}, \\ J_{\text{rp}} &= \mathbb{E}\big\{\frac{\mu^2(i)}{\bar{\mu}(i)}|y(i)|^2\big[1 - |y(i)|^2\big]^2\big\}.\end{aligned} \tag{2.43}$$

Considering the fact that the EMSE arises due to the presence of the MAI and additive noise, we will consider their impacts on the beamformer output in the following analysis.

Thus, the beamformer output can be expressed by

$$y(i) = \left[\boldsymbol{w}_{\text{opt}} - \tilde{\boldsymbol{w}}(i)\right]^H \boldsymbol{x}(i) = \boldsymbol{w}_{\text{opt}}^H \boldsymbol{x}(i) - e_a(i) = s_0(i) + M(i) + n(i) - e_a(i), \quad (2.44)$$

where $s_0(i)$ is the transmitted symbol of the desired user at time instant i, $M(i) = \sum_{k=1}^{q-1} \boldsymbol{w}_{\text{opt}}^H s_k(i)\boldsymbol{a}(\theta_k)$ is the output residual MAI caused as multiple users appear in the system, where $s_k(i)$ denotes the transmitted symbol of users with the exception of the desired one, and $n(i) = \boldsymbol{w}_{\text{opt}}^H \boldsymbol{n}(i)$ is the processed additive noise.

In order to develop further analysis, we make use of some properties, assumptions, and approximations.

Properties:$\{s, M, n, e_a\}$ are zero-mean random variables, and $\{s, M, n, e_a\}$ are mutually independent. The sources are independent and the processed additive noise n is a Gaussian random variable [61].

Approximation ii): The residual MAI is Gaussian [62].

The following analysis is on the basis of (2.42) and (2.43). We will study J_{lp} and J_{rp} in sequence, and then calculate the EMSE ζ_s for the proposed algorithms. We will drop time index i and use s to represent $s_0(i)$ for notation simplicity.

Substituting (2.44) into J_{lp} in (2.43) and using assumption i) and approximation ii), yields

$$J_{\text{lp}} = 2\mathbb{E}\left[\mu|e_a|^2(|s|^2 + |M|^2 + |n|^2 + |e_a|^2 - 1)\right]. \quad (2.45)$$

Define $\sigma_0^2 = \mathbb{E}\left[|s|^2\right]$, $\sigma_n^2 = \mathbb{E}\left[|n|^2\right]$, and $\sigma_M^2 = \mathbb{E}\left[|M|^2\right]$ as the desired signal power, the processed noise variance, and the residual MAI variance, respectively. In the steady-state environment, we have $\mathbb{E}\left[|n|^4\right] = 3\sigma_n^4$ and $\mathbb{E}\left[|n|^6\right] = 15\sigma_n^6$.

Note that the higher order of $\mathbb{E}\left[|e_a|^2\right]$ in (2.45) is negligible compared with the other terms as $i \to \infty$, it becomes

$$J_{\text{lp}} \approx 2K_1\mathbb{E}\left[\mu\right]\mathbb{E}\left[|e_a|^2\right], \quad (2.46)$$

where $K_1 = \sigma_0^2 + \sigma_M^2 + \sigma_n^2 - 1$.

At the same time, substituting (2.44) into J_{rp} in (2.43) and using assumption i) and

approximation ii), we have

$$J_{\text{rp}} = \mathbb{E}\big[\frac{\mu^2}{\bar{\mu}}(J_1 + J_2 + J_3 + J_4|e_a|^2)\big], \tag{2.47}$$

where
$J_1 = |s|^6 + 3|s|^4|M|^2 + 3|s|^4|n|^2 - 2|s|^4 + |s|^2 + 3|s|^2|M|^4 + 3|s|^2|n|^4 - 4|s|^2|M|^2 - 4|s|^2|n|^2 + 6|s|^2|M|^2|n|^2$;
$J_2 = |M|^6 + |n|^6 - 2|M|^4 - 2|n|^4 + |M|^2 + |n|^2 + 3|M|^4|n|^2 + 3|M|^2|n|^4 - 4|M|^2|n|^2$;
$J_3 = |e_a|^6 - 2|e_a|^4 + 3|M|^2|e_a|^4 + 3|s|^2|e_a|^4 + 3|n|^2|e_a|^4$;
$J_4 = 6|s|^2|n|^2 + 6|M|^2|n|^2 + 6|s|^2|M|^2 - 4|M|^2 - 4|n|^2 + 3|s|^4 - 4|s|^2 + 3|n|^4 + 3|M|^4 + 1$.

The higher order of $\mathbb{E}\big[|e_a|^2\big]$ in (2.47) can be ignored, yields,

$$J_{\text{rp}} \approx \frac{\mathbb{E}[\mu^2]}{\mathbb{E}[\bar{\mu}]}\big\{K_2 + K_3 + K_4\mathbb{E}\big[|e_a|^2\big]\big\}, \tag{2.48}$$

where
$K_2 = \sigma_0^6 + 3\sigma_0^4\sigma_M^2 + 3\sigma_0^4\sigma_n^2 - 2\sigma_0^4 + \sigma_0^2 + 9\sigma_0^2\sigma_M^4 + 9\sigma_0^2\sigma_n^4 - 4\sigma_0^2\sigma_M^2 - 4\sigma_0^2\sigma_n^2 + 6\sigma_0^2\sigma_M^2\sigma_n^2$;
$K_3 = 15\sigma_M^6 + 15\sigma_n^6 - 6\sigma_M^4 - 6\sigma_n^4 + \sigma_M^2 + \sigma_n^2 + 9\sigma_M^4\sigma_n^2 + 9\sigma_M^2\sigma_n^4 - 4\sigma_M^2\sigma_n^2$;
$K_4 = 6\sigma_0^2\sigma_n^2 + 6\sigma_M^2\sigma_n^2 + 6\sigma_0^2\sigma_M^2 - 4\sigma_M^2 - 4\sigma_n^2 - 4\sigma_0^2 + 3\sigma_0^4 + 9\sigma_n^4 + 9\sigma_M^4 + 1$.

From (2.42) and (2.43), equating J_{lp} and J_{rp} and rearranging it, we obtain

$$\zeta_s \approx \frac{\frac{\mathbb{E}[\mu^2]}{\mathbb{E}[\bar{\mu}]}(K_2 + K_3)}{2\mathbb{E}[\mu]K_1 - \frac{\mathbb{E}[\mu^2]}{\mathbb{E}[\bar{\mu}]}K_4} = \frac{\mathbb{E}[\mu^2]E\big[\|\boldsymbol{u}\|^2\big](K_2 + K_3)}{2\mathbb{E}[\mu]K_1 - \mathbb{E}[\mu^2]\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_4}, \tag{2.49}$$

where $\mathbb{E}\big[\|\boldsymbol{u}\|^2\big] = \sum_{k=1}^{q-1}\sigma_k^2\big\{\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_k) - \big[\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_k)/\big(\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0)\big)\big]\big\} + \sigma^2(m-1)$ [61], where σ_k^2 is the power of the kth user with the exception of the desired one, and additive noise at the output is a Gaussian random variable of type $n \sim \mathcal{N}(0, \sigma_n^2)$ with $\sigma_n = \|\boldsymbol{w}_{\text{opt}}\|\sigma$, $\boldsymbol{w}_{\text{opt}} = \boldsymbol{R}^{-1}\boldsymbol{p}$, and $\boldsymbol{p} = \mathbb{E}\big[\boldsymbol{x}d^*(i)\big]$. The term $\sigma_M = \sqrt{q-1}\boldsymbol{w}_{\text{opt}}^H\sigma_k\boldsymbol{a}(\theta_k)$ $(k \neq 0)$ with q is the number of users. Note that all the terms in (2.49) are relevant to the steady-state $i \to \infty$.

Equation (2.49) can be further simplified if we impose $\sigma_0^2 = 1$ and consider $\sigma_M^2 \ll \sigma_n^2$ [61], (2.49) becomes

$$\zeta_s \approx \frac{\mathbb{E}\big[\mu^2\big]\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_5}{2\sigma_n^2\mathbb{E}\big[\mu\big] - \mathbb{E}\big[\mu^2\big]\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_6}, \tag{2.50}$$

where $K_5 = 3\sigma_n^4 + 15\sigma_n^6$ and $K_6 = 2\sigma_n^2 + 9\sigma_n^4$.

Substituting (2.21) and (2.22) into (2.50), we obtain the EMSE for the MASS, which

is given by

$$\zeta_{s,\text{MASS}} \approx \frac{\alpha\tau\xi_{\min}\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_5}{\sigma_n^2(1-\alpha^2)-\alpha\tau\xi_{\min}\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_6}. \tag{2.51}$$

Substituting (2.31) and (2.32) into (2.50), we get the EMSE for the TAASS, which is given by

$$\zeta_{s,\text{TAASS}} \approx \frac{\alpha\tau(1-\beta)\xi_{\min}^2\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_5}{\sigma_n^2(1-\alpha^2)(1+\beta)-\alpha\tau(1-\beta)\xi_{\min}^2\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_6}. \tag{2.52}$$

The accuracy of the analysis is verified via simulations later.

2.6.3 Tracking Analysis

The energy conservation relation has been verified to provide the tracking analysis in a non-stationary environment [60]. The derivation in this section is based on the steady-state analysis and is specific for the proposed algorithms.

In the non-stationary environment, $\boldsymbol{w}_{\text{o}}(i)$ is not constant but assumed to vary following the model $\boldsymbol{w}_{\text{o}}(i+1)=\boldsymbol{w}_{\text{o}}(i)+\boldsymbol{q}(i)$, where $\boldsymbol{q}(i)$ denotes a random perturbation [18], [55]. This perturbation is introduced by the time variations of the system. The update model is invoked to track the variation. Thus, the update of the weight error vector can be expressed by

$$\tilde{\boldsymbol{w}}(i+1)=\tilde{\boldsymbol{w}}(i)+\boldsymbol{q}(i)-\mu(i)\boldsymbol{u}(i)F_e(i). \tag{2.53}$$

Squaring (2.53) yields

$$\|\tilde{\boldsymbol{w}}(i+1)\|^2+\bar{\mu}(i)|e_a(i)|^2=\|\tilde{\boldsymbol{w}}(i)+\boldsymbol{q}(i)\|^2+\bar{\mu}(i)|e_p(i)|^2. \tag{2.54}$$

The further analysis relies on the following assumption:

Assumption iv): The sequence $\{\boldsymbol{q}(i)\}$ is a stationary sequence of independent zero-mean vectors with positive definite covariance matrix $\boldsymbol{Q}=\mathbb{E}\big[\boldsymbol{q}(i)\boldsymbol{q}^H(i)\big]$. It is independent of the sequence $\big\{\boldsymbol{u}(i)\big\}$ and $\big\{n(i)\big\}$.

Under assumption iv), using $\mathbb{E}\big[\tilde{\boldsymbol{w}}^H(i)\boldsymbol{q}(i)\big]=0$ [63] and $\mathbb{E}\big[\|\tilde{\boldsymbol{w}}(i+1)\|^2\big]=$

$\mathbb{E}\big[\|\tilde{\boldsymbol{w}}(i)\|^2\big]$ and taking expectations, the energy equation (2.54) can be written as

$$\mathbb{E}\big[\bar{\mu}(i)|e_a(i)|^2\big] = \text{trace}(\boldsymbol{Q}) + \mathbb{E}\big[\bar{\mu}(i)|e_a(i) - \frac{\mu(i)}{\bar{\mu}(i)}F_e^*(i)|^2\big]. \tag{2.55}$$

In view of (2.55), it can be regarded as an extension of (2.43) with an addition of the system nonstationary contribution trace($\boldsymbol{Q}$). This is an advantage of the energy conservation approach over classic approaches [18], [55] since it allows us to develop a tracking analysis by analyzing the results in the stationary case.

Substituting $e(i) = |y(i)|^2 - 1$ and $F_e(i) = -y^*(i)e(i)$ into (2.55), we have

$$\begin{aligned} J_{t,\text{lp}} &\approx 2K_1\mathbb{E}\big[\mu\big]\mathbb{E}\big[|e_a|^2\big] \\ J_{t,\text{rp}} &\approx \text{trace}(\boldsymbol{Q}) + \frac{\mathbb{E}\big[\mu^2\big]}{\mathbb{E}\big[\bar{\mu}\big]}\big\{K_2 + K_3 + K_4\mathbb{E}\big[|e_a|^2\big]\big\}, \end{aligned} \tag{2.56}$$

and

$$\zeta_t \approx \frac{\text{trace}(\boldsymbol{Q}) + \mathbb{E}\big[\mu^2\big]\mathbb{E}\big[\|\boldsymbol{u}\|^2\big](K_2 + K_3)}{2\mathbb{E}\big[\mu\big]K_1 - \mathbb{E}\big[\mu^2\big]\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_4}, \tag{2.57}$$

where ζ_t denotes the EMSE corresponding to the tracking condition, the terms K_1, K_2, K_3 and K_4 are the same as those in (2.46) and (2.48), and it is assumed that $\boldsymbol{q}(i)$ is independent of $\{s, M, n, e_a\}$, and higher orders of $\big\{|e_a|^2\big\}$ are ignored.

If $\sigma_0^2 = 1$ and $\sigma_M^2 \ll \sigma_n^2$ are imposed on (2.56), we have a simpler expression

$$\zeta_t \approx \frac{\text{trace}(\boldsymbol{Q}) + \mathbb{E}\big[\mu^2\big]\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_5}{2\sigma_n^2\mathbb{E}\big[\mu\big] - \mathbb{E}\big[\mu^2\big]\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_6}, \tag{2.58}$$

where K_5 and K_6 are the same as those in (2.50).

Substituting (2.21), (2.22), and (2.31), (2.32) into (2.58), respectively, the EMSE for the tracking analysis with respect to the MASS and TAASS can be expressed by

$$\zeta_{t,\text{MASS}} \approx \frac{(1-\alpha)^2(1+\alpha)\text{trace}(\boldsymbol{Q}) + 2\alpha\tau^2\xi_{\text{min}}^2\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_5}{2\sigma_n^2(1-\alpha^2)\tau\xi_{\text{min}} - 2\alpha\tau^2\xi_{\text{min}}\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_6}, \tag{2.59}$$

Tab. 2.1: Simulation parameters for Case A and Case C

Adaptive algorithms	Case A				Case C	
	Expt. 1		Expt. 2		Expt. 7	
	MASS	TAASS	MASS	TAASS	MASS	TAASS
α	0.98	0.98	0.99	0.99	0.98	0.98
β	-	0.99	-	0.975	-	0.99
τ	10^{-3}	10^{-3}	10^{-4}	10^{-3}	10^{-3}	10^{-3}
μ_0	10^{-5}	10^{-4}	10^{-5}	10^{-4}	10^{-5}	10^{-5}
$\mu_{\max}$	0.003	0.006	0.003	0.006	0.003	0.006
$\mu_{\min}$	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}

Tab. 2.2: Simulation parameters for Case B

Adaptive algorithms	Case B							
	Expt. 3		Expt. 4		Expt. 5		Expt. 6	
	MASS	TAASS	MASS	TAASS	MASS	TAASS	MASS	TAASS
α	0.987	0.988	0.988	0.988	0.986	0.988	0.99	0.989
β	-	0.975	-	0.975	-	0.975	-	0.98
τ	10^{-4}	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-3}
μ_0	10^{-5}	10^{-4}	10^{-5}	10^{-4}	10^{-5}	10^{-4}	10^{-5}	10^{-4}
$\mu_{\max}$	0.003	0.006	0.003	0.006	0.004	0.006	0.004	0.007
$\mu_{\min}$	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}

and

$$\zeta_{t,\text{TAASS}} \approx \frac{(1-\alpha)^2(1+\alpha)(1+\beta)^2\text{trace}(\boldsymbol{Q}) + 2\alpha\tau^2(1-\beta)^2\xi_{\min}^4\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_5}{2\sigma_n^2(1-\alpha^2)(1-\beta^2)\tau\xi_{\min}^2 - 2\alpha\tau^2(1-\beta)^2\xi_{\min}^4\mathbb{E}\big[\|\boldsymbol{u}\|^2\big]K_6}. \quad (2.60)$$

2.7 Simulation Results

In this section, we illustrate the effectiveness and the advantages of the proposed MASS and TAASS algorithms over existing methods through simulations and verify the accuracy of the analyses. The CCM criterion is considered with the SG and RLS algorithms. Simulations are carried out in stationary and non-stationary scenarios. All simulations are performed by a ULA containing $m = 16$ sensor elements with half-wavelength spacing. The noise is zero mean additive white Gaussian noise. A total of $K = 1000$ runs are used to get each curve. In all experiments, the desired signal power is $\sigma_0^2 = 1$. The BPSK modulation scheme is employed to modulate the signals. The simulation parameters used in each experiment are listed in Table 2.1 and Table 2.2.

Fig. 2.4 (Expt. 1) compares the proposed MASS and TAASS algorithms with the FSS, ASS, and RLS methods by showing the bit error ratio (BER) versus the input signal-to-noise ratio (SNR). The input SNR is varied between 0 and 30 dB. The number of interferers is $q_{\text{inf}} = 5$ with one 4 dB above the power level of the desired user, one with the same power level of the desired user, and three -0.5 dB below the power level of

the desired user. The number of snapshots is fixed $N = 1000$. Note that the exact DOA of the desired user is known by the receiver beforehand. We set the first element of the initial weight vector $\boldsymbol{w}(0)$ equal to the corresponding element of $\boldsymbol{a}(\theta_0)$. Other parameters for the proposed and existing methods are tuned to optimize the performance, allowing for a fair comparison. It is clear that the BER values of all methods decrease following the increase of the input SNR. The proposed MASS and TAASS algorithms outperform the FSS and ASS ones. The curve of the TAASS algorithm is close to that of the RLS method, which shows its robustness in the studied scenario.

Fig. 2.5 (Expt. 2) depicts the performance of the studied algorithms under the mismatch (steering vector error) condition, which includes two experiments. Fig. 2.5(a) keeps the scenario in Fig. 2.4 except with a fixed input SNR$=$ 20 dB. The results indicate that the proposed algorithms outperform the existing FSS and ASS methods. The performance of the TAASS algorithm is close to the RLS method but with much lower complexity. The mismatch scenario is shown in Fig. 2.5(b). The DOA of the desired user estimated by the receiver is 1^o away from the exact direction. It is evident that the mismatch problem degrades the performance of all the methods. The proposed algorithms retain their predominance over the FSS and ASS methods. The curves of the TAASS and RLS algorithms are shown to interact when the snapshots increase, which exhibits the advantage of the proposed algorithm.

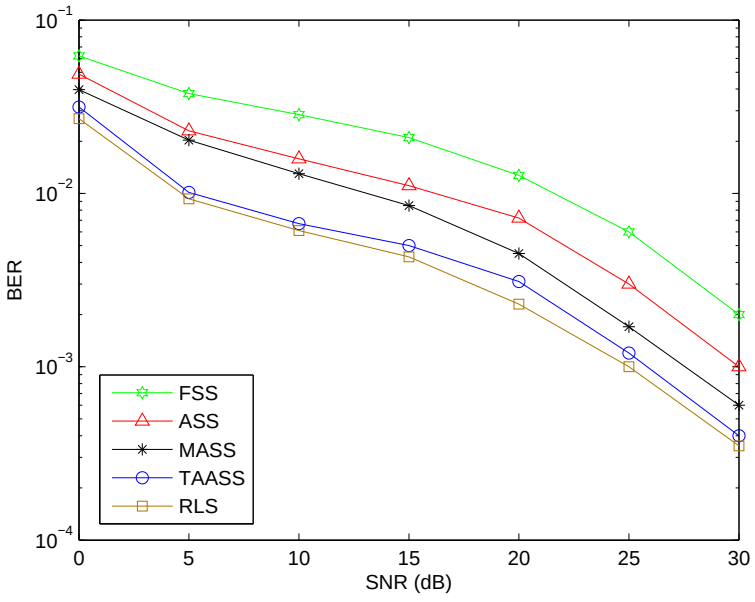

Fig. 2.4: Simulation result for Case A, Expt. 1: Curves for BER versus input SNR for various adaptive algorithms.

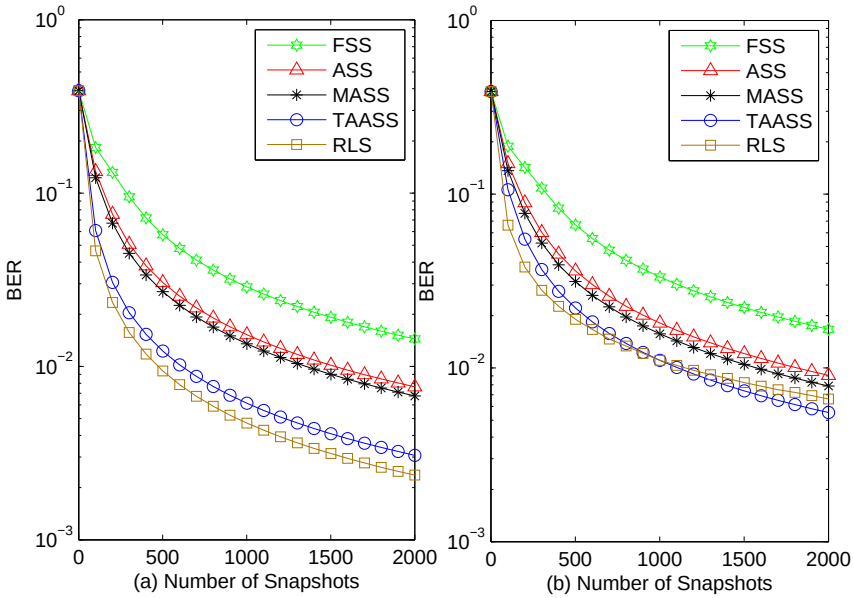

Fig. 2.5: Simulation result for Case A, Expt. 2: Curves for BER versus the number of snapshots for (a) ideal steering vector condition and (b) steering vector with mismatch$= 1^o$.

We compare the theoretical analysis with the simulation results in Fig. 2.6 (Expt. 3) and Fig. 2.7 (Expt. 4) for both the MASS and TAASS algorithms. In Fig. 2.6, the input SNR is fixed at 20 dB and the number of interferers is $q_{\text{inf}} = 3$ with all -0.5dB below the desired power. Fig. 2.6 (a) and Fig. 2.6 (b) correspond to the MASS and TAASS methods, respectively. The proposed algorithms converge to the steady-state that is in good match with the curves obtained from (2.51) and (2.52), where $\xi_{\text{min}} = 1 - \boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)$. The oscillation is due to the system noise and the users' status (e.g., power level and the number).

In Fig. 2.7, we increase $q_{\text{inf}} = 4$ with one 2 dB above the desired power, one with the same power as the desired user, and two -0.5 dB below the desired power. From this simulation, the proposed algorithms need more snapshots to come to the steady-state, which is higher than that in Fig. 2.6, but still close to their theoretical values. The TAASS method shows the superior performance over the MASS for both the theoretical value and simulation result under this severe condition. Comparing Fig. 2.6 with Fig. 2.7, we find that the number or/and the power level of the interferers increase, deteriorate the performance of both simulation and theory. Note that the ignored terms in (2.51) and (2.52) do not affect the accuracy significantly.

Fig. 2.8 (Expt. 5) compares the theoretical analysis and simulation results of the proposed algorithms by showing the EMSE versus the input SNR. It works with the same

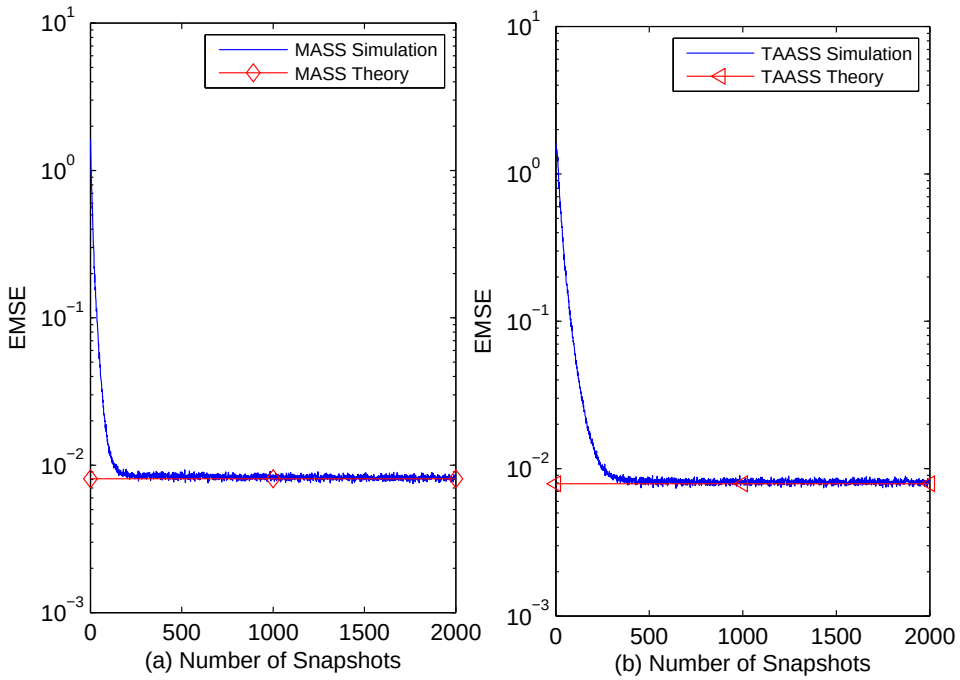

Fig. 2.6: Simulation results for Case B, Expt.3: Curves for EMSE versus the number of snapshots for (a) MASS and (b) TAASS.

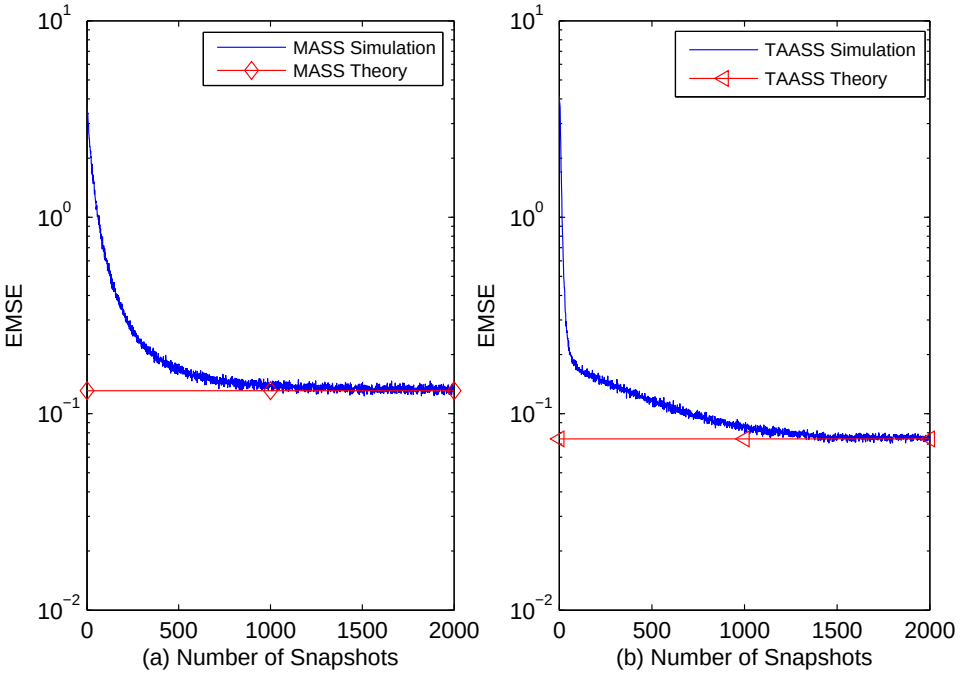

Fig. 2.7: Simulation results for Case B, Expt.4: Curves for EMSE versus the number of snapshots for (a) MASS and (b) TAASS.

scenario as in Fig. 2.6. In Fig. 2.8, we find that the EMSE decreases following the increase of the input SNR. Simulation results converge to the theoretical ones, especially with high input SNR. Fig. 2.9 (Expt. 6) shows the result in a more severe condition, where $q_{\text{inf}} = 4$ with one 3 dB above the desired power, one with the same power of the desired user, and two -0.5 dB below the power of the desired user. Compared with Fig. 2.8, the EMSE performance is deteriorated. Under this condition, the advantage of the TAASS algorithm is obvious, especially with high SNR values.

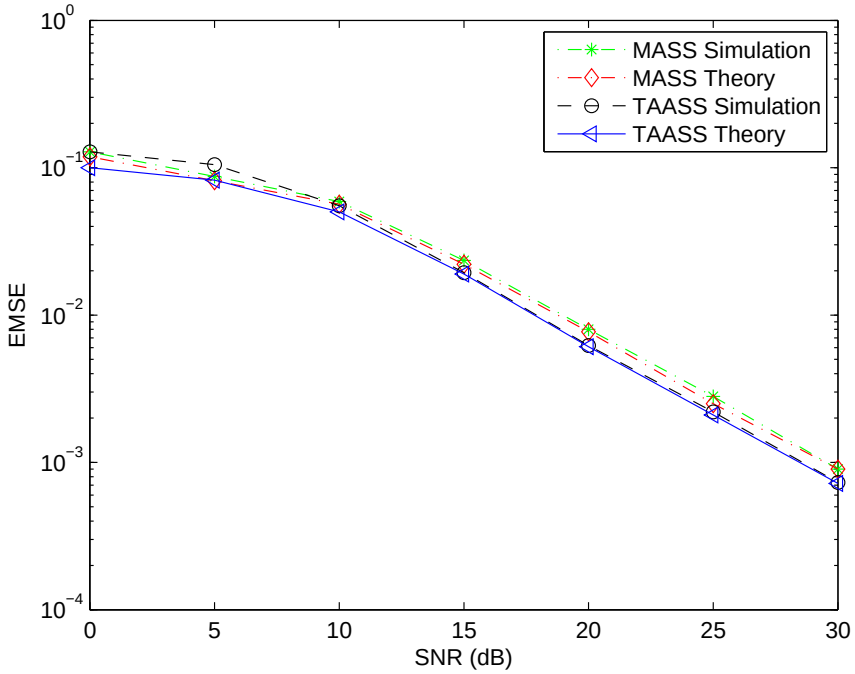

Fig. 2.8: Simulation result for Case B, Expt. 5: Curves for EMSE versus input SNR for proposed adaptive algorithms.

In Fig. 2.10 (Expt. 7), we evaluate the performance of the proposed MASS and TAASS algorithms in a non-stationary scenario, namely, when the number of users changes. The scenario starts with $q_{\text{inf}} = 4$ interferers, two with same power of the desired user and two with -0.5 dB below the power of the desired user. The input SNR is set to 20 dB. From the first stage (first 1500 snapshots) of Fig. 2.10, the convergence rate of the proposed MASS and TAASS is faster than those of the FSS and ASS methods. The convergence and output performance of the TAASS algorithms is close to the RLS method. Two more users with one 2 dB above the power of the desired user and one -0.5 dB below the desired power, enter the system at 1500 snapshots. This change makes the output SINR reduce suddenly and degrades the performance of all methods. The proposed algorithms keep faster convergence and better performance in comparison with the FSS and ASS

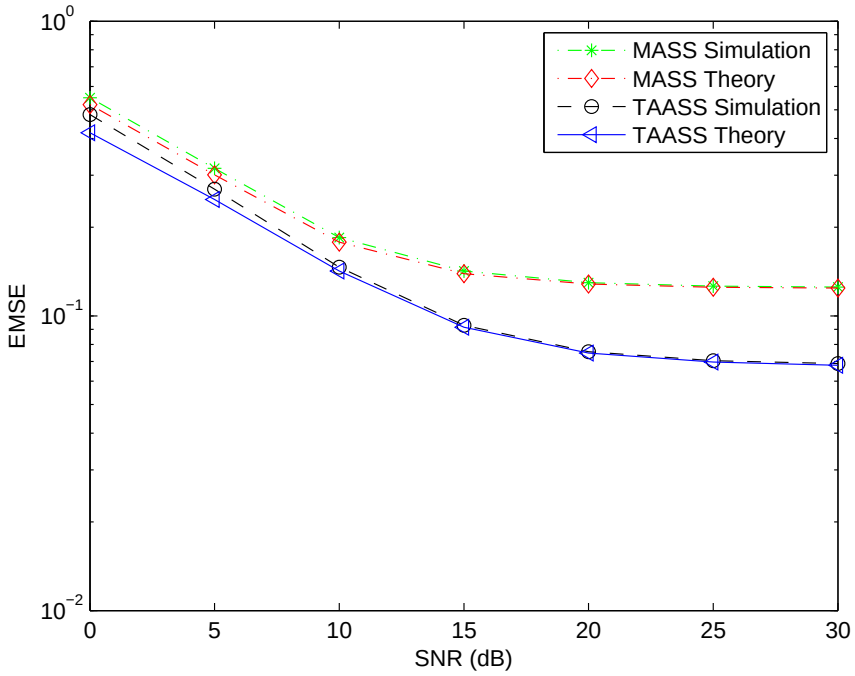

Fig. 2.9: Simulation result for Case B, Expt. 6: Curves for EMSE versus input SNR for proposed adaptive algorithms.

methods. The RLS method has a superior performance than the proposed algorithms but has to pay much higher computational cost. Note that the output SINR values of all the methods at 1500 snapshots are set to around -2 dB since it is convenient to show the convergence behaviors.

Fig. 2.11 (Expt. 7) depicts the step size adaptation of the proposed algorithms following the change of the scenario in Fig. 2.10. The large step size values appeared at the beginning when the algorithms are far from the optimal solutions to increase the convergence rate, and the small values are used when the algorithms are near the optimum to achieve a low level of misadjustment. The change of the scenario makes the step size increase again and then reduce to track this non-stationary scenario.

2.8 Conclusions

In this chapter, two adaptive SG algorithms with the CCM criterion are developed. They employ different adaptive mechanisms, which are based on the energy of prediction error and the time average of the correlation of the estimation error, respectively, to adjust

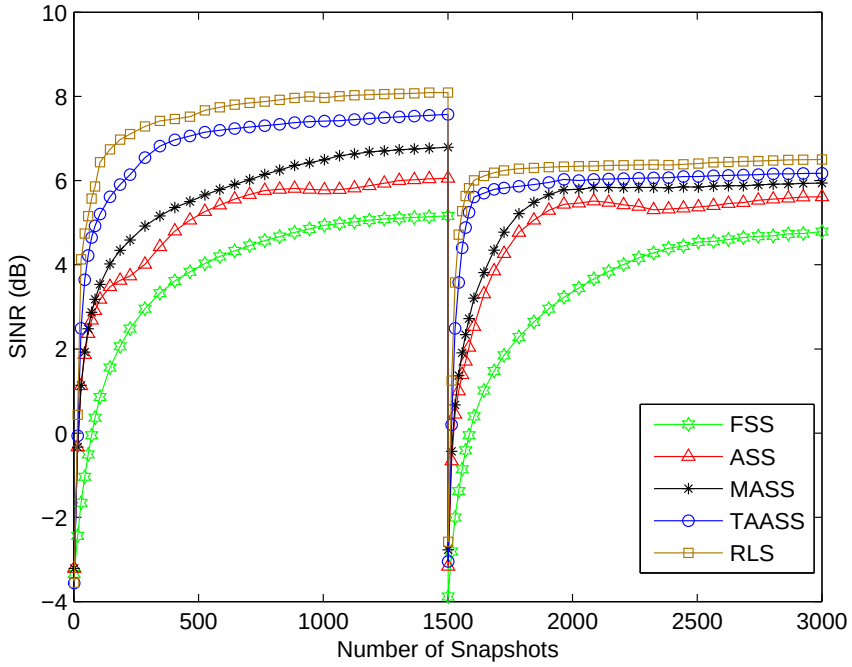

Fig. 2.10: Simulation results for Case C, Expt. 7: Curves for output SINR versus the number of snapshots for various adaptive algorithms.

the step size. The theoretical expressions of the EMSE, in both steady-state and tracking cases, were derived by using the energy conservation approach. The effects of additive noise and the MAI in the array output were considered in the analysis. Simulation results were carried out to compare the proposed algorithms with existing methods and verify the analysis. The proposed algorithms possess good performance and fast convergence rate in the stationary scenarios. They can be applied to the beamformer design and other communication techniques. Note that we didn't perform the simulation results for the tracking analysis in the non-stationary scenarios. The tracking analysis was given here for keeping the integrity of the analysis.

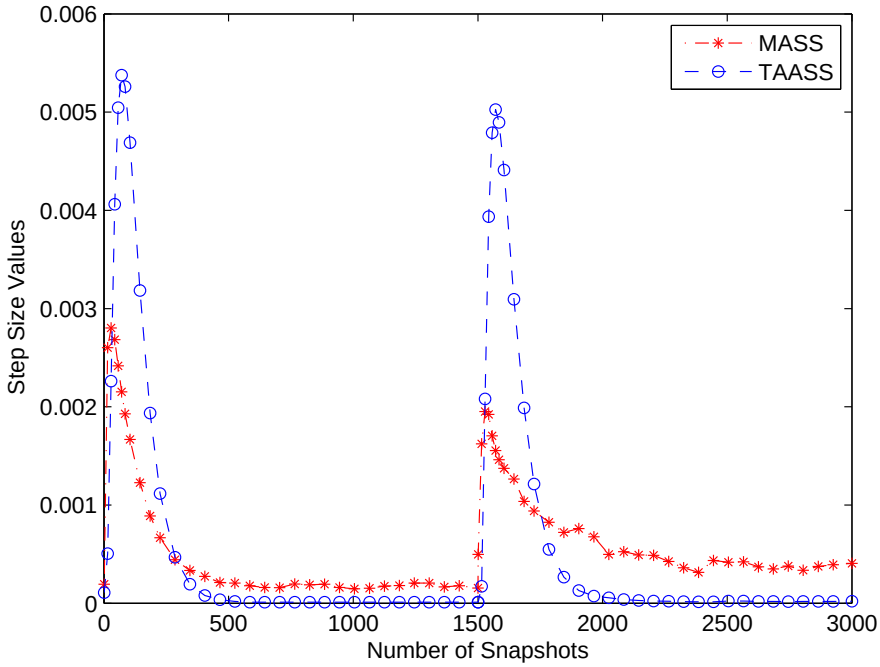

Fig. 2.11: Simulation result for Case C, Expt. 7: Curves for step size values versus the number of snapshots for proposed adaptive algorithms.

3. CONSTRAINED ADAPTIVE FILTERING ALGORITHMS BASED ON THE CONJUGATE GRADIENT METHOD FOR BEAMFORMING

Contents

3.1 Introduction

In this chapter, we employ another array processing algorithm, i.e., the conjugate gradient (CG), for the application of beamforming. The CG algorithm was reported to solve large systems of linear equations. For array processing, as explained in Chapter 1, the CG has an attractive tradeoff between performance and complexity as compared with the SG and RLS methods since it enjoys a convergence comparable to the RLS with a computational requirement which is intermediate between the SG and RLS methods [64]. Another class of algorithms related to the CG are those based on the MSWF [27] and the AVF [29, 30]. Since they belong to the reduced-rank class of techniques, we will give more details about these algorithms in Chapter 5 and Chapter 6. It is worth noting that the basis vectors of the CG, MSWF, and AVF span the same Krylov subspace [65]. Their main differences lie on the computational cost, structure of adaptation, and ease of implementation. Many adaptive algorithms based on the CG technique have been reported [66]- [68]. However, the incorporation of constraints in existing CG algorithms leads to a significant increase in the computational cost. The linear constraint here corresponds to the knowledge of the DOA of the SOI.

We develop two CG adaptive algorithms for the beamformer design according to the

CMV and CCM criteria. The existing methods yield a system of equations which requires the costly matrix inversion for ensuring the constraint and solving the system of equations. The proposed algorithms are motivated to circumvent this problem. A CG-based weight vector strategy is devised to incorporate the constraint of the array response in the proposed algorithms for the beamformer design. We create a simple relation between the CG-based weight vector and the matrix inversion and the array response of the SOI. The weight solution can be obtained by iteratively computing the CG-based weight vector. The proposed algorithms enforce the constraint in the system of equations without the matrix inversion and exhibit fast convergence and comparable performance with low-complexity. Furthermore, the numerical instability [18] found in the RLS algorithm is addressed by the proposed methods. The convexity property of the CCM criterion and its convergence analysis are explained. Simulations are performed to demonstrate the performance of the proposed algorithms against the best known techniques.

The remaining of this chapter is organized as follows: the problem statement about the incorporation of the constraints in the system of equations is introduced in section 3.2. Section 3.3 introduces the proposed CG algorithms with the CMV and CCM criteria. Section 3.4 is dedicated to the analyses of the new methods. Simulation results are discussed in Section 3.5, and conclusions are drawn in Section 3.6.

3.2 Problem Statement

The derivation of the array processing algorithms for the beamformer design is according to the CMV and CCM criteria, which have been given at the end of Chapter 2. In order to solve the optimization problem in (2.3) and (2.4), we resort to the method of Lagrange multipliers. The weight expressions obtained are [32], [94]

$$\boldsymbol{w}_{\text{cmv}} = [\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)]^{-1}\gamma\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0), \tag{3.1}$$

$$\boldsymbol{w}_{\text{ccm}} = [\boldsymbol{a}^H(\theta_0)\boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0)]^{-1}\gamma\boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0), \tag{3.2}$$

where $e_y = \big||y(i)|^2 - 1\big|$, $\boldsymbol{R} = \mathbb{E}[\boldsymbol{x}(i)\boldsymbol{x}^H(i)]$, and $\boldsymbol{R}_y = \mathbb{E}[e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)] \in \mathbb{C}^{m\times m}$ is a matrix with cross correlations between the output $y(i)$ and the received vector $\boldsymbol{x}(i)$. It is worth noting that $\boldsymbol{R}_y$ depends on $y(i)$, which is a function of the weight vector $\boldsymbol{w}(i)$. A solution of (3.2) can be obtained by setting an initial value of $\boldsymbol{w}(i)$ and running an iterative procedure, which will be shown in the following part.

Considering the expressions in (3.1) and (3.2), we can manipulate and organize them

into the following systems of equations

$$\left(\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)\right)\boldsymbol{R}\boldsymbol{w}_{\text{cmv}} = \gamma\boldsymbol{a}(\theta_0), \tag{3.3}$$

$$\left(\boldsymbol{a}^H(\theta_0)\boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0)\right)\boldsymbol{R}_y\boldsymbol{w}_{\text{ccm}} = \gamma\boldsymbol{a}(\theta_0), \tag{3.4}$$

where the terms $\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)$ and $\boldsymbol{a}^H(\theta_0)\boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0)$ are responsible for ensuring the constraints. The inversions of $\boldsymbol{R}$ and $\boldsymbol{R}_y$ increase the computational load and suffer from numerical instability. Numerous iterative algorithms can be used to solve general systems of equations [18, 19], [69]- [71]. Among them, CG is an efficient method with very attractive tradeoff between performance and complexity. Here, we plan to use the CG-based algorithm to enforce the constraint and allow the development of the beamformer design due to its attractive tradeoff between performance and complexity, and suitability for implementation.

3.3 Proposed Adaptive CG algorithms

We introduce two CG algorithms for the beamformer design according to the CMV and CCM criteria, respectively. The proposed methods avoid the matrix inversion and possess fast convergence and good performance with low-complexity. In this section, we first briefly review the existing CG algorithm [19]. Then we formulate a simple relation between a CG-based weight vector, the matrix inversion and the steering vector. On the basis of this relation, we derive the conventional conjugate gradient (CCG) and the modified conjugate gradient (MCG) algorithms in what follows.

3.3.1 Conjugate Gradient Algorithms

The CG algorithm can be employed for solving optimization problems of the form [72, 73]

$$\min_{\boldsymbol{v}} J(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^H\boldsymbol{R}\boldsymbol{v} - \Re\{\boldsymbol{b}^H\boldsymbol{v}\}, \tag{3.5}$$

where $\boldsymbol{R} \in \mathbb{C}^{m\times m}$ is the covariance matrix of the received vector $\boldsymbol{x}(i)$, $\boldsymbol{b} \in \mathbb{C}^{m\times 1}$ is the cross-correlation vector between the received vector $\boldsymbol{x}(i)$ and the desired response $d(i)$, and $\boldsymbol{v} \in \mathbb{C}^{m\times 1}$ is the CG weight vector. The CG algorithm solves (3.5) by iteratively updating the CG weight vector as

$$\boldsymbol{v}_k = \boldsymbol{v}_{k-1} + \alpha_k\boldsymbol{p}_k, \tag{3.6}$$

where $\boldsymbol{p}_k$ is the direction vector with conjugacy, i.e., $\boldsymbol{p}_k^H \boldsymbol{R} \boldsymbol{p}_l = 0$ for $k \neq l$, α_k is calculated by substituting (3.6) into (3.5) and then minimizing with respect to α_k, and $k = 1, \ldots, K$ is the iteration number. Note that $K \leq m$ since the covariance matrix is composed of at most m independent vectors [74].

The direction vector $\boldsymbol{p}_k$ in (3.6) is obtained by a linear combination of the previous direction vector and the negative gradient vector $\boldsymbol{g}_k = \boldsymbol{b} - \boldsymbol{R}\boldsymbol{v}_k$ [72]. Thus, it can be expressed as

$$\boldsymbol{p}_{k+1} = \boldsymbol{g}_k + \beta_k \boldsymbol{p}_k, \tag{3.7}$$

where β_k is chosen to provide conjugacy for the direction vectors. The CG algorithm and some related properties can be found in [19, 74, 75].

3.3.2 Proposed Conventional Conjugate Gradient (CCG) Algorithms

We describe the proposed algorithm for the CMV criterion in detail and extend it to the CCM criterion. Note that the CG algorithm with the minimum variance criterion has been reported [20]. We give the details of this algorithm since it uses the CG-based weight vector to create a relation between the matrix inversion and the steering vector of the SOI. This relation is important to the development of the proposed algorithms.

The CMV-CCG Algorithm

Let us describe an optimization problem for the CCG algorithm, which is

$$J(\boldsymbol{v}_{\text{cmv}}) = \frac{1}{2} \boldsymbol{v}_{\text{cmv}}^H \boldsymbol{R} \boldsymbol{v}_{\text{cmv}} - \Re\{\boldsymbol{a}^H(\theta_0) \boldsymbol{v}_{\text{cmv}}\}, \tag{3.8}$$

where $\boldsymbol{v}_{\text{cmv}} \in \mathbb{C}^{m \times 1}$ is a CG-based weight vector for the algorithm, the subscript "cmv" means that it is for the CMV criterion, $\boldsymbol{R}$ is the input covariance matrix, and $\boldsymbol{a}(\theta_0)$ is the steering vector of the SOI. By taking the gradient of (3.8) with respect to $\boldsymbol{v}_{\text{cmv}}$, equating it to a null vector and rearranging the expression, we obtain

$$\boldsymbol{v}_{\text{cmv}} = \boldsymbol{R}^{-1} \boldsymbol{a}(\theta_0). \tag{3.9}$$

It is clear that, by introducing the CG-based weight vector $\boldsymbol{v}_{\text{cmv}}$, we formulate a simple relation with the matrix inversion and the steering vector of the SOI. In the algorithm, $\boldsymbol{v}_{\text{cmv}}$ is iterated and then substituted into (3.1) for the weight solution. Note that (3.5) and (3.8)

are in the similar form but with different meanings since $\boldsymbol{v}_{\text{cmv}}$ is not directly employed for the weight solution but regarded as an intermediate weight vector for enforcing the constraint, solving the systems of equations, and avoiding the numerical problems caused by the matrix inversion.

In the algorithm, the iteration procedure is executed per snapshot. For the ith snapshot, $\boldsymbol{R}$ is replaced by $\boldsymbol{R}(i)$, which is fixed throughout the K iterations of the CCG operation. In practice, $\boldsymbol{R}(i)$ is estimated by its recursive version $\hat{\boldsymbol{R}}(i) = \lambda\hat{\boldsymbol{R}}(i-1) + \boldsymbol{x}(i)\boldsymbol{x}^H(i)$, where λ is the forgetting factor that is a positive constant close to, but less than 1. Taking the gradient of (3.8) with respect to $\boldsymbol{v}_{\text{cmv},k}(i)$ for the kth iteration and choosing the negative direction, we get the negative gradient

$$\boldsymbol{g}_{\text{cmv},k}(i) = \boldsymbol{a}(\theta_0) - \hat{\boldsymbol{R}}(i)\boldsymbol{v}_{\text{cmv},k}(i). \tag{3.10}$$

The CG-based weight vector $\boldsymbol{v}_{\text{cmv},k}(i)$ is defined as

$$\boldsymbol{v}_{\text{cmv},k}(i) = \boldsymbol{v}_{\text{cmv},k-1}(i) + \alpha_{\text{cmv},k}(i)\boldsymbol{p}_{\text{cmv},k}(i), \tag{3.11}$$

where $\boldsymbol{p}_{\text{cmv},k}(i)$ is the direction vector with conjugacy, i.e., $\boldsymbol{p}^H_{\text{cmv},k}(i)\hat{\boldsymbol{R}}(i)\boldsymbol{p}_{\text{cmv},l}(i) = 0$ for $k \neq l$ and $\alpha_{\text{cmv},k}(i)$ is calculated by substituting (3.11) into (3.8), taking the gradient with respect to $\alpha_{\text{cmv},k}(i)$, and using (3.11), which yields

$$\alpha_{\text{cmv},k}(i) = \left(\boldsymbol{p}^H_{\text{cmv},k}(i)\hat{\boldsymbol{R}}(i)\boldsymbol{p}_{\text{cmv},k}(i)\right)^{-1}\boldsymbol{g}^H_{\text{cmv},k-1}(i)\boldsymbol{p}_{\text{cmv},k}(i). \tag{3.12}$$

The direction vector in (3.11) is defined as

$$\boldsymbol{p}_{\text{cmv},k+1}(i) = \boldsymbol{g}_{\text{cmv},k}(i) + \beta_{\text{cmv},k}(i)\boldsymbol{p}_{\text{cmv},k}(i), \tag{3.13}$$

where $\beta_{\text{cmv},k}(i)$ is calculated by the Gram Schmidt process [72, 75] for the conjugacy

$$\beta_{\text{cmv},k}(i) = \left[\boldsymbol{g}^H_{\text{cmv},k-1}(i)\boldsymbol{g}_{\text{cmv},k-1}(i)\right]^{-1}\left[\boldsymbol{g}^H_{\text{cmv},k}(i)\boldsymbol{g}_{\text{cmv},k}(i)\right]. \tag{3.14}$$

After K iterations, the weight solution of the CCG algorithm with respect to the CMV criterion at time instant i is obtained by substituting $\boldsymbol{v}_{\text{cmv},K}(i)$ into (3.1), which is given by

$$\boldsymbol{w}_{\text{cmv}}(i) = \left[\boldsymbol{a}^H(\theta_0)\boldsymbol{v}_{\text{cmv},K}(i)\right]^{-1}\gamma\boldsymbol{v}_{\text{cmv},K}(i). \tag{3.15}$$

The CMV-CCG algorithm is summarized in Table 3.1, where δ is the regularization

Tab. 3.1: The CMV-CCG algorithm

Initialization:

$\boldsymbol{v}_0(1) = \mathbf{0};\ \hat{\boldsymbol{R}}(0) = \delta \boldsymbol{I}$

Update for each time instant $i = 1, \ldots, N$

STEP 1: Start:

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

$\boldsymbol{g}_0(i) = \boldsymbol{a}(\theta_0) - \hat{\boldsymbol{R}}(i)\boldsymbol{v}_0(i)$

$\boldsymbol{p}_1(i) = \boldsymbol{g}_0(i)$

STEP 2: For $k = 1, \ldots, K$**:**

$\alpha_k(i) = \left(\boldsymbol{p}_k^H(i)\hat{\boldsymbol{R}}(i)\boldsymbol{p}_k(i)\right)^{-1}\boldsymbol{g}_{k-1}^H(i)\boldsymbol{p}_k(i)$

$\boldsymbol{v}_k(i) = \boldsymbol{v}_{k-1}(i) + \alpha_k(i)\boldsymbol{p}_k(i)$

$\boldsymbol{g}_k(i) = \boldsymbol{g}_{k-1}(i) - \alpha_k(i)\hat{\boldsymbol{R}}(i)\boldsymbol{p}_k(i)$

$\beta_k(i) = \left(\boldsymbol{g}_{k-1}^H(i)\boldsymbol{g}_{k-1}(i)\right)^{-1}\left(\boldsymbol{g}_k^H(i)\boldsymbol{g}_k(i)\right)$

$\boldsymbol{p}_{k+1}(i) = \boldsymbol{g}_k(i) + \beta_k(i)\boldsymbol{p}_k(i)$

$k = k + 1$

STEP 3: After K **iterations:**

$\boldsymbol{v}_0(i+1) = \boldsymbol{v}_K(i)$

$\boldsymbol{w}_{\text{cmv}}(i) = \left(\boldsymbol{a}^H(\theta_0)\boldsymbol{v}_K(i)\right)^{-1}\gamma \boldsymbol{v}_K(i)$

$i = i + 1$

parameter to initialize the covariance matrix and the subscript "cmv" is removed for simplicity. The recursive formulation of $\boldsymbol{g}_k(i)$ is obtained by substituting (3.11) into (3.10) and recognizing that $\boldsymbol{g}_{k-1}(i) = \boldsymbol{a}(\theta_0) - \hat{\boldsymbol{R}}(i)\boldsymbol{v}_{k-1}(i)$. Note that the direction vector needs to be reset periodically, i.e., $\boldsymbol{p}_1(i) = \boldsymbol{g}_0(i)$, for ensuring the convergence [74, 75].

The CCM-CCG Algorithm

In order to derive the proposed algorithms with the CCM criterion, we consider the following optimization problem

$$J(\boldsymbol{v}_{\text{ccm}}) = \frac{1}{2}\boldsymbol{v}_{\text{ccm}}^H \boldsymbol{R}_y \boldsymbol{v}_{\text{ccm}} - \Re\{\boldsymbol{a}^H(\theta_0)\boldsymbol{v}_{\text{ccm}}\}, \tag{3.16}$$

where the subscript "ccm" means that it is for the CCM criterion and $\boldsymbol{R}_y$ is the matrix with cross correlations between $y(i)$ and $\boldsymbol{x}(i)$. For the ith snapshot, $\boldsymbol{R}_y(i)$ is estimated by $\hat{\boldsymbol{R}}_y(i) = \lambda_y \hat{\boldsymbol{R}}_y(i-1) + e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)$.

Performing a similar development as that for the CMV criterion, we get the proposed CCM-CCG algorithm, which is summarized in Table 3.2, where δ_y is to initialize the matrix $\boldsymbol{R}_y$. It is necessary to initialize the weight vector $\boldsymbol{w}_{\text{ccm}}(0)$ and run the iterative procedure since $e_y(i)$ depends on $\boldsymbol{w}_{\text{ccm}}(i)$.

Tab. 3.2: The CCM-CCG algorithm

Initialization:

$\boldsymbol{v}_0(1) = \mathbf{0};\ \boldsymbol{w}_{ccm}(0) = \boldsymbol{a}(\theta_0)/\|\boldsymbol{a}(\theta_0)\|^2;\ \hat{\boldsymbol{R}}_y(0) = \delta_y \boldsymbol{I}$

Update for each time instant $i = 1, \ldots, N$

STEP 1: Start:

$y(i) = \boldsymbol{w}_{ccm}^H(i-1)\boldsymbol{x}(i);\ e_y(i) = |y(i)|^2 - 1$

$\hat{\boldsymbol{R}}_y(i) = \lambda_y \hat{\boldsymbol{R}}_y(i-1) + e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)$

$\boldsymbol{g}_0(i) = \boldsymbol{a}(\theta_0) - \hat{\boldsymbol{R}}_y(i)\boldsymbol{v}_0(i);\ \boldsymbol{p}_1(i) = \boldsymbol{g}_0(i)$

STEP 2: For $k = 1, \ldots, K$**:**

$\alpha_k(i) = \left(\boldsymbol{p}_k^H(i)\hat{\boldsymbol{R}}_y(i)\boldsymbol{p}_k(i)\right)^{-1}\boldsymbol{g}_{k-1}^H(i)\boldsymbol{p}_k(i)$

$\boldsymbol{v}_k(i) = \boldsymbol{v}_{k-1}(i) + \alpha_k(i)\boldsymbol{p}_k(i)$

$\boldsymbol{g}_k(i) = \boldsymbol{g}_{k-1}(i) - \alpha_k(i)\hat{\boldsymbol{R}}_y(i)\boldsymbol{p}_k(i)$

$\beta_k(i) = \left[\boldsymbol{g}_{k-1}^H(i)\boldsymbol{g}_{k-1}(i)\right]^{-1}\left[\boldsymbol{g}_k^H(i)\boldsymbol{g}_k(i)\right]$

$\boldsymbol{p}_{k+1}(i) = \boldsymbol{g}_k(i) + \beta_k(i)\boldsymbol{p}_k(i)$

$k = k + 1$

STEP 3: After K **iterations:**

$\boldsymbol{v}_0(i+1) = \boldsymbol{v}_K(i)$

$\boldsymbol{w}_{ccm}(i) = \left[\boldsymbol{a}^H(\theta_0)\boldsymbol{v}_K(i)\right]^{-1}\gamma\boldsymbol{v}_K(i)$

$i = i + 1$

3.3.3 Proposed Modified Conjugate Gradient (MCG) Algorithms

The CCG algorithm proposed in the previous section operates K iterations per snapshot and runs the reset periodically for convergence. These operations increase the computational load in the sample-by-sample update. Here, we describe a modified CG (MCG) algorithm with only one iteration per snapshot. Compared with the existing methods, the proposed algorithm enforces the constraint with low complexity, avoids the matrix inversion and instability, and keeps fast convergence without the reset procedure.

The Proposed CMV-MCG Algorithm

The MCG algorithm was motivated from [76] for adaptive filtering. Here, we use this idea for the beamformer design. Note that the iteration number k is replaced by the snapshot number i since, in the proposed algorithm, only one iteration will be performed per snapshot. For simplicity, we will remove k in the subscript of terms.

The CG-based weight vector for the MCG algorithm is expressed by

$$\tilde{\boldsymbol{v}}_{\text{cmv}}(i) = \tilde{\boldsymbol{v}}_{\text{cmv}}(i-1) + \tilde{\alpha}_{\text{cmv}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i), \tag{3.17}$$

where $\tilde{\boldsymbol{p}}_{\text{cmv}}(i)$ is the direction vector at the ith snapshot, $\tilde{\alpha}_{\text{cmv}}(i)$ is the corresponding

coefficient, and, in what follows, all the quantities related to the proposed MCG algorithm are denoted by an over tilde.

From [76], one way to make the CG algorithm work with one iteration per snapshot is the application of the degenerated scheme, which means that $\tilde{\boldsymbol{g}}_{\text{cmv}}(i)$ is not orthogonal to the subspace spanned by $\tilde{\boldsymbol{p}}_{\text{cmv}}(l)$, where $l = 1, \ldots, i$, namely, the expanding subspace theorem [19, 74, 75] does not hold. Under this condition, we need to ensure that the coefficient $\tilde{\alpha}_{\text{cmv}}(i)$ satisfies the convergence bound [76, 77], which is given by

$$0 \leq \tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i) \leq 0.5\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1). \tag{3.18}$$

For deriving $\tilde{\alpha}_{\text{cmv}}(i)$, we consider a recursive expression for the negative gradient vector

$$\begin{aligned}\tilde{\boldsymbol{g}}_{\text{cmv}}(i) &= \boldsymbol{a}(\theta_0) - \hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{v}}_{\text{cmv}}(i) \\ &= (1-\lambda)\boldsymbol{a}(\theta_0) + \lambda\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1) \\ &\quad - \tilde{\alpha}_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i) - \boldsymbol{x}(i)\boldsymbol{x}^H(i)\tilde{\boldsymbol{v}}_{\text{cmv}}(i-1).\end{aligned} \tag{3.19}$$

Premultiplying (3.19) by $\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)$ yields

$$\begin{aligned}\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i) =& \lambda\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1) - \tilde{\alpha}_{\text{cmv}}(i)\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i) \\ &+ (1-\lambda)\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\boldsymbol{a}(\theta_0) - \tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\tilde{\boldsymbol{v}}_{\text{cmv}}(i-1).\end{aligned} \tag{3.20}$$

Taking the expectation of both sides and considering $\tilde{\boldsymbol{p}}_{\text{cmv}}(i)$ uncorrelated with $\boldsymbol{x}(i)$, $\boldsymbol{a}(\theta_0)$ and $\tilde{\boldsymbol{v}}_{\text{cmv}}(i-1)$ [76] yields

$$\begin{aligned}\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i)] \approx& \lambda\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)] - \mathbb{E}[\tilde{\alpha}_{\text{cmv}}(i)]\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i)] \\ &- \lambda\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\boldsymbol{a}(\theta_0)],\end{aligned} \tag{3.21}$$

where the optimal solution $\boldsymbol{R}\tilde{\boldsymbol{v}}_{\text{cmv,opt}} = \boldsymbol{a}(\theta_0)$ and $\mathbb{E}[\tilde{\boldsymbol{v}}_{\text{cmv}}(i-1) - \tilde{\boldsymbol{v}}_{\text{cmv,opt}}] \approx \boldsymbol{0}$ have been used with the assumption that the algorithm converges. Making a rearrangement of (3.21) and following the convergence bound (3.18), we obtain

$$\mathbb{E}[\tilde{\alpha}_{\text{cmv}}(i)] = \frac{\lambda\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1) - \tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\boldsymbol{a}(\theta_0)] - \mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i)]}{\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i)]}, \tag{3.22}$$

and

$$\begin{aligned} &\frac{(\lambda-0.5)\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)]-\lambda\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\boldsymbol{a}(\theta_0)]}{\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i)]} \leq \\ &\mathbb{E}[\tilde{\alpha}_{\text{cmv}}(i)] \leq \frac{\lambda\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)-\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\boldsymbol{a}(\theta_0)]}{\mathbb{E}[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i)]}. \end{aligned} \tag{3.23}$$

The inequalities in (3.23) are satisfied if we define

$$\begin{aligned} \tilde{\alpha}_{\text{cmv}}(i) =& \big[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i)\big]^{-1} \\ & \big\{\lambda\big[\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)-\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\boldsymbol{a}(\theta_0)\big]-\tilde{\eta}\tilde{\boldsymbol{p}}^H_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)\big\}, \end{aligned} \tag{3.24}$$

where $0 \leq \tilde{\eta} \leq 0.5$.

The direction vector $\tilde{\boldsymbol{p}}_{\text{cmv}}(i)$ is defined by

$$\tilde{\boldsymbol{p}}_{\text{cmv}}(i+1) = \tilde{\boldsymbol{g}}_{\text{cmv}}(i) + \tilde{\beta}_{\text{cmv}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i) \tag{3.25}$$

where $\tilde{\beta}_{\text{cmv}}(i)$ is computed for avoiding the reset procedure by employing the Polak-Ribiere approach (Eq. (25) in [76]), which is stated as

$$\tilde{\beta}_{\text{cmv}}(i) = \big[\tilde{\boldsymbol{g}}^H_{\text{cmv}}(i-1)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)\big]^{-1}\big[\tilde{\boldsymbol{g}}_{\text{cmv}}(i)-\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)\big]^H\tilde{\boldsymbol{g}}_{\text{cmv}}(i). \tag{3.26}$$

Until now, we derived the proposed MCG algorithm for the CMV criterion, whose weight solution is given by substituting the CG-based weight vector into (3.1), i.e.,

$$\tilde{\boldsymbol{w}}_{\text{cmv}}(i) = \big[\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{v}}_{\text{cmv}}(i)\big]^{-1}\gamma\tilde{\boldsymbol{v}}_{\text{cmv}}(i). \tag{3.27}$$

The proposed CMV-MCG algorithm is summarized in Table 3.3. Again, we remove the subscript "cmv" for compact expressions. Clearly, compared with (3.1), the weight solution in (3.27) ensures the constraint and solves the systems of equations without the matrix inversion and thus avoids numerical instability.

Tab. 3.3: The CMV-MCG algorithm

Initialization:

$\tilde{\boldsymbol{v}}(0) = \boldsymbol{0};\ \tilde{\boldsymbol{g}}(0) = \tilde{\boldsymbol{p}}(1) = \boldsymbol{a}(\theta_0);\ \hat{\boldsymbol{R}}(0) = \tilde{\delta}\boldsymbol{I}$

Update for each time instant $i = 1, \ldots, N$

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

$\tilde{\alpha}(i) = \left[\tilde{\boldsymbol{p}}^H(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}(i)\right]^{-1}\left\{\lambda\left[\tilde{\boldsymbol{p}}^H(i)\tilde{\boldsymbol{g}}(i-1) - \tilde{\boldsymbol{p}}^H(i)\boldsymbol{a}(\theta_0)\right] - \tilde{\eta}\tilde{\boldsymbol{p}}^H(i)\tilde{\boldsymbol{g}}(i-1)\right\}$
$(0 \le \tilde{\eta} \le 0.5)$

$\tilde{\boldsymbol{v}}(i) = \tilde{\boldsymbol{v}}(i-1) + \tilde{\alpha}(i)\tilde{\boldsymbol{p}}(i)$

$\tilde{\boldsymbol{g}}(i) = (1-\lambda)\boldsymbol{a}(\theta_0) + \lambda\tilde{\boldsymbol{g}}(i-1) - \tilde{\alpha}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}(i) - \boldsymbol{x}(i)\boldsymbol{x}^H(i)\tilde{\boldsymbol{v}}(i-1)$

$\tilde{\beta}(i) = [\tilde{\boldsymbol{g}}^H(i-1)\tilde{\boldsymbol{g}}(i-1)]^{-1}[\tilde{\boldsymbol{g}}(i) - \tilde{\boldsymbol{g}}(i-1)]^H\tilde{\boldsymbol{g}}(i)$

$\tilde{\boldsymbol{p}}(i+1) = \tilde{\boldsymbol{g}}(i) + \tilde{\beta}(i)\tilde{\boldsymbol{p}}(i)$

$\tilde{\boldsymbol{w}}_{\text{cmv}}(i) = [\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{v}}(i)]^{-1}\gamma\tilde{\boldsymbol{v}}(i)$

$i = i + 1$

Tab. 3.4: The CCM-MCG algorithm

Initialization:

$\tilde{\boldsymbol{v}}(0) = \boldsymbol{0};\ \tilde{\boldsymbol{g}}(0) = \tilde{\boldsymbol{p}}(1) = \boldsymbol{a}(\theta_0);\ \hat{\boldsymbol{R}}_y(0) = \tilde{\delta}_y\boldsymbol{I};\ \tilde{\boldsymbol{w}}_{\text{ccm}}(0) = \boldsymbol{a}(\theta_0)/\|\boldsymbol{a}(\theta_0)\|^2$

Update for each time instant $i = 1, \ldots, N$

$y(i) = \tilde{\boldsymbol{w}}_{\text{ccm}}^H(i-1)\boldsymbol{x}(i);\ e_y(i) = |y(i)|^2 - 1$

$\hat{\boldsymbol{R}}_y(i) = \lambda_y\hat{\boldsymbol{R}}_y(i-1) + e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)$

$\tilde{\alpha}(i) = \left[\tilde{\boldsymbol{p}}^H(i)\hat{\boldsymbol{R}}_y(i)\tilde{\boldsymbol{p}}(i)\right]^{-1}\left\{\lambda_y\left[\tilde{\boldsymbol{p}}^H(i)\tilde{\boldsymbol{g}}(i-1) - \tilde{\boldsymbol{p}}^H(i)\boldsymbol{a}(\theta_0)\right] - \tilde{\eta}\tilde{\boldsymbol{p}}^H(i)\tilde{\boldsymbol{g}}(i-1)\right\}$
$(0 \le \tilde{\eta} \le 0.5)$

$\tilde{\boldsymbol{v}}(i) = \tilde{\boldsymbol{v}}(i-1) + \alpha(i)\tilde{\boldsymbol{p}}(i)$

$\tilde{\boldsymbol{g}}(i) = (1-\lambda_y)\boldsymbol{a}(\theta_0) + \lambda_y\tilde{\boldsymbol{g}}(i-1) - \tilde{\alpha}(i)\hat{\boldsymbol{R}}_y(i)\tilde{\boldsymbol{p}}(i) - e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\tilde{\boldsymbol{v}}(i-1)$

$\tilde{\beta}(i) = [\tilde{\boldsymbol{g}}^H(i-1)\tilde{\boldsymbol{g}}(i-1)]^{-1}[\tilde{\boldsymbol{g}}(i) - \tilde{\boldsymbol{g}}(i-1)]^H\tilde{\boldsymbol{g}}(i)$

$\tilde{\boldsymbol{p}}(i+1) = \tilde{\boldsymbol{g}}(i) + \tilde{\beta}(i)\tilde{\boldsymbol{p}}(i)$

$\tilde{\boldsymbol{w}}_{\text{ccm}}(i) = [\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{v}}(i)]^{-1}\gamma\tilde{\boldsymbol{v}}(i)$

$i = i + 1$

The Proposed CCM-MCG Algorithm

Regarding the CCM criterion, $\tilde{\boldsymbol{w}}_{\text{ccm}}(0)$ needs to be initialized for the iteration procedure. Correspondingly, the negative gradient vector is given by

$$\begin{aligned}\tilde{\boldsymbol{g}}_{\text{ccm}}(i) =& (1-\lambda_y)\boldsymbol{a}(\theta_0) + \lambda_y\tilde{\boldsymbol{g}}_{\text{ccm}}(i-1) \\ & - \tilde{\alpha}_{\text{ccm}}(i)\hat{\boldsymbol{R}}_y(i)\tilde{\boldsymbol{p}}_{\text{ccm}}(i) - e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\tilde{\boldsymbol{v}}_{\text{ccm}}(i-1).\end{aligned} \tag{3.28}$$

Following the same derivation as for the CMV criterion, we will get the CCM-MCG algorithm, which is summarized in Table 3.4. Comparing with the CCG algorithm, the MCG is a non-reset and low complexity algorithm with one iteration per snapshot.

3.4 Analysis of the Proposed Methods

In this section, we investigate some of the characteristics of the proposed CCG and MCG algorithms. Specifically, we first determine the convexity condition for the global convergence of the CCM criterion. Then, the complexity requirements for the proposed algorithms are considered and compared with the SG [18], RLS [18], MSWF [27], and AVF [31] methods. At last, we analyze the convergence properties of the CG-based weight vector.

3.4.1 Global Convergence and Properties

Here, we focus on the analysis of the CCM criterion in (2.4), which is a fourth-order function with an elaborate structure that contains undesired local minima. We show in Appendix B that the convexity of the CCM cost function can be enforced by properly selecting the constant γ. Therefore, the global convergence for the constrained adaptive algorithms can be guaranteed.

3.4.2 Computational Complexity

We detail the computational complexity of the proposed and analyzed algorithms. We remark that the complexity is estimated by taking into account the number of additions and multiplications required by the algorithms for each snapshot. The comparison of the complexity for different algorithms is listed in Table 3.5, where m is the number of sensor elements and r is the rank for the reduced-rank algorithms.

Tab. 3.5: Comparison of the computational complexity

Algorithm	Additions	Multiplications
CMV-SG	$4m-2$	$4m+3$
CCM-SG	$4m$	$4m+7$
CMV-CCG	$K(m^2+4m-2)+2m^2-1$	$K(m^2+4m+1)+3m^2+3m$
CCM-CCG	$K(m^2+4m-2)+2m^2+m-2$	$K(m^2+4m+1)+3m^2+5m$
CMV-MCG	$2m^2+7m-3$	$3m^2+9m+4$
CCM-MCG	$2m^2+8m-3$	$3m^2+11m+5$
CMV-RLS	$4m^2-m-1$	$5m^2+5m-1$
CCM-RLS	$4m^2-m$	$5m^2+5m+2$
CMV-MSWF	$(r-1)m^2+rm+m+4r^2-2r-2$	$(r-1)m^2+2rm+5r^2+5r$
CCM-MSWF	$(r-1)m^2+rm+m-4r^2-2r-1$	$(r-1)m^2+2rm+5r^2+5r+3$
AVF	$r(4m^2+m-2)+5m^2-m-1$	$r(5m^2+3m)+8m^2+2m$

It is obvious that the complexity of the proposed CCG algorithms depends on the number of iterations K. For the case of the MCG, the complexity is lower than the other studied algorithms except the SG, which sacrifices the performance as a trade-off. Compared with the RLS method, the complexity of the MCG algorithm reduces significantly if m (e.g., $m = 60$) is large for some applications such as those found in sonar or radar. The reduction is not visible if m is small, e.g., for wireless communications. However, the proposed algorithm provides an efficient way for the beamformer design and avoids numerical instability that occurs in the RLS method.

3.4.3 Convergence Analysis

We analyze the convergence behavior of the CG-based weight vector. The convergence analysis was detailed in [74], in which the analysis was derived by considering an important property of the Krylov subspace [19]. Here, we use a more direct way to prove this property and simplify the derivation.

According to the CG algorithm, the direction vector $\boldsymbol{p}_{k+1}(i)$ at the kth iteration for the ith snapshot is constructed by the residual $\boldsymbol{g}_k(i)$ and subtracting out any components that are not the conjugacy with the previous $\boldsymbol{p}_k(i)$. In other words, the direction vectors are built from the residuals. Thus, the subspace spanned by the residuals is equal to the subspace spanned by the direction vectors.

On the other hand, we know that $\boldsymbol{g}_k(i)$ of the CCG algorithm with respect to the CMV criterion is a linear combination of the previous residuals and $\hat{\boldsymbol{R}}(i)\boldsymbol{p}_k(i)$. If defining $\boldsymbol{\mathcal{S}}_k(i)$ as the subspace spanned by the direction vectors and recalling $\boldsymbol{p}_{k+1}(i) \in \boldsymbol{\mathcal{S}}_k(i)$ implies that each new $\boldsymbol{\mathcal{S}}_{k+1}(i)$ is formed from the previous $\boldsymbol{\mathcal{S}}_k(i)$ and $\hat{\boldsymbol{R}}(i)\boldsymbol{\mathcal{S}}_k(i)$, we have

$$\begin{aligned} \boldsymbol{\mathcal{S}}_k(i) &= \text{span}\{\boldsymbol{p}_1(i), \hat{\boldsymbol{R}}(i)\boldsymbol{p}_1(i), \ldots, \hat{\boldsymbol{R}}^{k-1}(i)\boldsymbol{p}_1(i)\} \\ &= \text{span}\{\boldsymbol{g}_0(i), \hat{\boldsymbol{R}}(i)\boldsymbol{g}_0(i), \ldots, \hat{\boldsymbol{R}}^{k-1}(i)\boldsymbol{g}_0(i)\}. \end{aligned} \tag{3.29}$$

As we know, the residual vector can be written as

$$\boldsymbol{g}_k(i) = \boldsymbol{a}(\theta_0) - \hat{\boldsymbol{R}}(i)\boldsymbol{v}_k(i) = \hat{\boldsymbol{R}}(i)\boldsymbol{\varrho}_k(i), \tag{3.30}$$

where $\boldsymbol{\varrho}_k(i) = \boldsymbol{v}_{\text{opt}}(i) - \boldsymbol{v}_k(i)$ is the CG-based weight error at the kth iteration and $\boldsymbol{v}_{\text{opt}}(i)$ is the optimal solution at the ith snapshot. According to (3.30), the second expression in

(3.29) is given by

$$\boldsymbol{\mathcal{S}}_k(i) = \text{span}\{\hat{\boldsymbol{R}}(i)\boldsymbol{\varrho}_0(i), \hat{\boldsymbol{R}}^2(i)\boldsymbol{\varrho}_0(i), \ldots, \hat{\boldsymbol{R}}^k(i)\boldsymbol{\varrho}_0(i)\}, \tag{3.31}$$

which is the well-known Krylov subspace [19]. For a fixed k, this subspace holds an important property, which is

$$\boldsymbol{\varrho}_{k+1}(i) = \big(\boldsymbol{I} + \sum_{j=1}^{k} \psi_j(i)\hat{\boldsymbol{R}}^j(i)\big)\boldsymbol{\varrho}_0(i), \tag{3.32}$$

where $\boldsymbol{I}$ is an identity matrix and the coefficient $\psi_j(i)$ is a function of $\alpha_l(i)$, where $l = j, \ldots, k+1$, and $\beta_{l'}(i)$ with $l' = j, \ldots, k$. This property has been verified in [75]. Here, we use an alternative way to get it. Substituting (3.12) and (3.13) into $\boldsymbol{v}_{k+1}(i)$ iteratively, we get

$$\begin{aligned}
\boldsymbol{v}_{k+1}(i) &= \boldsymbol{v}_0(i) + \sum_{j=1}^{k+1} \alpha_j(i)\boldsymbol{p}_j(i) \\
&= \boldsymbol{v}_0(i) + \sum_{j=1}^{k+1} \alpha_j(i)\big[\boldsymbol{g}_{j-1}(i) + \beta_{j-1}(i)\boldsymbol{g}_{j-2}(i) + \ldots \\
&\quad + \beta_{j-1}(i)\ldots\beta_2(i)\boldsymbol{g}_1(i) + \beta_{j-1}(i)\ldots\beta_1(i)\boldsymbol{p}_1(i)\big] \\
&= \boldsymbol{v}_0(i) + \big\{L_{g_k}(i)\boldsymbol{g}_k(i) + L_{g_{k-1}}\boldsymbol{g}_{k-1}(i) + \ldots \\
&\quad + L_{g_1}(i)\boldsymbol{g}_1(i) + L_{p_1}(i)\boldsymbol{p}_1(i)\big\},
\end{aligned} \tag{3.33}$$

where
$L_{g_k}(i) = \alpha_{k+1}(i)$;
$L_{g_{k-1}}(i) = \alpha_k(i) + \alpha_{k+1}(i)\beta_k(i)$;
$L_{g_1}(i) = \alpha_2(i) + \alpha_3(i)\beta_2(i) + \alpha_4(i)\beta_3(i)\beta_2(i) + \ldots + \alpha_{k+1}(i)\beta_k(i)\beta_{k-1}(i)\ldots\beta_2(i)$;
$L_{p_1}(i) = \alpha_1(i) + \alpha_2(i)\beta_1(i) + \alpha_3(i)\beta_2(i)\beta_1(i) + \ldots + \alpha_{k+1}(i)\beta_k(i)\beta_{k-1}(i)\ldots\beta_1(i)$.

In (3.33), the coefficients $L_{g_l}(i)$ for $l = 1, \ldots, k$ are constants and $\boldsymbol{p}_1(i) = \boldsymbol{g}_0(i)$. Thus, this implies that $\big\{\boldsymbol{g}_0(i), \ldots, \boldsymbol{g}_k(i)\big\} \in \boldsymbol{\mathcal{S}_k}(i)$. Subtracting (3.33) from $\boldsymbol{v}_{\text{opt}}(i)$ and combining the expressions in (3.30), (3.31), and (3.33), we obtain

$$\boldsymbol{\varrho}_{k+1}(i) = \boldsymbol{\varrho}_0(i) + \sum_{j=1}^{k} \psi_j(i)\hat{\boldsymbol{R}}^j(i)\boldsymbol{\varrho}_0(i), \tag{3.34}$$

where $\psi_j(i)$ has been defined in (3.32). Making a rearrangement leads to (3.32).

The importance of the expression (3.32) is to measure the error energy norm

$\|\boldsymbol{\varrho}_{k+1}(i)\|_{\hat{\boldsymbol{R}}(i)} = \left(\boldsymbol{\varrho}_{k+1}^H(i)\hat{\boldsymbol{R}}(i)\boldsymbol{\varrho}_{k+1}(i)\right)^{1/2}$ for the convergence analysis. The expression in parentheses of (3.32) can be written as a polynomial $P_k(\hat{\boldsymbol{R}}(i))$ of degree k [75]. Then, we have

$$\boldsymbol{\varrho}_{k+1}(i) = P_k(\hat{\boldsymbol{R}}(i))\boldsymbol{\varrho}_0(i), \tag{3.35}$$

where $\boldsymbol{\varrho}_0(i)$ can be defined as a linear combination of distinct eigenvectors of $\hat{\boldsymbol{R}}(i)$, which yields $\boldsymbol{\varrho}_0(i) = \sum_j \xi_j(i)\boldsymbol{z}_j(i)$, where $\xi_j(i)$ are scalars not all zero and j is the index that corresponds to the number of the eigenvectors $\boldsymbol{z}_j(i)$. If we notice that $P_k(\hat{\boldsymbol{R}}(i))\boldsymbol{z}_j(i) = P_k(\vartheta_j(i))\boldsymbol{z}_j(i)$ [75], (3.35) can be expressed as

$$\boldsymbol{\varrho}_{k+1}(i) = \sum_j \xi_j(i)P_k(\vartheta_j(i))\boldsymbol{z}_j(i), \tag{3.36}$$

and so

$$\|\boldsymbol{\varrho}_{k+1}(i)\|^2_{\hat{\boldsymbol{R}}(i)} = \sum_j \xi_j^2(i)P_k^2(\vartheta_j(i))\vartheta_j(i), \tag{3.37}$$

where $\vartheta_j(i)$ is the eigenvalue corresponding to $\boldsymbol{z}_j(i)$.

The proposed CCG algorithm tries to find the polynomial $P_k(\vartheta_j(i))$ that minimizes (3.37) for the convergence, which should be fast even with the worst eigenvector that maximizes the terms on the right hand side of (3.37). Therefore, it implies

$$\begin{aligned}\|\boldsymbol{\varrho}_{k+1}(i)\|^2_{\hat{\boldsymbol{R}}(i)} &\leq \min_{P_k} \max_{\vartheta(i)\in\Lambda(\hat{\boldsymbol{R}}(i))} P_k^2(\vartheta(i)) \sum_j \xi_j^2(i)\vartheta_j(i) \\ &= \min_{P_k} \max_{\vartheta(i)\in\Lambda(\hat{\boldsymbol{R}}(i))} P_k^2(\vartheta(i))\|\boldsymbol{\varrho}_0(i)\|^2_{\hat{\boldsymbol{R}}(i)},\end{aligned} \tag{3.38}$$

where $\Lambda(\hat{\boldsymbol{R}}(i))$ is the set of the eigenvalues of $\hat{\boldsymbol{R}}(i)$. In order to analyze the connection between the error energy norm and the eigenvalues in (3.38), we employ the Chebyshev polynomials, which yields [75]

$$P_k(\vartheta(i)) = \frac{T_k\left(\frac{\vartheta_{max}(i)+\vartheta_{min}(i)-2\vartheta(i)}{\vartheta_{max}(i)-\vartheta_{min}(i)}\right)}{T_k\left(\frac{\vartheta_{max}(i)+\vartheta_{min}(i)}{\vartheta_{max}(i)-\vartheta_{min}(i)}\right)}, \tag{3.39}$$

where $T_k(\omega) = \frac{1}{2}[(\omega+\sqrt{\omega^2-1})^k + (\omega-\sqrt{\omega^2-1})^k]$ denotes the Chebyshev polynomials of degree k. This polynomial obeys the oscillating property of Chebyshev polynomials on the domain $\vartheta_{min}(i) \leq \vartheta(i) \leq \vartheta_{max}(i)$ [74].

Since the maximum value of the numerator of (3.39) is one, we have

$$\begin{aligned}\|\boldsymbol{\varrho}_{k+1}(i)\|_{\hat{\boldsymbol{R}}(i)} &= \left(\boldsymbol{\varrho}_{k+1}^H(i)\hat{\boldsymbol{R}}(i)\boldsymbol{\varrho}_{k+1}(i)\right)^{1/2} \\ &\leq T_k\left(\frac{\vartheta_{max}(i)+\vartheta_{min}(i)}{\vartheta_{max}(i)-\vartheta_{min}(i)}\right)^{-1}\|\boldsymbol{\varrho}_0(i)\|_{\hat{\boldsymbol{R}}(i)} \\ &= T_k\left(\frac{\kappa(i)+1}{\kappa(i)-1}\right)^{-1}\|\boldsymbol{\varrho}_0(i)\|_{\hat{\boldsymbol{R}}(i)} \\ &= 2\left[\left(\frac{\sqrt{\kappa(i)}+1}{\sqrt{\kappa(i)}-1}\right)^k+\left(\frac{\sqrt{\kappa(i)}-1}{\sqrt{\kappa(i)}+1}\right)^k\right]^{-1}\|\boldsymbol{\varrho}_0(i)\|_{\hat{\boldsymbol{R}}(i)},\end{aligned} \tag{3.40}$$

where $\kappa(i) = \vartheta_{max}(i)/\vartheta_{min}(i)$ is the condition number. The second term inside the brackets tends to zero as k increases, so the CCG algorithm convergence is governed by [74]

$$\|\boldsymbol{\varrho}_{k+1}(i)\|_{\hat{\boldsymbol{R}}(i)} \leq 2\left(\frac{\sqrt{\kappa(i)}-1}{\sqrt{\kappa(i)}+1}\right)^k\|\boldsymbol{\varrho}_0(i)\|_{\hat{\boldsymbol{R}}(i)}. \tag{3.41}$$

In conclusion, the convergence behavior of the CG-based weight vector is related to the CG-based weight error $\boldsymbol{\varrho}_0(i)$ and the condition number $\kappa(i)$, which should oscillate around $\kappa(i) = 1$ for the optimal solution, i.e., the convergence is finished in one iteration.

3.5 Simulations

In this section, we assess the effectiveness of the proposed algorithms over existing methods with the CMV and the CCM criteria via computer simulations. The simulations are carried out under both stationary and non-stationary scenarios by a ULA containing m sensor elements with half-wavelength spacing. For each experiment, 1000 runs are executed to get the curves. In all simulations, the desired signal power is $\sigma_0^2 = 1$ and the noise is spatially and temporally white Gaussian. The BPSK modulation scheme is employed and $\gamma = 1$ is set to satisfy the condition for the convexity of the CCM criterion.

We compare the proposed algorithms with the SG [14], RLS [18], MSWF [28], and AVF [29] methods according to the CMV criterion by showing the output signal-to-interference-plus-noise ratio (SINR) versus the input signal-to-noise ratio (SNR). The ULA is equipped with $m = 10$ sensor elements. We consider that the system has $q_{\text{inf}} = 3$ interferers, which have the same power of the desired user. The forgetting factor is $\lambda = \lambda_f = 0.998$ and the coefficients $\eta = \eta_y = 0.49$ since they lead to the best

performance. The regularization parameters are $\delta = 0.002$ for the CCG algorithm and $\tilde{\delta} = 0.001$ for the MCG algorithm. The iteration number is $K = m/2$ for the CCG algorithm. In Fig. 3.1, the output SINRs of the studied algorithms are very close except for the SG method. The MSWF and RLS algorithms show superior performance over the other methods. The AVF converges faster than the other methods. A general shortcoming of these algorithms is the high computational cost. Conversely, the proposed MCG algorithm converges quickly and reaches a comparable high performance with low complexity. It is worth noting that the MSWF and AVF algorithms do not show advantages in the current scenario since they are more suitable to the large array (e.g., $m \geq 30$) scenarios [27, 28, 65].

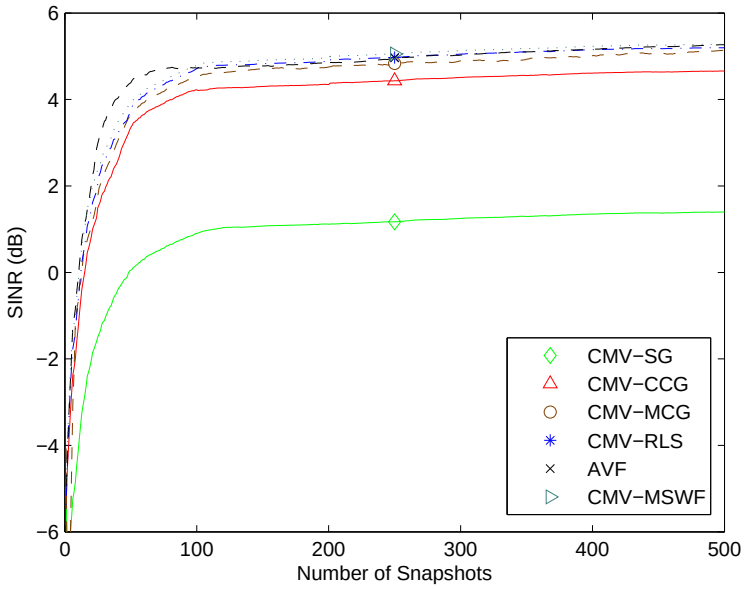

Fig. 3.1: Output SINR versus the number of snapshots with $q = 4$ users and $m = 10$ sensor elements.

Fig. 3.2 compares the proposed algorithms with the SG and RLS methods according to the design criteria. There are $q_{\text{inf}} = 5$ interferers in the system with one 5 dB above the power of the desired user, one with the same power of the desired user, and three -0.5 dB below the desired power. The CCM weight vectors $\boldsymbol{w}_{\text{ccm}}(0)$ and $\tilde{\boldsymbol{w}}_{\text{ccm}}(0)$ are initialized for keeping the constraint and running the iteration. The number of snapshots is fixed $N = 1000$. It is observed that the SINRs of the RLS and proposed algorithms increase with the increase of the input SNR, whereas the SG results show only a small improvement. This is explained by the fact that SG methods are subject to the eigenvalue spread of the covariance matrix of the received vector. The proposed MCG curves approach the RLS ones but with lower-complexity. Also, it is evident that the performance of the MCG

algorithms is better than that of the CCG methods.

It is shown in Fig. 3.2 that the adaptive algorithms for the CCM criterion achieve superior performance to those with the CMV criterion. For efficient presentation and convenience, we only illustrate the simulation results for the CCM criterion in the following simulations.

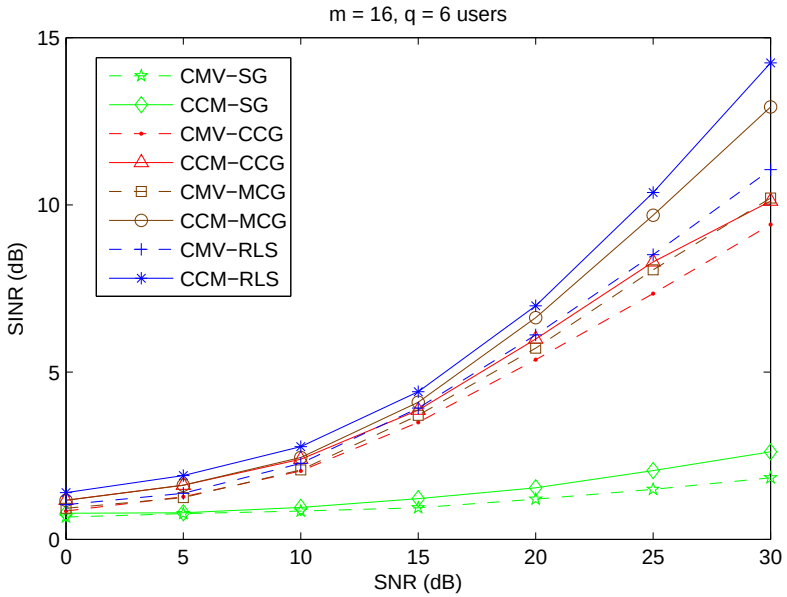

Fig. 3.2: Output SINR versus input SNR for the proposed algorithms with $q = 6$ users and $m = 16$ sensor elements.

In Fig. 3.3, we illustrate the performance of the proposed algorithms with an increasing number of users. We consider the input SNR$=$ 20 dB, interference-to-noise ratio (INR)$=$ 20 dB, and $N = 1000$. The fact that the number of the interferers increases, deteriorates the output SINR of all the algorithms. However, the results of the proposed algorithms are still in good match with that of the RLS method. As the number of the interferers reach a reasonably large value, the performance of the new algorithms is close to the RLS, which shows that the proposed algorithms are robust in a severe environment.

Fig. 3.4 shows the beampatterns of the array of the existing and proposed algorithms. The DOA of the desired user is $\theta_0 = 50^o$. There are five interferers with one 5 dB ($\theta_1 = 40^o$), one 0 dB ($\theta_2 = 70^o$), and three -5 dB ($\theta_3 = 20^o$, $\theta_4 = 30^o$, and $\theta_5 = 60^o$) above the desired power. The input SNR$=$ 20 dB and the number of snapshots $N = 1000$. From Fig. 3.4, the mainlobe beams of the studied algorithms direct at the direction of the desired user. The proposed algorithms have nulls (arrows in Fig. 3.4) at the directions of arrival of the interferers, especially for the MCG method, which forms the nulls as deep

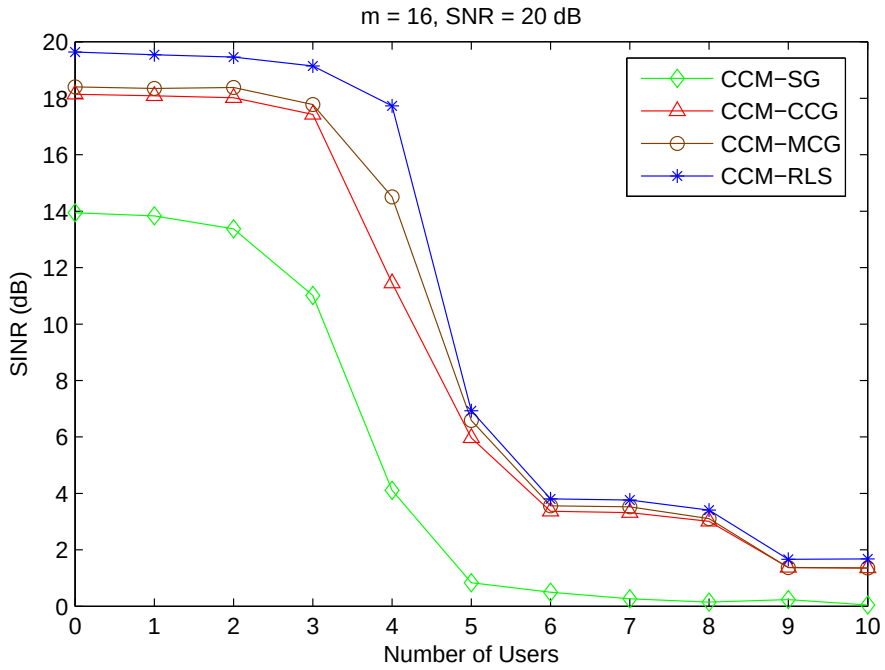

Fig. 3.3: Output SINR versus the number of users (q) for the proposed algorithms with $m = 16$ sensor elements.

as those of the RLS method.

The mismatch (steering vector error) condition is analyzed in Fig. 3.5, which includes two experiments. Fig. 3.5(a) shows the output SINR of each method versus the number of snapshots with the known DOA of the desired user. The system works under the same condition as that in Fig. 3.2 with SNR$= 20$ dB. The results demonstrate that the proposed algorithms converge faster and have better output SINR than the SG algorithm. The performance under the mismatch scenario is given in Fig. 3.5(b). The estimated DOA of the desired user is a constant value 1^0 away from the actual direction. It indicates that the mismatch problem induces a worse performance to all the algorithms. The convergence rate of all the methods reduces whereas the devised algorithms are more robust to this mismatch compared with the SG method and work with lower computational complexity compared with RLS method, especially for the MCG algorithm, whose curve reaches the steady-state rapidly and is very close to that of the RLS method.

In the last experiment, we evaluate the performance of the proposed and analyzed algorithms in a non-stationary scenario, namely, when the number of users change.

In Fig. 3.6, the system starts with $q_{\text{inf}} = 5$ interferers, one with the same power of the desired user and the rest -0.5 dB below the desired power. Two more users with one 5

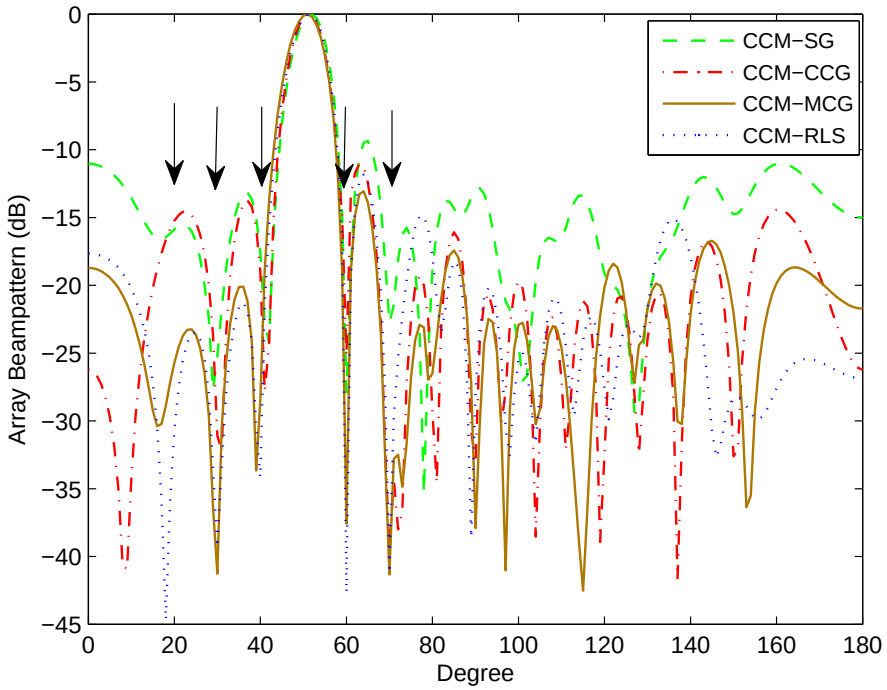

Fig. 3.4: Array beampattern versus degree for the proposed algorithms with $m = 16$ sensor elements.

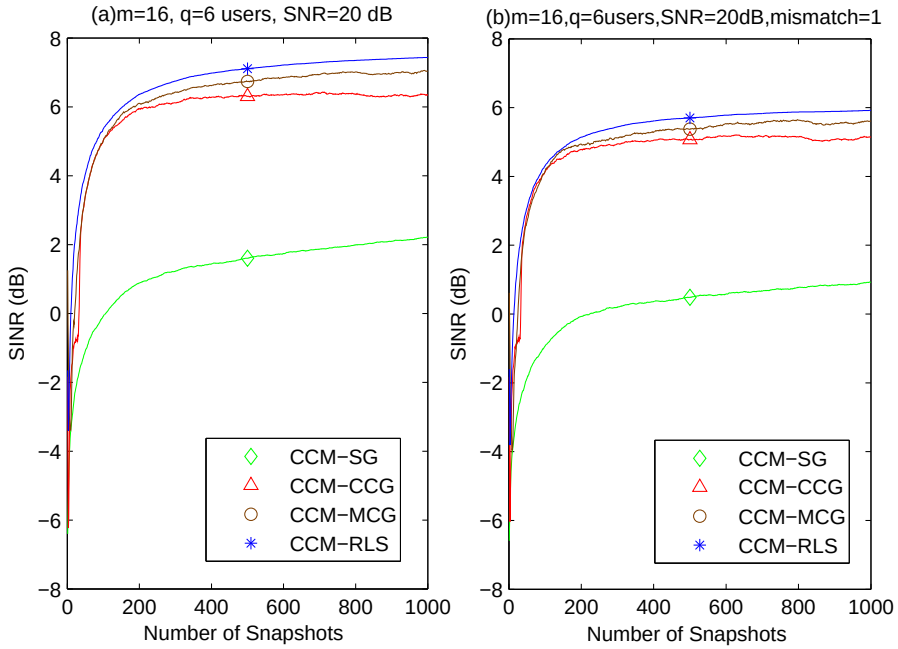

Fig. 3.5: Output SINR versus the number of snapshots for (a) ideal steering vector condition. (b) steering vector with mismatch.

dB above the desired power and one -0.5 dB below the desired power enter the system at the 1000th snapshot. The coefficients are set to the same values as those in the first experiment. From Fig. 3.6, we see that the SINR performance of the algorithms degrades at $N = 1000$. Note that we set the SINR values of all the methods at $N = 1000$ around -10 dB to show the convergence behavior. The proposed algorithms rapidly track the change and recover to a steady-state. The MCG algorithm recovers quickly and achieves a better solution. The CCM-RLS method achieves the best output but with a relatively slow response at the second stage.

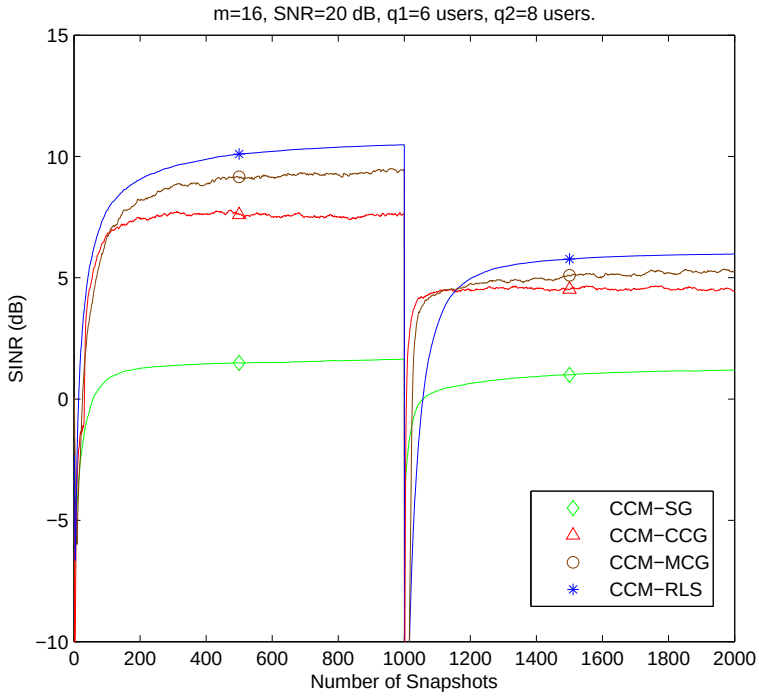

Fig. 3.6: Output SINR versus the number of snapshots in a scenario where additional interferers suddenly enter and/or leave the system.

Fig. 3.7 depicts the coefficients $\tilde{\alpha}_{\text{ccm}}(i)$ and $\tilde{\beta}_{\text{ccm}}(i)$ of the proposed MCG algorithm in the non-stationary scenario, respectively. Both $\tilde{\alpha}_{\text{ccm}}(i)$ and $\tilde{\beta}_{\text{ccm}}(i)$ are close to zero in the steady-state condition since, for $\tilde{\alpha}_{\text{ccm}}(i)$, according to $\tilde{\boldsymbol{v}}_{\text{ccm}}(i) = \tilde{\boldsymbol{v}}_{\text{ccm}}(i-1) + \tilde{\alpha}_{\text{ccm}}(i)\tilde{\boldsymbol{p}}_{\text{ccm}}(i)$, $\tilde{\alpha}_{\text{ccm}}(i) = 0$ means that $\tilde{\boldsymbol{v}}_{\text{ccm}}(i) = \tilde{\boldsymbol{v}}_{\text{ccm}}(i-1)$, which proves the convergence, and for $\tilde{\beta}_{\text{ccm}}(i)$, according to $\tilde{\beta}_{\text{ccm}}(i) = \left[\tilde{\boldsymbol{g}}_{\text{ccm}}^H(i-1)\tilde{\boldsymbol{g}}_{\text{ccm}}(i-1)\right]^{-1}\left[\tilde{\boldsymbol{g}}_{\text{ccm}}(i) - \tilde{\boldsymbol{g}}_{\text{ccm}}(i-1)\right]^H\tilde{\boldsymbol{g}}_{\text{ccm}}(i)$, the residual vector $\tilde{\boldsymbol{g}}_{\text{ccm}}(i)$ will be close to zero after the algorithm converges and so $\tilde{\beta}_{\text{ccm}}(i) \rightarrow 0$. The only interruptions for both figures occur when the extra interferers come into the system, which verifies the adaptability of the coefficients.

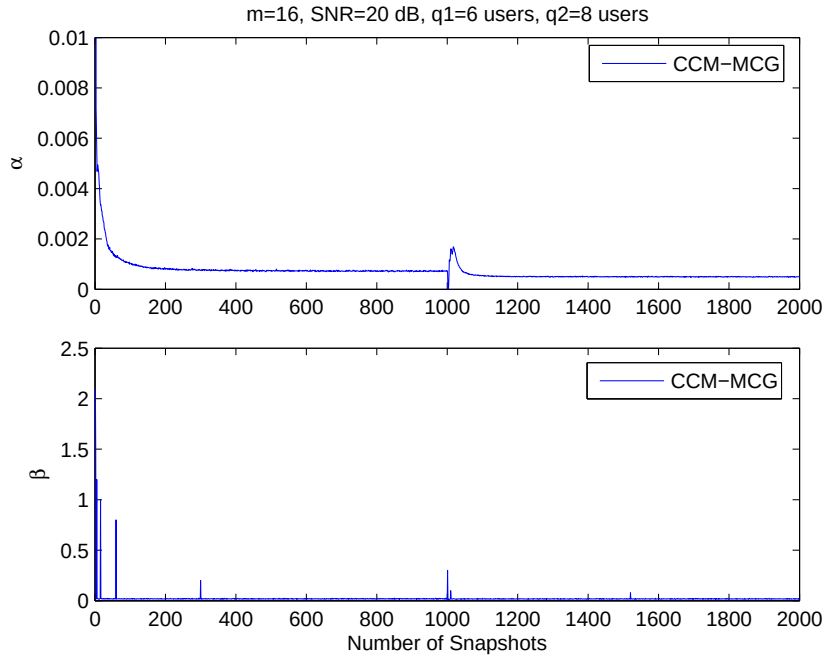

Fig. 3.7: Step size values α and β of the proposed CCM-MCG algorithm in a scenario where additional interferers suddenly enter and/or leave the system.

3.6 Conclusions

In this chapter, we introduced CG-based adaptive algorithms with the CMV and CCM criteria for beamforming. We use a CG-based weight vector to create a relation between the constrained system of equation and the weight expression. A complexity comparison was given for illustrating the advantage of the proposed algorithms over the existing ones. The CCM convexity property was established and a convergence analysis for the CG-based weight vector was derived. Simulation results showed that the proposed algorithms achieve fast convergence and tracking abilities with low complexity in the studied scenarios.

4. ADAPTIVE REDUCED-RANK CMV BEAMFORMING AND DOA ALGORITHMS BASED ON JOINT ITERATIVE OPTIMIZATION OF FILTERS

Contents

In this chapter, we introduce CMV reduced-rank algorithms based on joint iterative optimization (JIO) of filters for beamforming and DOA estimation. This chapter can be divided into two parts. In the first part (section 4.1-4.7), we introduce the JIO scheme and use it to develop CMV adaptive algorithms for beamforming. In the second part (section 4.8-4.12), we employ the JIO scheme to derive a DOA estimation algorithm.

4.1 Introduction for Beamforming

The adaptive array processing algorithms we introduced in the previous chapters are full-rank methods. The common drawback of the full-rank methods is that they require a large amount of samples to reach the steady-state when the number of elements in the

filter is large. Furthermore, in dynamic scenarios, filters with many elements usually fail or provide poor performance in tracking signals embedded in interference and noise.

Reduced-rank signal processing was originally motivated to provide a way out of this dilemma [21]- [25]. For the application of beamforming, reduced-rank schemes project the received vector onto a lower dimensional subspace and perform the filter optimization within this subspace. One of the popular reduced-rank schemes is MSWF, which employs MMSE [26], and its extended versions that utilize the CMV and CCM criteria were reported in [27, 50]. Another technique that resembles the MSWF is the AVF [29, 30] , which generates the same Krylov subspace as proved in [65, 80]. Despite the improved convergence and tracking performance achieved by these methods, they require high computational cost and suffer from numerical problems.

We propose CMV reduced-rank algorithms based on constrained joint iterative optimization (JIO) of filters for beamforming. The proposed scheme, whose initial results were reported in [81], jointly optimizes a transformation matrix and a reduced-rank filter that operates at the output of the transformation matrix. The essence of the proposed approach is to change the role of adaptive CMV filters. The bank of adaptive filters is responsible for performing dimensionality reduction, whereas the reduced-rank filter effectively forms the beam in the direction of the SOI. We describe the CMV expressions for the design of the transformation matrix and the reduced-rank filter and present SG and RLS algorithms for efficiently implementing the method. We also introduce an automatic rank estimation algorithm for determining the most adequate rank for the proposed algorithms. An analysis of the stability and the convergence properties of the proposed algorithms is presented and semi-analytical expressions are derived for predicting their performance.

4.2 Problem Statement

The CMV optimization problem has been given in (2.3) and the weight solution in (3.1), which can be estimated via the SG or RLS algorithms. However, the laws that govern their convergence and tracking behaviors imply that they depend on the number of elements m and on the eigenvalue spread of the input covariance matrix $\boldsymbol{R}$.

A reduced-rank algorithm must extract the most important features of the processed data by performing dimensionality reduction. This mapping is carried out by a transfor-

mation matrix $\boldsymbol{T}_r \in \mathbb{C}^{m\times r}$ with $r \leq m$ on the received data as given by

$$\bar{\boldsymbol{x}}(i) = \boldsymbol{T}_r^H(i)\boldsymbol{x}(i), \tag{4.1}$$

where, in what follows, all r dimensional quantities are denoted with a over "bar". The resulting projected received vector $\bar{\boldsymbol{x}}(i)$ is the input to a filter represented by $\bar{\boldsymbol{w}}(i) = [\bar{w}_1(i), \bar{w}_2(i), \ldots, \bar{w}_r(i)]^T \in \mathbb{C}^{r\times 1}$. The filter output is

$$y(i) = \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{x}}(i). \tag{4.2}$$

The reduced-rank filter $\bar{\boldsymbol{w}}(i)$ is designed according to minimizing the following cost function

$$J_{\text{mv}}\big(\bar{\boldsymbol{w}}(i)\big) = \mathbb{E}\big[|\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{x}}(i)|^2\big], \quad \text{subject to } \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{a}}(\theta_0) = \gamma, \tag{4.3}$$

where $\bar{\boldsymbol{a}}(\theta_0) = \boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0)$ is the reduced-rank steering vector and $\gamma = 1$ is set.

The weight solution of the above problem is

$$\bar{\boldsymbol{w}}_{\text{opt}} = \frac{\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta_0)}{\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta_0)}, \tag{4.4}$$

where $\bar{\boldsymbol{R}} = \mathbb{E}[\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)] = \boldsymbol{T}_r^H\boldsymbol{R}\boldsymbol{T}_r$ is the reduced-rank covariance matrix. The associated MV for a CMV filter with rank r is

$$\text{MV} = \frac{1}{\boldsymbol{a}^H(\theta)\boldsymbol{T}_r\big(\boldsymbol{T}_r^H\boldsymbol{R}\boldsymbol{T}_r\big)^{-1}\boldsymbol{T}_r^H\boldsymbol{a}(\theta_0)}. \tag{4.5}$$

The above development shows that the main problem is how to cost-effectively design $\boldsymbol{T}_r$ to perform dimensionality reduction on $\boldsymbol{x}(i)$, resulting in an improved convergence and tracking performance over the full-rank filter. In Appendix D, we provide a necessary and sufficient condition for $\boldsymbol{T}_r$ to preserve the MV of optimal full-rank filter and discuss the existence of multiple solutions. In the following, we detail our proposed reduced-rank method.

4.3 Proposed Reduced-rank Method

The proposed JIO scheme, depicted in Fig. 4.1, employs a matrix $\boldsymbol{T}_r(i)$ to perform dimensionality reduction on a data vector $\boldsymbol{x}(i)$. The reduced-rank filter $\bar{\boldsymbol{w}}(i)$ processes

the reduced-rank data vector $\bar{\boldsymbol{x}}(i)$ in order to yields scalar estimate $y(i)$. The transformation matrix $\boldsymbol{T}_r(i)$ and the reduced-rank filter $\bar{\boldsymbol{w}}(i)$ are jointly optimized in the proposed scheme according to the MV criterion subject to a constraint that ensures that the reduced-rank array response is equal to unity in the direction of the SOI.

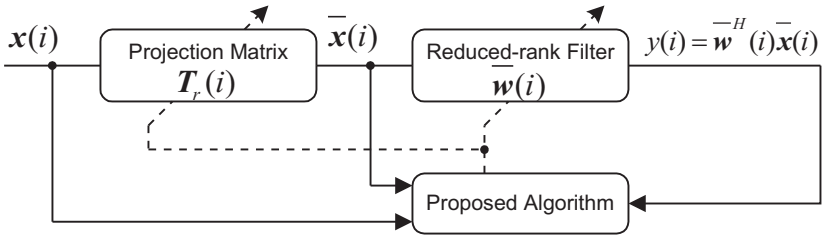

Fig. 4.1: The JIO reduced-rank structure.

The transformation matrix is structured as a bank of r full-rank filters $\boldsymbol{t}_j(i) = [t_{1,j}(i), t_{2,j}(i), \ldots, t_{m,j}(i)]^T \in \mathbb{C}^{m\times 1}$, $(j = 1, \ldots, r)$ as given by $\boldsymbol{T}_r(i) = [\boldsymbol{t}_1(i), \boldsymbol{t}_2(i), \ldots, \boldsymbol{t}_r(i)]$. The output $y(i)$ of the proposed reduced-rank scheme can be expressed as a function of the received vector $\boldsymbol{x}(i)$, the transformation matrix $\boldsymbol{T}_r(i)$, and the reduced-rank filter $\bar{\boldsymbol{w}}(i)$:

$$y(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i) = \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{x}}(i). \tag{4.6}$$

It is interesting to note that for $r = 1$, the proposed scheme becomes a conventional full-rank CMV filtering scheme with an additional weight parameter w_r that provides an amplitude gain. For $r > 1$, the signal processing tasks are changed and the full-rank CMV filters compute a subspace projection and the reduced-rank filter provides a unity gain in the direction of the SOI. This rationale is fundamental to the exploitation of the low-rank nature of signals in typical beamforming scenarios.

The CMV expressions for the filters $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}(i)$ can be computed via minimizing

$$\begin{aligned} J_{\text{JIO-MV}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)\big) &= \mathbb{E}\big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)|^2\big] = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{R}\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i), \\ &\text{subject to } \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = 1. \end{aligned} \tag{4.7}$$

In order to solve the above problem, we resort to the method of Lagrange multipliers and transform the constrained optimization into an unconstrained one expressed by

$$L_{\text{un}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)\big) = \mathbb{E}\big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)|^2\big] + 2\,\Re\big[\lambda\big(\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) - 1\big)\big], \tag{4.8}$$

where λ is a scalar Lagrange multiplier. By fixing $\bar{\boldsymbol{w}}(i)$, minimizing (4.8) with respect to $\boldsymbol{T}_r(i)$ and solving for λ, we have

$$\boldsymbol{T}_r(i) = \frac{\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)}, \tag{4.9}$$

where $\boldsymbol{R} = \mathbb{E}[\boldsymbol{x}(i)\boldsymbol{x}^H(i)]$ and $\bar{\boldsymbol{R}}_{\bar{w}} = \mathbb{E}[\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)]$. By fixing $\boldsymbol{T}_r(i)$, minimizing (4.8) with respect to $\bar{\boldsymbol{w}}(i)$ and solving for λ, we arrive at the expression

$$\bar{\boldsymbol{w}}(i) = \frac{\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta_0)}{\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta_0)}, \tag{4.10}$$

where $\bar{\boldsymbol{R}} = \mathbb{E}[\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{T}_r(i)] = \mathbb{E}[\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)]$. The associated MV is

$$\text{MV} = \frac{1}{\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta_0)}. \tag{4.11}$$

Note that the filter expressions in (4.9) and (4.10) are not closed-form solution for $\bar{\boldsymbol{w}}(i)$ and $\boldsymbol{T}_r(i)$ since (4.9) is a function of $\bar{\boldsymbol{w}}(i)$ and (4.10) depends on $\boldsymbol{T}_r(i)$. Thus, it is necessary to iterate (4.9) and (4.10) with initial values to obtain a solution. An analysis of the optimization problem in (4.7) is given in Appendix E. Unlike existing approaches based on MSWF [27] and AVF [30], the proposed scheme provides an iterative exchange of information between the reduced-rank filter and the transformation matrix and leads to a much simpler adaptive implementation. The transformation matrix reduces the dimension of the input data, whereas the reduced-rank filter yields a unity response in the direction of the SOI. The key strategy lies in the joint optimization of the filters. The rank r must be set by the designer to ensure appropriate performance or can be estimated via another algorithm. In the next section, we seek iterative solutions via adaptive algorithms for the design of $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}(i)$, and automatic rank adaptation algorithms.

4.4 Adaptive Algorithms

In this section, we present adaptive SG and RLS versions of the proposed scheme. We also consider the important issue of automatically determining the rank of the scheme via the proposal of an adaptation technique. We then provide the computational complexity in arithmetic operations of the proposed reduced-rank algorithms.

4.4.1 Stochastic Gradient Algorithm

We present a low-complexity SG adaptive reduced-rank algorithm, which was reported in [81] and is reproduced here for convenience. By computing the instantaneous gradient terms of (4.8) with respect to $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}(i)$, we get

$$\begin{aligned} \nabla L_{\text{un},\boldsymbol{T}_r(i)} &= y^*(i)\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i) + 2\lambda_{T_r}\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i), \\ \nabla L_{\text{un},\bar{\boldsymbol{w}}(i)} &= y^*(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i) + 2\lambda_{\bar{w}}\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0). \end{aligned} \tag{4.12}$$

By introducing the positive step sizes μ_{T_r} and $\mu_{\bar{w}}$, using the gradient rules $\boldsymbol{T}_r(i+1) = \boldsymbol{T}_r(i) - \mu_{T_r}\nabla L_{\text{un},\boldsymbol{T}_r(i)}$ and $\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{w}}(i) - \mu_{\bar{w}}\nabla L_{\text{un},\bar{\boldsymbol{w}}(i)}$, enforcing the constraint and solving the resulting equations, we obtain

$$\boldsymbol{T}_r(i+1) = \boldsymbol{T}_r(i) - \mu_{T_r}y^*(i)[\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i) - \big(\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0)\big)^{-1}\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\boldsymbol{a}^H(\theta_0)\boldsymbol{x}(i)], \tag{4.13}$$

$$\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{w}}(i) - \mu_{\bar{w}}y^*(i)[\boldsymbol{I} - \big(\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0)\big)^{-1}\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)]\bar{\boldsymbol{x}}(i), \tag{4.14}$$

where $y(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)$. The proposed scheme trades-off a full-rank filter against one transformation matrix $\boldsymbol{T}_r(i)$ and one reduced-rank adaptive filter $\bar{\boldsymbol{w}}(i)$ operating simultaneously and exchanging information. We call the SG-based algorithm JIO-SG.

4.4.2 Recursive Least Squares Algorithms

Here we derive an RLS adaptive reduced-rank algorithm for the proposed method. To this end, let us first consider the Lagrangian

$$L_{\text{un}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)\big) = \sum_{l=1}^{i}\alpha^{i-l}|\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(l)|^2 + 2\,\Re[\lambda\big(\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) - 1\big)], \tag{4.15}$$

where α is the forgetting factor chosen as a positive constant close to, but less than 1.

Fixing $\bar{\boldsymbol{w}}(i)$, computing the gradient of (4.15) with respect to $\boldsymbol{T}_r(i)$, equating the gradient to a null vector and solving for λ, we obtain

$$\boldsymbol{T}_r(i) = \frac{\hat{\boldsymbol{R}}^{-1}(i)\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_0)\hat{\boldsymbol{R}}^{-1}(i)\boldsymbol{a}(\theta_0)}, \tag{4.16}$$

where $\hat{\boldsymbol{R}}(i) = \sum_{l=1}^{i}\alpha^{i-l}\boldsymbol{x}(l)\boldsymbol{x}^H(l)$ is the estimate of the input covariance matrix, and

$\bar{\boldsymbol{R}}_{\bar{w}}(i) = \bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)$ is the reduced-rank weight matrix at time instant i. The computation of (4.16) includes the inversion of $\hat{\boldsymbol{R}}(i)$ and $\bar{\boldsymbol{R}}_{\bar{w}}(i)$, which may increase significantly the complexity and create numerical problems. However, the expression in (4.16) can be further simplified using the constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = 1$. The details of the derivation of the proposed RLS algorithm and the simplification are given in Appendix F. The simplified expression for $\boldsymbol{T}_r(i)$ is given by

$$\boldsymbol{T}_r(i) = \frac{\hat{\boldsymbol{\Phi}}(i)\boldsymbol{a}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)}{\boldsymbol{a}^H(\theta_0)\hat{\boldsymbol{\Phi}}(i)\boldsymbol{a}(\theta_0)}, \tag{4.17}$$

where $\hat{\boldsymbol{\Phi}}(i) = \hat{\boldsymbol{R}}^{-1}(i)$. Employing the matrix inversion lemma [18], we obtain

$$\boldsymbol{k}(i) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}{1+\alpha^{-1}\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}, \tag{4.18}$$

$$\hat{\boldsymbol{\Phi}}(i) = \alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1) - \alpha^{-1}\boldsymbol{k}(i)\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1), \tag{4.19}$$

where $\boldsymbol{k}(i) \in \mathbb{C}^{m\times 1}$ is Kalman gain vector. We set $\hat{\boldsymbol{\Phi}}(0) = \delta\boldsymbol{I}$ to start the recursion of (4.19), where δ is a positive constant.

Assuming $\boldsymbol{T}_r(i)$ is known and taking the gradient of (4.15) with respect to $\bar{\boldsymbol{w}}(i)$, equating the terms to a zero vector and solving for λ, we obtain the reduced-rank filter

$$\bar{\boldsymbol{w}}(i) = \frac{\hat{\bar{\boldsymbol{\Phi}}}(i)\boldsymbol{a}(\bar{\theta}_0)}{\bar{\boldsymbol{a}}^H(\theta)\hat{\bar{\boldsymbol{\Phi}}}(i)\bar{\boldsymbol{a}}(\theta_0)}, \tag{4.20}$$

where $\hat{\bar{\boldsymbol{\Phi}}}(i) = \hat{\bar{\boldsymbol{R}}}^{-1}(i)$ and $\hat{\bar{\boldsymbol{R}}}(i) = \sum_{l=1}^{i}\alpha^{i-l}\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)$ is the estimate of the reduced-rank covariance matrix. Using the matrix inversion lemma, we have

$$\bar{\boldsymbol{k}}(i) = \frac{\alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}{1+\alpha^{-1}\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}, \tag{4.21}$$

$$\hat{\bar{\boldsymbol{\Phi}}}(i) = \alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}(i)\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1), \tag{4.22}$$

where $\bar{\boldsymbol{k}}(i) \in \mathbb{C}^{r\times 1}$ is the reduced-rank gain vector and $\hat{\bar{\boldsymbol{\Phi}}}(i) = \hat{\bar{\boldsymbol{R}}}^{-1}(i)$. The recursion of (4.22) is initialized by choosing $\hat{\bar{\boldsymbol{\Phi}}}(0) = \bar{\delta}\boldsymbol{I}$, where $\bar{\delta}$ is a positive constant.

The proposed RLS algorithm trades-off a full-rank filter with m coefficients against one transformation matrix $\boldsymbol{T}_r(i)$, given in (4.17)-(4.19) and one reduced-rank adaptive filter $\bar{\boldsymbol{w}}(i)$, given in (4.20)-(4.22), operating simultaneously and exchanging information. We call this proposed algorithm JIO-RLS.

4.4.3 Complexity of Proposed Algorithms

Here, we evaluate the computational complexity of the proposed and analyzed CMV algorithms. The complexity of the proposed JIO-SG algorithm is $4rm + m + 2r - 3$ for additions and $4rm + m - 7r + 3$ for multiplications, which grow linearly with rm and are about r times higher than the full-rank SG algorithm and significantly lower than the MSWF-SG [27]. If $r \ll m$ then the additional complexity can be acceptable provided the gains in performance justify them. In the case of the proposed JIO-RLS algorithm the complexity is quadratic with m^2 and r^2. This corresponds to a complexity slightly higher than the one observed for the full-rank RLS algorithm, provided r is significantly smaller than m, and comparable to the cost of the MSWF-RLS [27] and the AVF [30].

4.4.4 Automatic Rank Selection

The performance of the algorithms described in the previous subsections depends on the rank r. This motivates the development of methods to automatically adjust r on the basis of the cost function. Unlike prior methods for rank selection which utilize the MSWF-based algorithms [27] or the AVF-based recursions [82], we focus on an approach that jointly determines r based on the LS criterion computed by the filters $\boldsymbol{T}_r^{(r)}(i)$ and $\bar{\boldsymbol{w}}^{(r)}(i)$, where the subscript (r) denotes the rank used for the adaptation. In particular, we present a method for automatically selecting the ranks of the algorithms based on the exponentially weighted *a posteriori* least-squares type cost function described by

$$J_{\text{pmv}}\big(\boldsymbol{T}_r^{(r)}(i-1), \bar{\boldsymbol{w}}^{(r)}(i-1)\big) = \sum_{l=1}^{i} \varrho^{i-l} |\bar{\boldsymbol{w}}^{(r)\ H}(i-1)\boldsymbol{T}_r^{(r)\ H}(i-1)\boldsymbol{x}(l)|^2, \quad (4.23)$$

where ϱ is the exponential weight factor that is required as the optimal rank r can change as a function of the time instant i. The key quantities to be updated are the transformation matrix $\boldsymbol{T}_r^{(r)}(i)$, the reduced-rank filter $\bar{\boldsymbol{w}}^{(r)}(i)$, the associated reduced-rank steering vector $\bar{\boldsymbol{a}}(\theta_0)$ and the inverse of the estimate of the reduced-rank covariance matrix $\hat{\bar{\boldsymbol{\Phi}}}(i)$ (for JIO-RLS). To this end, we define the following extended transformation matrix $\boldsymbol{T}_r^{(r)}(i)$ and

the extended reduced-rank filter weight vector $\bar{\boldsymbol{w}}^{(r)}(i)$ as follows:

$$\begin{aligned}\boldsymbol{T}_r^{(r)}(i) &= \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,r_{\min}} & \dots & t_{1,r_{\max}} \\ t_{2,1} & t_{2,2} & \dots & t_{2,r_{\min}} & \dots & t_{2,r_{\max}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{m,1} & t_{m,2} & \dots & t_{m,r_{\min}} & \dots & t_{m,r_{\max}} \end{bmatrix}, \\ \bar{\boldsymbol{w}}^{(r)}(i) &= \begin{bmatrix} \bar{w}_1 & \bar{w}_2 & \dots & \bar{w}_{r_{\min}} & \dots & \bar{w}_{r_{\max}} \end{bmatrix}^T.\end{aligned} \tag{4.24}$$

The extended transformation matrix $\boldsymbol{T}_r^{(r)}(i)$ and the extended reduced-rank filter weight vector $\bar{\boldsymbol{w}}^{(r)}(i)$ are updated along with the associated quantities $\bar{\boldsymbol{a}}(\theta_0)$ and $\hat{\bar{\boldsymbol{\Phi}}}(i)$ for the maximum allowed rank $r_{\max}$ and then the proposed rank adaptation algorithm determines the rank that is best for each time instant i using the cost function in (4.23). The proposed rank adaptation algorithm is then given by

$$r_{\text{opt}} = \arg \min_{r_{\min} \le j \le r_{\max}} J_{\text{pmv}}\big(\boldsymbol{T}_r^{(j)}(i-1), \bar{\boldsymbol{w}}^{(j)}(i-1)\big), \tag{4.25}$$

where j is an integer between $r_{\min}$ and $r_{\max}$. Note that a smaller rank may provide faster adaptation during the initial stages of the estimation procedure and a greater rank usually yields a better steady-state performance. Our studies reveal that the range for which the rank r of the proposed algorithms have a positive impact on the performance of the algorithms is limited, being from $r_{\min} = 3$ to $r_{\max} = 8$ for the reduced-rank filter recursions. These values are rather insensitive to the system load (number of users), to the number of array elements and work very well for all scenarios and algorithms examined. The additional complexity of the proposed rank adaptation algorithm is that it requires the update of all involved quantities with the maximum allowed rank $r_{\max}$ and the computation of the cost function in (4.23). This procedure can significantly improve the convergence performance and can be relaxed (the rank can be made fixed) once the algorithm reaches steady state. Choosing an inadequate rank for adaptation may lead to performance degradation, which gradually increases as the adaptation rank deviates from the optimal rank. A mechanism for automatically adjusting $r_{\min}$ and $r_{\max}$ based on a figure of merit and the processed data would be an important technique to be investigated. For example, this mechanism could in principle adjust $r_{\min}$ and $r_{\max}$ in order to address the needs of the model and the performance requirements. This remains a topic for future investigation.

One can also argue that the proposed rank adaptation may not be universally applied to signal processing problems, even though it has been proven highly effective to the problems we dealt with. Another possibility for rank adaptation is the use of the cross-validation (CV) method reported in [82]. This approach selects the lengths of the filters

that minimize a cost function that is estimated on the basis of data that have not been used in the process of building the filters themselves. This approach based on the concept of "leave one out" can be used to determine the rank without requiring any prior knowledge or the setting of a range of values [82]. A drawback of this method is that it may significantly increase the length of the filters, resulting in higher complexity. Other possible approaches for rank selection may rely on some prior knowledge about the environment and the system for inferring the required rank for operation. The development of cost-effective methods for rank selection remains an interesting area for investigation.

4.5 Analysis of Algorithms

In this section, we present the stability of the MSE convergence analyses of the proposed SG algorithms. Specifically, we consider the joint optimization approach and derive conditions of stability for the proposed SG algorithm. We then assume that the algorithms will convergence and carry out the MSE convergence analysis in order to semi-analytically determine the MSE upon convergence. The RLS algorithms are expected to convergence to the optimal CMV filter and this has been verified in our studies. A discussion on the preservation of the MV performance, the existence of multiple solutions and an analysis of the optimization of the proposed scheme valid for both SG and RLS algorithms is included in Appendix D.

4.5.1 Stability Analysis

In order to establish conditions for the stability of the JIO-SG algorithm, we define the error matrices at time instant i as

$$\boldsymbol{e}_{T_r}(i) = \boldsymbol{T}_r(i) - \boldsymbol{T}_{r,\text{opt}}, \tag{4.26}$$

$$\boldsymbol{e}_{\bar{w}}(i) = \bar{\boldsymbol{w}}(i) - \bar{\boldsymbol{w}}_{\text{opt}}, \tag{4.27}$$

where $\boldsymbol{T}_{r,\text{opt}}$ and $\bar{\boldsymbol{w}}_{\text{opt}}$ are the optimal parameter estimators. Since we are dealing with a joint optimization procedure, both filters have to be considered jointly. By substituting the expressions of $\boldsymbol{e}_{T_r}(i)$ and $\boldsymbol{e}_{\bar{w}}(i)$ in (4.13) and (4.14), respectively, and rearranging the

terms, we obtain

$$\begin{aligned}\boldsymbol{e}_{T_r}(i+1) = &\left\{\boldsymbol{I} - \mu_{T_r}[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)]\boldsymbol{x}(i)\boldsymbol{x}^H(i)\right\}\boldsymbol{e}_{T_r}(i) \\ &- \mu_{T_r}[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)]\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{x}^H(i)\boldsymbol{T}_r(i)\boldsymbol{e}_{\bar{w}}(i) \\ &+ \mu_{T_r}[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)]\boldsymbol{x}(i)\boldsymbol{x}^H(i)[\boldsymbol{T}_r(i)(\mathbf{I} - \bar{\boldsymbol{w}}_{\text{opt}}\bar{\boldsymbol{w}}^H(i)) - \boldsymbol{T}_{r,\text{opt}}],\end{aligned} \tag{4.28}$$

$$\begin{aligned}\boldsymbol{e}_{\bar{w}}(i+1) = &\left\{\boldsymbol{I} - \mu_{\bar{w}}[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0))^{-1}\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)]\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)\right\}\boldsymbol{e}_{\bar{w}}(i) \\ &- \mu_{\bar{w}}[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0))^{-1}\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)]\bar{\boldsymbol{x}}(i)\boldsymbol{x}^H(i)\boldsymbol{e}_{T_r}(i) \\ &+ \mu_{\bar{w}}[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0))^{-1}\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0))\boldsymbol{T}_r^H(i)]\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)(\boldsymbol{T}_r(i)(\boldsymbol{I} - \bar{\boldsymbol{w}}_{\text{opt}}) - \boldsymbol{T}_{r,\text{opt}}).\end{aligned} \tag{4.29}$$

Taking expectations and simplifying the terms, we have

$$\begin{bmatrix} \mathbb{E}[\boldsymbol{e}_{T_r}(i+1)] \\ \mathbb{E}[\boldsymbol{e}_{\bar{w}}(i+1)] \end{bmatrix} = \boldsymbol{P}\begin{bmatrix} \mathbb{E}[\boldsymbol{e}_{T_r}(i)] \\ \mathbb{E}[\boldsymbol{e}_{\bar{w}}(i)] \end{bmatrix} + \boldsymbol{T}, \tag{4.30}$$

where

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{I} - \mu_{T_r}[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)]\boldsymbol{x}(i)\boldsymbol{x}^H(i) & -\mu_{T_r}[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)]\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{x}^H(i)\boldsymbol{T}_r(i) \\ -\mu_{\bar{w}}[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0))^{-1}\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)]\bar{\boldsymbol{x}}(i)\boldsymbol{x}^H(i) & \boldsymbol{I} - \mu_{\bar{w}}[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0))^{-1}\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)]\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i) \end{bmatrix}$$

$$\boldsymbol{T} = \begin{bmatrix} \mu_{T_r}[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)]\boldsymbol{x}(i)\boldsymbol{x}^H(i)[\boldsymbol{T}_r(i)(\mathbf{I} - \bar{\boldsymbol{w}}_{\text{opt}}\bar{\boldsymbol{w}}^H(i)) - \boldsymbol{T}_{r,\text{opt}}] \\ \mu_{\bar{w}}[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0))^{-1}\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0))\boldsymbol{T}_r^H(i)]\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)(\boldsymbol{T}_r(i)(\boldsymbol{I} - \bar{\boldsymbol{w}}_{\text{opt}}) - \boldsymbol{T}_{r,\text{opt}}) \end{bmatrix}.$$

The previous equations imply that the stability of the algorithms depends on the spectral radius of $\boldsymbol{P}$. For convergence, the step sizes should be chosen such that the eigenvalues of $\boldsymbol{P}^H\boldsymbol{P}$ are less than one. Unlike the stability analysis of most adaptive algorithms [18], in the proposed approach the terms are more involved and depend on each other as evidenced by the equations in $\boldsymbol{P}$ and $\boldsymbol{T}$.

4.5.2 MSE Convergence Analysis

Let us consider in this part an analysis of the MSE in steady state. This follows the general steps of the MSE convergence analysis of [18] even though novel elements will be introduced in the proposed framework. These novel elements in the analysis are the joint optimization of the two adaptive filters $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}_r(i)$ of the proposed scheme and a strategy to incorporate the effect of the step size of the recursions in (4.13) and (4.14).

Let us define the MSE at time $i+1$ using the relations

$$\boldsymbol{e}_w(i+1) = \boldsymbol{w}(i+1) - \boldsymbol{w}_{\text{opt}}, \tag{4.31}$$

and

$$\xi(i) = \mathbb{E}[\boldsymbol{w}^H(i)\boldsymbol{x}(i)\boldsymbol{x}(i)\boldsymbol{w}(i)], \tag{4.32}$$

where $\boldsymbol{w}_{\text{opt}}$ is the full-rank optimal weight solution and $\boldsymbol{w}(i) = \boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i)$ with m coefficients is the r rank approximation of a full-rank filter obtained with an inverse mapping performed by $\boldsymbol{T}_r(i)$.

The MSE of the proposed scheme can be expressed by

$$\begin{aligned} \text{MSE}(i) &= \mathbb{E}[|d(i) - \boldsymbol{w}^H(i)\boldsymbol{x}(i)|^2] \\ &= \epsilon_{\text{min}} + \xi(i) - \xi_{\text{min}} - \mathbb{E}[\boldsymbol{e}_w^H(i)]\boldsymbol{a}(\theta_0) - \boldsymbol{a}^H(\theta_0)\mathbb{E}[\boldsymbol{e}_w(i)] \\ &= \epsilon_{\text{min}} + \xi_{\text{ex}}(i) - \mathbb{E}[\boldsymbol{e}_w^H(i)]\boldsymbol{a}(\theta_0) - \boldsymbol{a}^H(\theta_0)\mathbb{E}[\boldsymbol{e}_w(i)], \end{aligned} \tag{4.33}$$

where $d(i)$ corresponds to the desired signal and $\epsilon_{\text{min}} = \mathbb{E}[|d(i) - \boldsymbol{w}_{\text{opt}}^H\boldsymbol{x}(i)|^2]$ is the MSE with

$$\boldsymbol{w}_{\text{opt}} = \xi_{\text{min}}\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0), \tag{4.34}$$

where $\xi_{\text{min}} = 1/(\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0))$ is the minimum variance, and $\xi_{\text{ex}}(i) = \xi(i) - \xi_{\text{min}}$ is the excess MSE due to the adaptation process at the time instant i. Since $\lim_{i\to\infty}\mathbb{E}[\boldsymbol{e}_w(i)] = \boldsymbol{0}$, we have

$$\lim_{i\to\infty}\text{MSE}(i) = \epsilon_{\text{min}} + \lim_{i\to\infty}\xi_{\text{ex}}(i), \tag{4.35}$$

where $\xi_{\text{ex}}(\infty)$ term in (4.35) is the steady-state excess MSE resulting from the adaptation process. The main difference here from prior work lies in the fact that this refers to the excess MSE produced by a r rank approximation filter $\boldsymbol{w}(i)$. In order to analyze the trajectory of $\xi(i)$, let us rewrite it as

$$\begin{aligned} \xi(i) &= \mathbb{E}[\boldsymbol{w}^H(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{w}(i)] \\ &= \mathbb{E}[\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i)] \\ &= \text{trace}\,\mathbb{E}[\boldsymbol{R}_w(i)\boldsymbol{R}], \end{aligned} \tag{4.36}$$

where $\boldsymbol{R}_w(i) = \mathbb{E}[\boldsymbol{w}(i)\boldsymbol{w}^H(i)] = \boldsymbol{w}_{\text{opt}}\boldsymbol{w}_{\text{opt}}^H + \mathbb{E}[\boldsymbol{e}_w(i)]\boldsymbol{w}_{\text{opt}}^H + \boldsymbol{w}_{\text{opt}}\mathbb{E}[\boldsymbol{e}_w^H(i)] + \boldsymbol{R}_{e_w}(i)$ [27].

To proceed with the analysis, we must define the quantities $\boldsymbol{R} = \boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Psi}^H$, where the columns of $\boldsymbol{\Psi}$ are the eigenvectors of the symmetric and positive semi-definite matrix $\boldsymbol{R}$ and $\boldsymbol{\Lambda}$ is the diagonal matrix of the corresponding eigenvalues, $\boldsymbol{R}_{e_w}(i) = \mathbb{E}[\boldsymbol{e}_w(i)\boldsymbol{e}_w^H(i)]$,

the rotated tap error vector $\tilde{\boldsymbol{e}}_w(i) = \boldsymbol{\Psi}^H \boldsymbol{e}_w(i)$, the rotated signal vectors $\bar{\boldsymbol{x}}(i) = \boldsymbol{\Psi}^H \boldsymbol{x}(i)$, $\tilde{\boldsymbol{a}}(\theta_0) = \boldsymbol{\Psi}^H \boldsymbol{a}(\theta_0)$ and $\boldsymbol{R}_{\tilde{e}_w}(i) = \mathbb{E}[\tilde{e}_w(i)\tilde{\boldsymbol{e}}_w^H(i)] = \boldsymbol{\Psi}^H \boldsymbol{R}_{e_w}(i)\boldsymbol{\Psi}$. Rewriting (4.36) in terms of the above transformed quantities we have

$$\begin{aligned} \xi(i) &= \text{trace}\, \mathbb{E}[\boldsymbol{\Lambda}\boldsymbol{\Psi}^H \boldsymbol{R}_w \boldsymbol{\Psi}] \\ &= \xi_{\text{min}} + \text{trace}\, \big\{\mathbb{E}\big[\tilde{\boldsymbol{e}}_w(i)\tilde{\boldsymbol{a}}^H(\theta_0) + \tilde{\boldsymbol{a}}(\theta_0)\mathbb{E}[\tilde{\boldsymbol{e}}_w^H(i)] + \boldsymbol{\Lambda}\boldsymbol{R}_{\tilde{e}_w}(i)\big]\big\}. \end{aligned} \tag{4.37}$$

Since $\lim_{i\to\infty} \mathbb{E}[\tilde{\boldsymbol{e}}_w(i)] = \boldsymbol{0}$, then $\lim_{i\to\infty} \xi(i) = \xi_{\text{min}} + \text{trace}[\boldsymbol{\Lambda}\boldsymbol{R}_{\tilde{e}_w}]$. Thus, it is evident that to assess the evolution of $\xi(i)$ it is sufficient to study $\boldsymbol{R}_{\tilde{e}_w}(i)$.

Using $\boldsymbol{e}_{T_r}(i)$ and $\boldsymbol{e}_{\bar{w}}(i)$ and combining them to compute $\boldsymbol{e}_w(i)$, we get

$$\begin{aligned} \boldsymbol{e}_w(i) &= \boldsymbol{w}(i) - \boldsymbol{w}_{\text{opt}} = \boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i) - \boldsymbol{T}_{r,\ \text{opt}}\bar{\boldsymbol{w}}_{\text{opt}} \\ &= \boldsymbol{e}_{T_r}(i)\boldsymbol{e}_{\bar{w}}(i) + \boldsymbol{T}_{r,\text{opt}}\boldsymbol{e}_{\bar{w}}(i) + \boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}. \end{aligned} \tag{4.38}$$

Substituting the expressions for $\boldsymbol{e}_{T_r}(i+1)$ and $\boldsymbol{e}_{\bar{w}}(i+1)$ in (4.28) and (4.29), respectively, to compute $\boldsymbol{e}_w(i+1)$, we have

$$\begin{aligned} \boldsymbol{e}_w(i+1) =& \boldsymbol{e}_w(i) - \mu_{\bar{w}} y^*(i)\boldsymbol{T}_r(i)\bar{\boldsymbol{r}}_p(i) - \mu_{T_r} y^*(i)\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{w}}(i) \\ &+ \mu_{T_r}\mu_{\bar{w}}\big(y^*(i)\big)^2 \boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{r}}_p(i) + \boldsymbol{T}_{r,\text{opt}}\boldsymbol{e}_{\bar{w}}(i) + \boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}, \end{aligned} \tag{4.39}$$

where

$$\begin{aligned} \boldsymbol{T}_{r_p}(i) &= \big(\boldsymbol{I} - (\boldsymbol{a}^H(\theta_0)\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big)\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i) \\ \bar{\boldsymbol{r}}_p(i) &= (\boldsymbol{I} - (\boldsymbol{T}_r(i)\boldsymbol{a}^H(\theta_0)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{T}_r(i)\boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\boldsymbol{T}_r^H(i)\boldsymbol{T}_r(i)\boldsymbol{x}(i). \end{aligned}$$

We can further rewrite the expression above in order to obtain a more compact and convenient representation as

$$\boldsymbol{e}_w(i+1) = (\boldsymbol{I} - \boldsymbol{Q})\boldsymbol{e}_w(i) + \boldsymbol{B}\boldsymbol{C} + \mu_{T_r}\mu_{\bar{w}}\big(y^*(i)\big)^2 \boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{r}}_p(i) + \boldsymbol{e}_{T_r}\bar{\boldsymbol{w}}_{\text{opt}}, \tag{4.40}$$

where

$$\begin{aligned} \boldsymbol{Q} &= \mu_{\bar{w}}\boldsymbol{T}_r(i)\bar{\boldsymbol{r}}_p(i)\boldsymbol{x}^H(i) + \mu_{T_r}\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{x}^H(i) - \boldsymbol{T}_{r,\text{opt}} \\ \boldsymbol{B} &= -\mu_{\bar{w}}\boldsymbol{T}_r(i)\bar{\boldsymbol{r}}_p(i)\boldsymbol{x}^H(i) - \mu_{T_r}\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{x}^H(i) \\ \boldsymbol{C} &= \boldsymbol{e}_{\boldsymbol{T}_r}(i)\bar{\boldsymbol{w}}_{\text{opt}} + \boldsymbol{T}_{r,\text{opt}}\boldsymbol{e}_{\bar{w}}(i) + \boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}. \end{aligned}$$

Now, we need to compute $\boldsymbol{R}_{e_w}(i+1) = \mathbb{E}[\boldsymbol{e}_w(i+1)\boldsymbol{e}_w^H(i+1)]$ by using the result in (4.40), which yields,

$$\begin{aligned}\boldsymbol{R}_{e_w}(i+1) =&(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{R}_{e_w}(i)(\boldsymbol{I}-\boldsymbol{Q})^H + (\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{e}_w(i)\boldsymbol{C}^H\boldsymbol{B}^H \\ &+ \mu_{T_r}\mu_{\bar{w}}(y^*)^2(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{e}_w(i)(\bar{\boldsymbol{r}}_p^H(i)\boldsymbol{T}_{r_p}^H(i)) \\ &+ (\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{e}_w(i)\bar{\boldsymbol{w}}_{\text{opt}}^H\boldsymbol{T}_{r,\text{opt}}^H + \boldsymbol{B}\boldsymbol{C}\boldsymbol{e}_w^H(i)(\boldsymbol{I}-\boldsymbol{A})^H \\ &+ \boldsymbol{B}\boldsymbol{C}\boldsymbol{C}^H\boldsymbol{B}^H + \mu_{T_r}\mu_{\bar{w}}(y^*(i))^2\boldsymbol{B}\boldsymbol{C}\bar{\boldsymbol{r}}_p^H(i)\boldsymbol{T}_{r_p}^H(i) \\ &+ \boldsymbol{B}\boldsymbol{C}\boldsymbol{w}_{\text{opt}}^H\boldsymbol{e}_{T_r}^H(i) + \mu_{T_r}\mu_{\bar{w}}(y^*(i))^2\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{r}}_p(i)\boldsymbol{e}_w^H(i)(\boldsymbol{I}-\boldsymbol{Q})^H \\ &+ \mu_{T_r}\mu_{\bar{w}}(y^*(i))^2\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{r}}_p(i)\boldsymbol{C}^H\boldsymbol{Q}^H \\ &+ (\mu_{T_r}\mu_{\bar{w}})^2|y^*|^4\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{r}}_p(i)\bar{\boldsymbol{r}}_p^H(i)\boldsymbol{T}_{r_p}(i) \\ &+ \mu_{T_r}\mu_{\bar{w}}(y^*(i))^2\boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}\bar{\boldsymbol{r}}_p(i)\boldsymbol{T}_{r_p}(i) \\ &- \boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}\boldsymbol{e}_w^H(i)(\boldsymbol{I}-\boldsymbol{Q})^H + \boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}\boldsymbol{C}^H\boldsymbol{B}^H \\ &+ \boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}\bar{\boldsymbol{w}}_{\text{opt}}^H\boldsymbol{e}_{T_r}^H(i).\end{aligned} \tag{4.41}$$

Since $\mathbb{E}[\boldsymbol{e}_w(i)] = \boldsymbol{0}$ and $\mathbb{E}[\boldsymbol{e}_{T_r}(i)] = \boldsymbol{0}$, we can simplify the previous expression and obtain

$$\begin{aligned}\boldsymbol{R}_{e_w}(i+1) =&(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{R}_{e_w}(i)(\boldsymbol{I}-\boldsymbol{Q})^H \\ &+ \boldsymbol{B}\boldsymbol{C}\boldsymbol{C}^H\boldsymbol{B}^H + \mu_{T_r}\mu_{\bar{w}}(y^*(i))^2\boldsymbol{B}\boldsymbol{C}\bar{\boldsymbol{r}}_p^H(i)\boldsymbol{T}_{r_p}^H(i) \\ &+ \mu_{T_r}\mu_{\bar{w}}(y^*(i))^2\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{r}}_p(i)\boldsymbol{C}^H\boldsymbol{Q}^H \\ &+ (\mu_{T_r}\mu_{\bar{w}})^2|y^*(i)|^4\boldsymbol{T}_{r_p}(i)\bar{\boldsymbol{r}}_p(i)\bar{\boldsymbol{r}}_p^H(i)\boldsymbol{T}r_p(i) \\ &+ \boldsymbol{e}_{T_r}(i)\bar{\boldsymbol{w}}_{\text{opt}}\bar{\boldsymbol{w}}_{\text{opt}}^H\boldsymbol{e}_{T_r}^H(i).\end{aligned} \tag{4.42}$$

Solving for $\boldsymbol{R}_{e_w}$, the MSE can be computed by

$$\text{MSE}(i+1) = \epsilon_{\text{min}} + \text{trace}[\boldsymbol{\Lambda}\boldsymbol{R}_{\tilde{e}_w}(i)] = \epsilon_{\text{min}} + \text{trace}[\boldsymbol{\Lambda}\boldsymbol{\Psi}\boldsymbol{R}_{e_w}(i)\boldsymbol{\Psi}^H]. \tag{4.43}$$

It should be remarked that the expression for $\boldsymbol{R}_{e_w}(i)$ is quite involved and requires a semi-analytical approach with the aid of computer simulations for its computation. This is because the terms resulting from the joint adaptation create numerous extra terms in the expression of $\boldsymbol{R}_{e_w}(i)$, which are very difficult to isolate. We found that using computer simulations to pre-compute the terms of $\boldsymbol{R}_{e_w}(i)$ as a function of the step sizes was more practical and resulted in good match between the semi-analytical and simulated curves. In the following section, we will demonstrate that it is able to predict the performance of the proposed SG algorithm.

4.6 Simulations

In this section we evaluate the performance of the proposed and the analyzed beamforming algorithms via computer simulations. We also verify the validity of the MSE convergence analysis of the previous section. A smart antenna system with a ULA containing m sensor elements is considered for assessing the beamforming algorithms. In particular, the performance of the proposed JIO-SG and JIO-RLS algorithms is compared with existing techniques, namely, the full-rank CMV-SG [14] and CMV-RLS [17], and the reduced-rank algorithms with $\boldsymbol{T}_r(i)$ designed according to the MSWF [27], the AVF [82] and the optimal linear beamformer that assumes the knowledge of the covariance matrix [8]. In particular, the algorithms are compared in terms of the MSE and the SINR. For each scenario, 200 runs are used to obtain the curves. In all simulations, the desired signal power is $\sigma_d^2 = 1$. The filters are initialized as $\bar{\boldsymbol{w}}(0) = [1, 0, \ldots, 0]$ and $\boldsymbol{T}_r(0) = [\boldsymbol{I}_r^T \mathbf{0}_{r\times(m-r)}^T]$, where $\mathbf{0}_{r\times(m-r)}$ is a $r \times (m-r)$ matrix.

4.6.1 MSE Analysis Performance

In this part, we verify that the results in (4.40) and (4.42) on MSE convergence analysis of the proposed reduced-rank SG algorithms can provide a means of estimating the MSE upon convergence. The steady state MSE between the desired and the estimated symbol obtained through simulation is compared with the steady state MSE computed via the expressions derived in Section 4.5. In order to illustrate the usefulness of our analysis we have carried out some experiments. To semi-analytically compute the MSE for the SG recursion, we have used (4.34) and assumed the knowledge of the data covariance matrix $\mathbf{R}$. We consider 5 interferers ($q = 6$ users in total - the SOI and the interferers) at -60^o, -30^o, 0^o, 45^o, 60^o with powers following a log-normal distribution with associated standard deviation 3 dB around the SOI's power level, which impinges on the array at 15^o.

We compare the results obtained via simulations with those obtained by the semi-analytical approach presented in Section 4.5. In particular, we consider two sets of parameters in order to check the validity of our approach. One of the sets has larger step sizes ($\mu_{T_r} = 0.0025$ and $\mu_{\bar{w}} = 0.01$), whereas the other set employs smaller step sizes ($\mu_{T_r} = 0.001$ and $\mu_{\bar{w}} = 0.001$) for the recursions. The results shown in Fig. 4.2 indicate that the curves obtained with the semi-analytical approach agrees with those obtained via simulations for both sets of parameters, verifying the validity of our analysis. The initial values of the curves are not exact 0 dB since the initialization of the reduced-rank weight vector $\bar{\boldsymbol{w}}(0)$ influences the beamformer output. Note that the algorithms with smaller step

sizes converge slower than the algorithms equipped with larger step sizes. However, the proposed algorithms with smaller step sizes converge to the same level of MSE as the optimal CMV, whereas the proposed algorithms with larger step sizes exhibit a higher level of misadjustment. The MSE of the optimal CMV is not as low as -15 dB due to the impact of the interference. In what follows, we will consider the convergence rate of the proposed reduced-rank algorithms in comparison with existing algorithms.

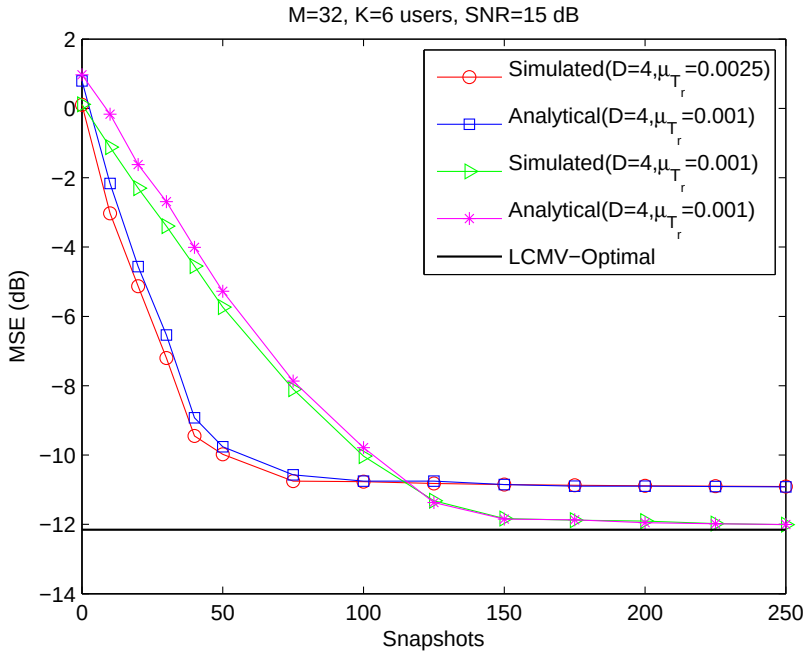

Fig. 4.2: MSE analytical versus simulated performance for the proposed reduced-rank SG algorithm.

4.6.2 SINR Performance

In the first two experiments, we consider 7 interferers at -60^o, -45^o, -30^o, -15^0, 0^o, 45^o, 60^o with powers following a log-normal distribution with associated standard deviation 3 dB around the SOI's power level. The SOI impinges on the array at 30^o. The parameters of the algorithms are optimized.

We first evaluate the SINR performance of the analyzed algorithms against the rank r using optimized parameters (μ_{T_r}, $\mu_{\bar{w}}$ and forgetting factor λ) for all schemes and $N = 250$ snapshots. The results in Fig. 4.3 indicate that the best rank for the proposed scheme is $r = 4$ (which will be used in the second scenario) and the corresponding output SINR value is very close to that of the optimal full-rank CMV filter. Our studies with systems

with different sizes show that r is relatively invariant to the system size, which brings considerable computational savings. In practice, the rank r can be adapted in order to obtain fast convergence and ensure good steady-state performance and tracking after convergence.

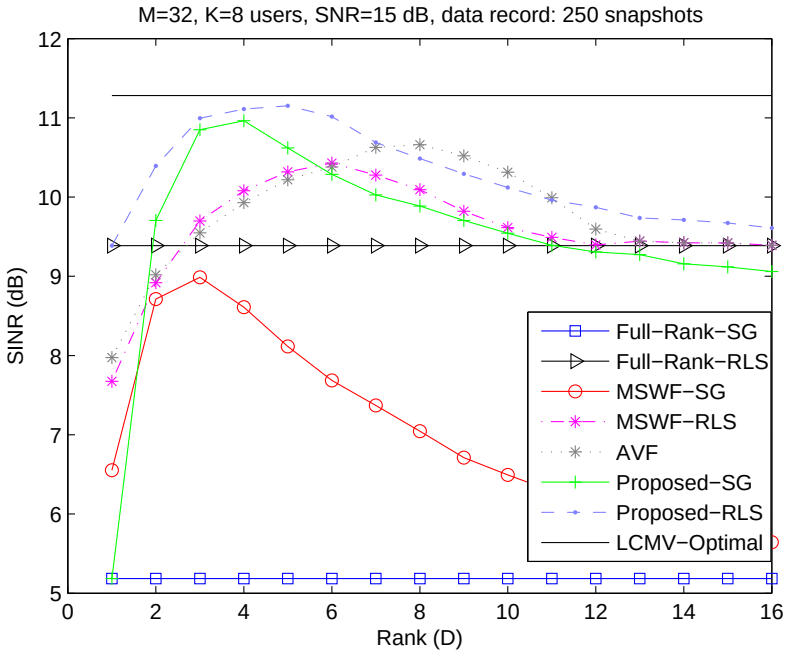

Fig. 4.3: SINR performance of CMV algorithms against rank (r) with $m = 32$, SNR=15dB, $N = 250$ snapshots.

We show another scenario in Fig. 4.4 where the adaptive LCMV filters are set to converge to the same level of SINR. The parameters used to obtain these curves are also shown. The SG version of the MSWF is known to have problems in these situations since it does not tridiagonalize its covariance matrix [27], being unable to approach the optimal LCMV. The curves show an excellent performance for the proposed scheme which converges much faster than the full-rank-SG algorithm, and is also better than the more complex MSWF-RLS and AVF schemes.

In the next experiment, we consider the design of the proposed adaptive reduced-rank CMV algorithms equipped with the automatic rank selection method. We consider 5 interferers at -60^o, -30^o, 0^o, 45^o, 60^o with equal powers to the SOI, which impinges on the array at 15^o. Specifically, we evaluate the proposed rank selection algorithms against the use of fixed ranks, namely, $r = 3$ and $r = 8$ for both SG and RLS algorithms. The results show that the proposed automatic rank selection method is capable of ensuring an excellent trade-off between convergence speed and steady-state performance, as illus-

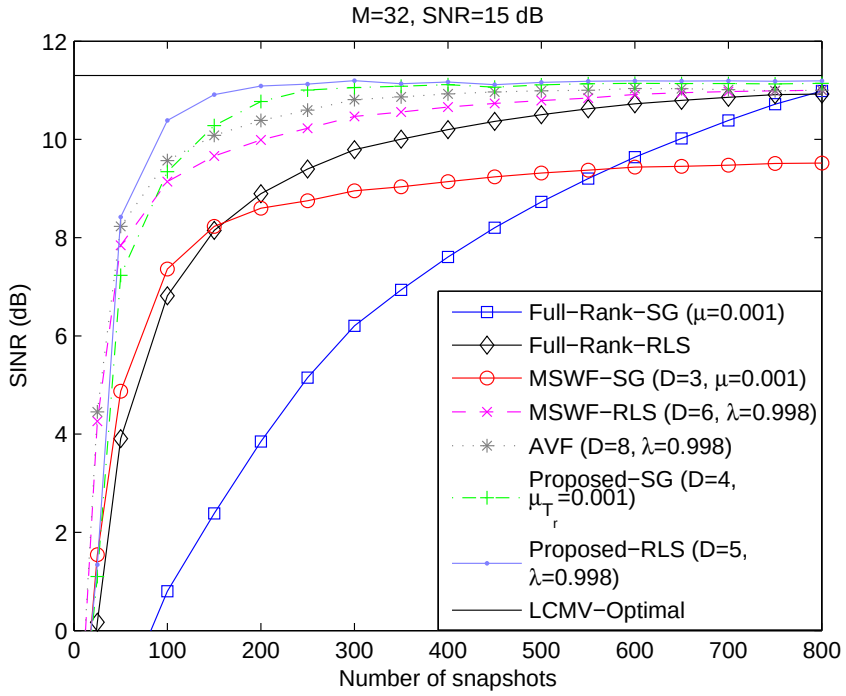

Fig. 4.4: SINR performance of CMV algorithms against snapshots) with $m = 32$, SNR=15dB.

trated in Fig 4.5. In particular, the proposed algorithm can achieve a significantly faster convergence performance than the scheme with fixed rank $r = 8$.

In the last experiment, we consider a non-stationary scenario where the system has 6 users with equal power and the environment experiences a sudden change at time $i = 800$. The 5 interferers impinge on the ULA at -60^o, -30^o, 0^o, 45^o, 60^o with equal powers to the SOI, which impinges on the array at 15^o. At time instant $i = 800$ we have 3 interferers with 5 dB above the SOI's power level entering the system with DOAs -45^o, -15^o and 30^o, whereas one interferer with DOA 45^o and a power level equal to the SOI exits the system. The proposed and existing adaptive beamforming algorithms are equipped with automatic rank adaptation techniques and have to adjust their parameters in order to suppress the interferers. We optimize the step sizes and the forgetting factors of all the algorithms in order to ensure that they converge as fast as they can. The results of this experiment are depicted in Fig. 4.6. The curves show that the proposed reduced-rank algorithms have a superior performance to the existing algorithms.

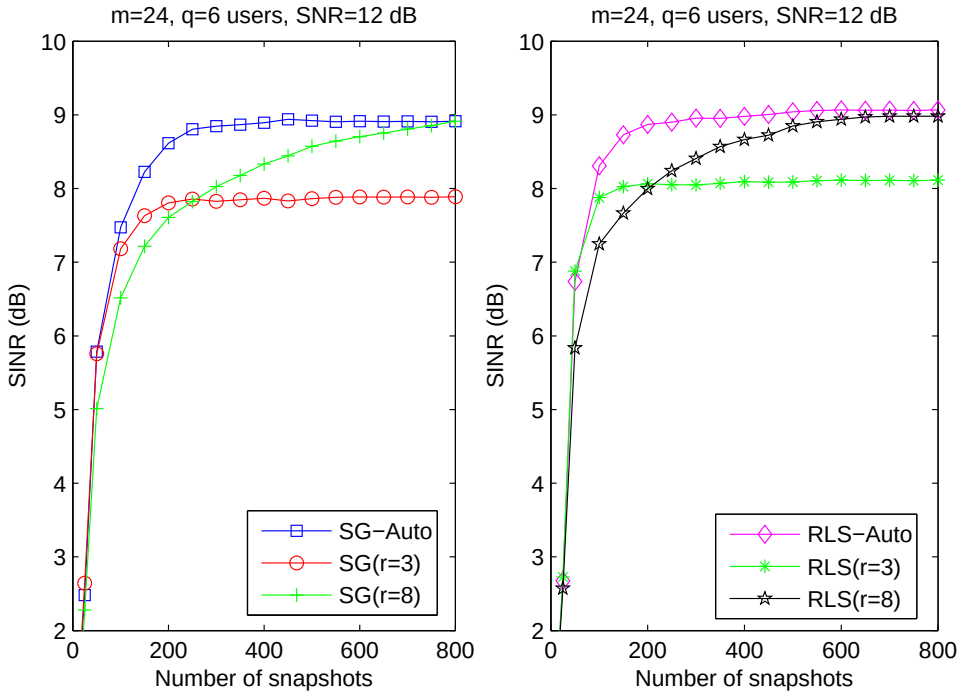

Fig. 4.5: SINR performance of CMV (a) SG and (b) RLS algorithms against snapshots with $m = 24$, SNR$=$ 12dB with automatic rank selection.

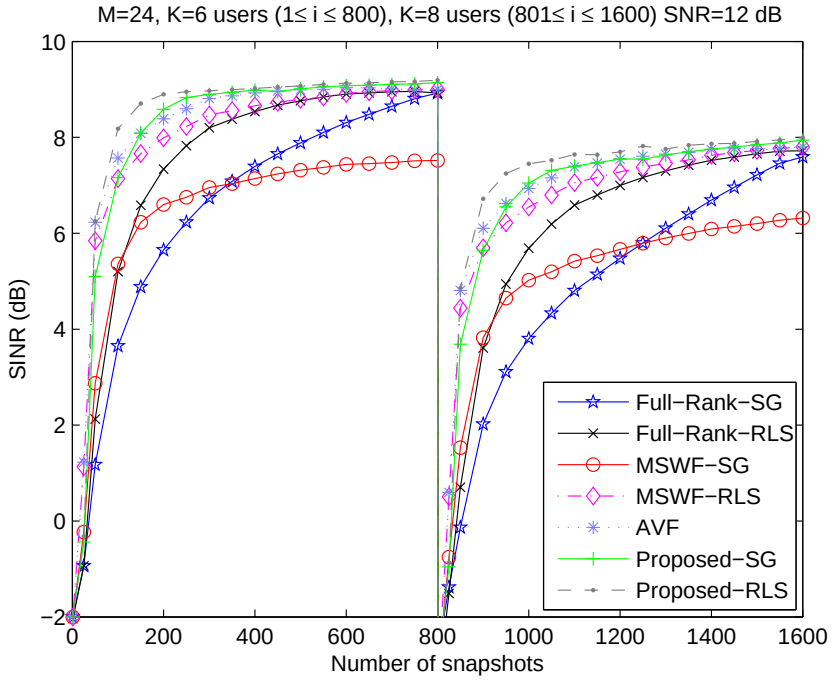

Fig. 4.6: SINR performance of CMV algorithms against snapshots with $m = 24$, SNR$=$ 12dB in a non-stationary scenario.

4.7 Introduction for DOA Estimation

In the following sections, we will introduce a minimum variance DOA estimation algorithm based on the JIO scheme. In many array processing related fields such as radar, sonar, and wireless communications, the information of interest extracted from the received signals is the direction of arrival (DOA) of waves transmitted from radiating sources to the antenna array. The DOA estimation problem has received considerable attention in the last several decades [96]. Many DOA estimation algorithms have been reported in the literature, e.g., [9, 32], and the references therein. The most representative algorithms can be categorized into Capon's method [33], maximum-likelihood (ML) [38], and subspace-based methods [34]- [37].

As reviewed in chapter 1, the Capon's method calculates and plots the output power spectrum over the scanning angles and determines the DOA by locating the peaks in the spectrum. Its resolution strongly relies on the number of available snapshots and the array size. The ML type algorithms are robust for the DOA estimation since they exhibit superior resolution in severe scenarios with low input SNR or the number of snapshots is small. However, the implementation of the ML type methods are always quite difficult and require intensive computational cost, which limit their practical applications.

The subspace-based algorithms, which exploit the structure of the received data to decompose the observation space into a signal subspace and a corresponding orthogonal noise subspace, play an important role for the DOA estimation. According to the approach to compute the signal subspace, the subspace-based methods can be classified into eigen-decomposition, subspace tracking, and basis vectors based algorithms. Among the well-known eigen-decomposition algorithms are MUSIC [34], and ESPRIT [35]. The MUSIC algorithm plots the output power spectrum by scanning the possible angles and selects the peaks to estimate the directions of the sources. ESPRIT achieves improved resolution by employing a displacement invariance in some specific array structures [32]. The subspace tracking techniques [97]- [100] avoid a direct eigen-decomposition and utilize iteration procedures to estimate the signal subspace. They provide an trade-off between the resolution and the complexity. The basis vectors based subspace algorithms include auxiliary vector (AV) [37] and conjugate gradient (CG) [36] that were proposed recently. The AV basis vectors or the CG residual vectors constitute the signal subspace for the DOA estimation without eigen-decomposition.

Our contribution is to employ the JIO reduced-rank scheme introduced in this chapter to develop adaptive algorithms for the DOA estimation. The transformation matrix and the reduced-rank weight vector of the JIO scheme are jointly optimized according to the

minimum variance (MV) criterion for plotting the output power spectrum over the possible scanning angles. We derive a constrained RLS-based algorithm to iteratively estimate the transformation matrix and the reduced-rank weight vector. A spatial smoothing (SS) approach [102, 103] is employed to handle the problem of highly correlated sources. It is efficient to resolve the DOA estimation problem with large arrays and a small number of snapshots, and exhibits a dominance when many sources are present. The Capon's method and subspace-based methods are inferior to the proposed method for direction finding with a large number of users. Although the ML algorithm is robust to these severe conditions, with large arrays it needs extremely high computational cost that is not acceptable in practice. Furthermore, the proposed algorithm works well without knowledge of the number of the sources, which significantly degrades the resolution of the subspace-based and the ML methods.

4.8 Problem Statement

According to the system model introduced in chapter 2, the input covariance matrix can be expressed by

$$\boldsymbol{R} = \mathbb{E}[\boldsymbol{x}(i)\boldsymbol{x}^H(i)] = \boldsymbol{A}(\theta)\boldsymbol{R}_s\boldsymbol{A}^H(\theta) + \sigma_n^2\boldsymbol{I}, \tag{4.44}$$

where $\boldsymbol{R}_s = \mathbb{E}[\boldsymbol{s}(i)\boldsymbol{s}^H(i)]$ denotes the signal covariance matrix, which is diagonal if the sources are uncorrelated and is nondiagonal and nonsingular for partially correlated sources, and $\mathbb{E}[\boldsymbol{n}(i)\boldsymbol{n}^H(i)] = \sigma_n^2\boldsymbol{I}_{m\times m}$. The matrix $\boldsymbol{R}$ must be estimated in practice. In this work, we use a time-average estimate given by

$$\hat{\boldsymbol{R}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}(i)\boldsymbol{x}^H(i). \tag{4.45}$$

It is clear that a small number of snapshots results in a poor estimation of the input covariance matrix, which degrades the DOA estimation of the Capon's method and most subspace-based methods and thus reduces their probability of resolution. With large arrays, the resolution can be compensated to a certain extent whereas the computational cost increases. Besides, eigen-decomposition and subspace tracking based methods suffer from the problem of highly correlated sources. The recent AV and CG algorithms could deal with these severe scenarios while lose their superiority when many sources need to be located.

4.9 The JIO Scheme for DOA Estimation

In this section, we introduce the JIO strategy according to the MV criterion to calculate the output power of each scanning angle and find peaks for the DOA estimation.

The JIO reduced-rank scheme for the DOA estimation includes a transformation matrix $\boldsymbol{T}_r$, which is structured as a bank of r full-rank vectors $\boldsymbol{t}_l = [t_{1,l}, t_{2,l}, \ldots, t_{m,l}]^T \in \mathbb{C}^{r\times 1}$ $l = 1, \ldots, r$ as given by $\boldsymbol{T}_r = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_r]$, and a reduced-rank filter with $\bar{\boldsymbol{w}}_\theta = [\bar{w}_{\theta,1}, \bar{w}_{\theta,2}, \ldots, \bar{w}_{\theta,r}]^T \in \mathbb{C}^{r\times 1}$. The transformation matrix generates the reduced-rank received vector $\bar{\boldsymbol{x}}(i)$, which is processed by the reduced-rank filter for computing the output power of the current scanning angle. Both $\boldsymbol{T}_r$ and $\bar{\boldsymbol{w}}_\theta$ are formulated by minimizing the MV cost function

$$\hat{\theta} = \arg\min_{\theta} \bar{\boldsymbol{w}}_\theta^H \boldsymbol{T}_r^H \boldsymbol{R} \boldsymbol{T}_r \bar{\boldsymbol{w}}_\theta; \quad \text{subject to } \bar{\boldsymbol{w}}_\theta^H \boldsymbol{T}_r^H \boldsymbol{a}(\theta) = 1. \tag{4.46}$$

By using the Lagrange multiplier and considering the constraint condition, we can get the expressions for $\boldsymbol{T}_r$ and $\bar{\boldsymbol{w}}_\theta$, which are

$$\boldsymbol{T}_r = \frac{\boldsymbol{R}^{-1}\boldsymbol{a}(\theta)}{\boldsymbol{a}^H(\theta)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta)} \frac{\bar{\boldsymbol{w}}_\theta^H}{\|\bar{\boldsymbol{w}}_\theta\|^2}, \tag{4.47}$$

$$\bar{\boldsymbol{w}}_\theta = \frac{\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta)}{\bar{\boldsymbol{a}}^H(\theta)\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta)}, \tag{4.48}$$

where $\bar{\boldsymbol{R}} = \mathbb{E}[\bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)] \in \mathbb{C}^{r\times r}$ is the reduced-rank covariance matrix and $\bar{\boldsymbol{a}}(\theta) = \boldsymbol{T}_r^H \boldsymbol{a}(\theta) \in \mathbb{C}^{r\times 1}$ is the reduced-rank steering vector of the current scanning angle. Note that for a small number of snapshots ($N < m$), $\boldsymbol{R}^{-1}$ and $\bar{\boldsymbol{R}}^{-1}$ are in practice calculated by either employing diagonal loading or the pseudo-inverse. We notice that the proposed reduced-rank weight vector $\bar{\boldsymbol{w}}_\theta$ is more general for dealing with the DOA estimation. Specifically, for $r = m$ and $\boldsymbol{T}_r = \boldsymbol{I}$, it is equivalent to the MV weight solution, and, for $1 < r < m$, it operates under a lower dimension and thus reduces the complexity.

The output power for each scanning angle can be calculated by substituting the expressions of $\boldsymbol{T}_r$ in (4.47) and $\bar{\boldsymbol{g}}_\theta$ in (4.48) into (4.46), which yields

$$P(\theta) = \left(\bar{\boldsymbol{a}}^H(\theta)\bar{\boldsymbol{R}}^{-1}\bar{\boldsymbol{a}}(\theta)\right)^{-1}. \tag{4.49}$$

By searching all possible angles, we could find peaks in the output power spectrum that corresponds to the DOAs of the sources.

4.10 Proposed Reduced-Rank Algorithms

In this section, we derive a constrained RLS algorithm based on the JIO reduced-rank scheme for the DOA estimation. The proposed algorithm jointly estimates the transformation matrix and the reduced-rank weight vector. A SS technique is employed in the algorithm to deal with the problem of highly correlated sources for the resolution improvement.

The transformation matrix $\boldsymbol{T}_r$ and the reduced-rank weight vector $\bar{\boldsymbol{w}}_\theta$ are computed according to the MV design in (4.46). Using the method of LS, the constraint in (4.46) can be incorporated by the method of Lagrange multiplier in the form

$$L_{\text{un}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}_\theta(i)\big) = \sum_{l=1}^{i} \alpha^{i-l} |\bar{\boldsymbol{w}}_\theta^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(l)|^2 + 2\,\Re\{\lambda[\bar{\boldsymbol{w}}_\theta^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta) - 1]\}, \tag{4.50}$$

where α is a forgetting factor, which is a positive constant close to, but less than 1, and λ is the Lagrange multiplier.

Fixing $\bar{\boldsymbol{w}}_\theta(i)$ and $\boldsymbol{T}_r(i)$, respectively, and take the gradient of $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}_\theta(i)$, solving the Lagrange multiplier, we have

$$\boldsymbol{T}_r(i) = \frac{\hat{\boldsymbol{R}}^{-1}(i)\boldsymbol{a}(\theta)}{\boldsymbol{a}^H(\theta)\hat{\boldsymbol{R}}^{-1}(i)\boldsymbol{a}(\theta)} \frac{\bar{\boldsymbol{w}}_\theta^H(i)}{\|\bar{\boldsymbol{w}}_\theta(i)\|^2}, \tag{4.51}$$

$$\bar{\boldsymbol{w}}_\theta(i) = \frac{\hat{\bar{\boldsymbol{R}}}^{-1}(i)\bar{\boldsymbol{a}}(\theta)}{\bar{\boldsymbol{a}}^H(\theta)\hat{\bar{\boldsymbol{R}}}^{-1}(i)\bar{\boldsymbol{a}}(\theta)}, \tag{4.52}$$

where $\hat{\boldsymbol{R}}(i) = \sum_{l=1}^{i} \alpha^{i-l}\boldsymbol{x}(l)\boldsymbol{x}^H(l) \in \mathbb{C}^{m\times m}$ and $\hat{\bar{\boldsymbol{R}}}(i) = \sum_{l=1}^{i} \alpha^{i-l}\bar{\boldsymbol{x}}(l)\bar{\boldsymbol{x}}^H(l) \in \mathbb{C}^{r\times r}$ are the estimates of the full-rank and reduced-rank covariance matrix at time instant i, respectively. Their recursive forms are $\hat{\boldsymbol{R}}(i) = \alpha\hat{\boldsymbol{R}}(i-1) + \boldsymbol{x}(i)\boldsymbol{x}^H(i)$ and $\hat{\bar{\boldsymbol{R}}}(i) = \alpha\hat{\bar{\boldsymbol{R}}}(i) = \alpha\hat{\bar{\boldsymbol{R}}}(i) + \bar{\boldsymbol{x}}(i)\bar{\boldsymbol{x}}^H(i)$.

Note that the expression of the transformation matrix in (4.51) is a function of $\bar{\boldsymbol{w}}_\theta(i)$ while the reduced-rank weight vector obtained from (4.52) depends on $\boldsymbol{T}_r(i)$. The proposed algorithm provides an iterative exchange of information between the transformation matrix and the reduced-rank weight vector and thus leads to an improved performance.

Now, we utilize the matrix inversion lemma [18] to realize the DOA estimation without the matrix inversion in (4.51) and (4.52). Specifically, defining $\hat{\boldsymbol{\Phi}}(i) = \hat{\boldsymbol{R}}(i)$, which can

be computed in a recursive form given by

$$\boldsymbol{k}(i) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}{1+\alpha^{-1}\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}, \tag{4.53}$$

$$\hat{\boldsymbol{\Phi}}(i) = \alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1) - \alpha^{-1}\boldsymbol{k}(i)\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1), \tag{4.54}$$

where $\boldsymbol{k}(i) \in \mathbb{C}^{m\times 1}$ is the Kalman gain vector. $\hat{\boldsymbol{\Phi}}(0) = \delta \boldsymbol{I}_{m\times m}$ where δ is a positive value needs to be set for numerical stability.

Given $\hat{\bar{\boldsymbol{\Phi}}}(i) = \hat{\bar{\boldsymbol{R}}}(i)$, we have

$$\bar{\boldsymbol{k}}(i) = \frac{\alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}{1+\alpha^{-1}\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}, \tag{4.55}$$

$$\hat{\bar{\boldsymbol{\Phi}}}(i) = \alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}(i)\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1), \tag{4.56}$$

where $\bar{\boldsymbol{k}}(i) \in \mathbb{C}^{r\times 1}$ is the reduced-rank gain vector, which is initialized by $\hat{\bar{\boldsymbol{\Phi}}}(0) = \bar{\delta}\boldsymbol{I}_{r\times r}$ with $\delta > 0$.

Substituting the recursive procedures (4.53)-(4.56) into the LS version method instead of the matrix inversion results in the JIO-RLS DOA estimation method, which is concluded in Table 4.1, where δ and $\bar{\delta}$ are selected according to the input signal-to-noise ratio (SNR) [18], $\hat{\boldsymbol{\Phi}}$ is the inversion of the estimate of the input covariance matrix after N snapshots, the scanning angle $\theta_n = n\triangle^o$, $\triangle^o$ is the search step, and $n = 1, 2, \ldots, 180^o/\triangle^o$. For a simple and convenient search, we make $180^o/\triangle^o$ an integer. It is necessary to initialize $\boldsymbol{T}_r(0)$ to start the iteration due to the dependence between $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{g}}_\theta(i)$. The JIO-RLS algorithm retains the advantageous property of the iterative exchange of information between the rank reduction matrix and the auxiliary reduced-rank vector, which avoids the degradation of the resolution, and utilizes a recursive procedure to compute $\hat{\boldsymbol{R}}^{-1}$ and $\hat{\bar{\boldsymbol{R}}}^{-1}$ for the reduction of the complexity.

The output power $P(\theta_n)$ is much higher if the scanning angle $\theta_n = \theta_k$, $k = 0, \ldots, q-1$, which corresponds on the positions of the sources, compared with other scanning angles with respect to the noise level. Thus, we can estimate the DOAs of the sources by finding the peaks in the output power spectrum.

For correlated sources, we use the forward/backword SS technique [103] in our LS version algorithm. It is based on the averaging the covariance matrix of identical overlapping arrays and so requires an array of identical elements equipped with some form of periodic structure, such as the ULA. For its application, we split a ULA antenna array

Tab. 4.1: Proposed JIO-RLS algorithm

Initialization:

$\boldsymbol{T}_r(0) = [\boldsymbol{I}_r^T \ \boldsymbol{0}_{r\times(m-r)}^T]$; δ, $\bar{\delta}$ =positive constants;

$\hat{\boldsymbol{\Phi}}(0) = \delta \boldsymbol{I}_{m\times m}$; $\hat{\bar{\boldsymbol{\Phi}}}(0) = \bar{\delta}\boldsymbol{I}_{r\times r}$.

Update for each time instant $i = 1, \ldots, N$

$\bar{\boldsymbol{x}}(i) = \boldsymbol{T}_r^H(i-1)\boldsymbol{x}(i)$

$\bar{\boldsymbol{a}}(\theta_n) = \boldsymbol{T}_r^H(i-1)\boldsymbol{a}(\theta_n)$

$\bar{\boldsymbol{k}}(i) = \frac{\alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}{1+\alpha^{-1}\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}$

$\hat{\bar{\boldsymbol{\Phi}}}(i) = \alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}(i)\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1)$

$\bar{\boldsymbol{w}}_{\theta_n}(i) = \frac{\hat{\bar{\boldsymbol{\Phi}}}(i)\bar{\boldsymbol{a}}(\theta_n)}{\bar{\boldsymbol{a}}^H(\theta_n)\hat{\bar{\boldsymbol{\Phi}}}(i)\bar{\boldsymbol{a}}(\theta_n)}$

$\boldsymbol{k}(i) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}{1+\alpha^{-1}\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}$

$\hat{\boldsymbol{\Phi}}(i) = \alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1) - \alpha^{-1}\boldsymbol{k}(i)\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1)$

$\boldsymbol{T}_r(i) = \frac{\hat{\boldsymbol{\Phi}}(i)\boldsymbol{a}(\theta_n)}{\boldsymbol{a}(\theta_n)^H\hat{\boldsymbol{\Phi}}(i)\boldsymbol{a}(\theta_n)} \frac{\bar{\boldsymbol{w}}_{\theta_n}^H(i)}{\|\bar{\boldsymbol{w}}_{\theta_n}(i)\|^2}$

Output power

$P(\theta_n) = 1/\big(\bar{\boldsymbol{a}}^H(\theta_n)\hat{\bar{\boldsymbol{\Phi}}}\bar{\boldsymbol{a}}(\theta_n)\big)$

into a set of forward and conjugate backward subarrays. From the forwarding, we divide the ULA into overlapping subarrays of size n, with the elements $\{1, \ldots, n\}$ forming the first subarrays, the elements $\{2, \ldots, n+1\}$ forming the second subarray, etc., and $J = m - n + 1$ as the number of subarrays. From the backwarding, the first backward subarray is formed using elements $\{m, m-1, \ldots, m-n+1\}$, the second subarray is formed using elements $\{m-1, m-2, \ldots, m-n\}$, and so on. Note that the selection of n needs to follow $n \geq q$ [103]. The SS preprocessing operates on $\boldsymbol{x}(i)$ to estimate the forward and backward subarray covariance matrices that are averaged to get the forward/conjugate backward smoothed covariance matrix. The JIO LS version algorithm is followed to realized the DOA estimation, which is called JIO-SS.

In terms of the computational cost, the Capon, MUSIC, and ESPRIT algorithms work with $O(m^3)$, and the recent AV and CG methods have a higher complexity [37]. The subspace tracking method denoted as approximated power iteration (API) [100] for use with the MUSIC and ESPRIT algorithms requires $O(qm + q^3)$. The API is significantly simpler than the direct eigen-decomposition [100], however, its complexity can become very high as the number of sources q becomes large. For the proposed algorithm, the complexity is $O(m^2)$ due to the use of the matrix inversion lemma [18]. Besides, the cost of computing $\hat{\boldsymbol{\Phi}}(i)$ is invariable for the grid search, namely, the result computed for the first scanning angle can be used for the rest. The complexity of the reduced-rank process is $O(mr)$, which is much less than $O(m^2)$ if r is much less than m for large arrays.

4.11 Simulations

Simulations are performed for a ULA with half a wavelength interelement spacing. We compare the proposed JIO-RLS algorithm with Capon's method, subspace-based methods with the API implementation [100], the ML method [38], and carry out $K = 1000$ runs to get each curve. BPSK sources separated by 3^o with powers $\sigma_s^2 = 1$ are considered and the noise is spatially and temporally white Gaussian. The search step is $\triangle^o = 0.5^o$. We suppose that the DOAs are resolved if $|\hat{\theta} - \theta_k| < |\theta_k - \theta_{k-1}|/2$ [36]. We measure the performance of the proposed DOA estimation algorithm by showing the probability of resolution, which is the probability of the number of the correct DOA estimations under $K = 1000$ runs. In the first experiment, there are $q = 2$ highly correlated sources in the system with correlation value $c = 0.9$, which are generated as follows [36]:

$$s_1 \sim \mathcal{N}(0, \sigma_s^2) \quad \text{and} \quad s_2 = cs_1 + \sqrt{1 - c^2}s_3, \tag{4.57}$$

where $s_3 \sim \mathcal{N}(0, \sigma_s^2)$. The array size is $m = 30$ and the number of snapshots $N = 10$ is fixed. We set $\alpha = 0.998$, $r = 4$, and $\delta = \bar{\delta} = 10^{-3}$. The SS technique is employed in the existing and proposed algorithms (except ML) to improve the resolution. The subarray size is $n = 28$ for the proposed method. The probability of resolution [32, 36, 37] is plotted against the input SNR values. The resolution of the ML algorithm is superior to that of other methods at the low SNR values. However, the heavy computational load prevents its use in practice for large arrays. The proposed JIO-SS algorithm outperforms other existing methods except ML for all input SNR values.

In Fig.4.8, we set the sources to be uncorrelated but increase the number of sources by setting $q = 10$. The number of snapshots is $N = 20$ and the array size is $m = 40$. From Fig.4.8, the curves between the proposed and the ESPRIT algorithms are shown to intersect when the input SNR values increase. The proposed JIO-RLS DOA estimation algorithm shows a better resolution with low SNRs. The complexity of the ESPRIT algorithm is high due to the eigen decomposition. The API approach estimates the signal subspace with a low-complexity to implement ESPRIT. This estimation is insufficient with a small number of snapshots and thus results in a poorer resolution, so does MUSIC(API).

In Fig.4.9, we keep the scenario as that in Fig. 4.8 and assess the root-mean-squared error (RMSE) performance of the proposed and existing algorithms, and compare them with the Cramér-Rao bound (CRB), which is calculated according to [39]. The RMSE here is the root-mean-squared error between the estimated DOAs and the actual ones. As can be seen from this figure, the RMSE values of the proposed algorithm are around 10 dB away from the CRB in the threshold region (low input SNR) and then closely follow

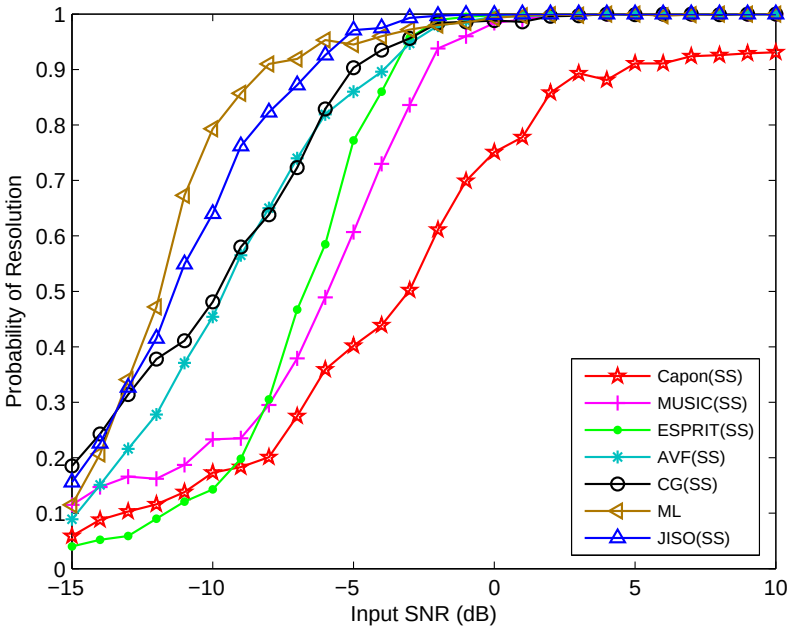

Fig. 4.7: Probability of resolution versus input SNR.

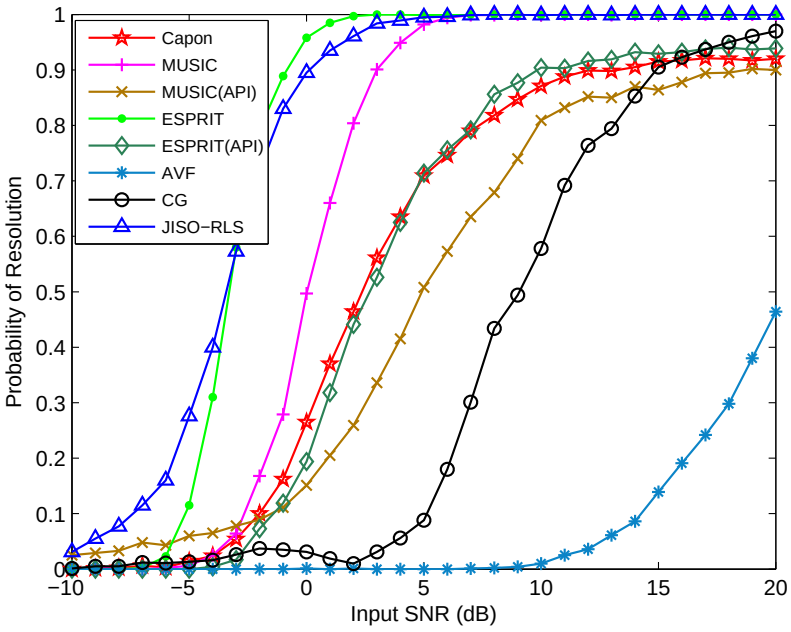

Fig. 4.8: Probability of resolution versus input SNR.

the CRB curve with the increase of the SNR. The improved RMSE performance of the proposed algorithm over existing methods is evident.

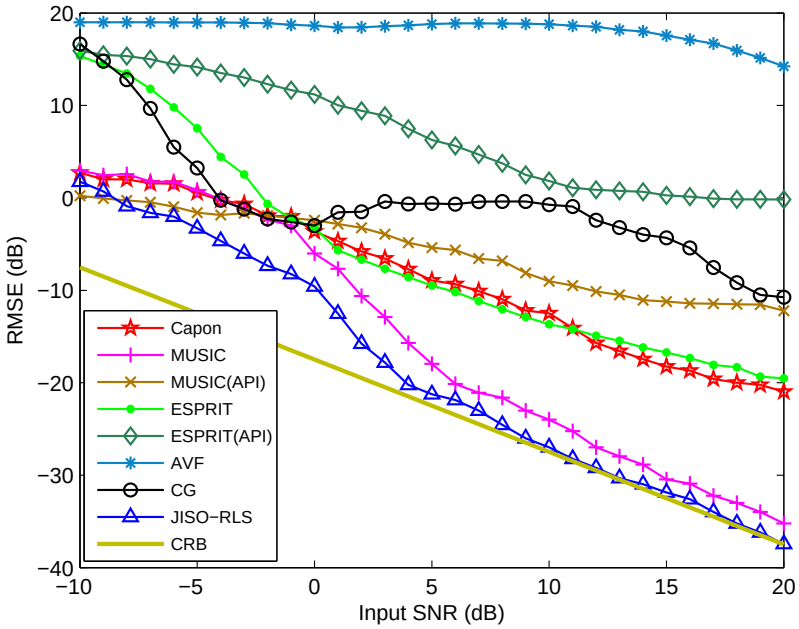

Fig. 4.9: RMSE versus input SNR.

4.12 Conclusions

In the first part of this chapter, we proposed reduced-rank CMV beamforming algorithms based on JIO of filters. The proposed scheme is based on a constrained joint iterative optimization of filters according to the MV criterion. We described CMV expressions for the design of the transformation matrix and the reduced-rank filter and developed the RLS adaptive algorithm for their efficient implementation along with an automatic rank selection technique. An analysis of the stability and the convergence properties of the proposed algorithms was presented and semi-analytical expressions were derived for predicting the MSE performance. In the second part of this chapter, we introduced a reduced-rank method based on a rank reduction matrix and an auxiliary reduced-rank parameter vector for DOA estimation. The rank reduction matrix maps the full-rank covariance matrix into a lower dimension and the auxiliary reduced-rank vector is combined to calculate the output power spectrum for each scanning angle. We have derived an efficient algorithm to jointly update the rank reduction matrix and the auxiliary vector. The proposed JIO-RLS algorithm shows a good resolution for large arrays with uncorrelated or correlated sources.

5. ADAPTIVE REDUCED-RANK CCM ALGORITHMS BASED ON JOINT ITERATIVE OPTIMIZATION OF FILTERS AND AUXILIARY VECTOR FILTERING FOR BEAMFORMING

Contents

5.1 Introduction

In chapter 4, we presented a review about the current reduced-rank signal processing algorithms [27, 28, 30, 82]. As we discussed in chapter 2, the CCM criterion is a positive measure of the deviation of the beamformer output from a constant modulus condition subject to a constraint on the array response of the desired signal, which provides more information than the CMV for the parameter estimation of constant modulus constellations in the beamformer design. Thus, it motivates us to develop the reduced-rank adaptive algorithms according to the CCM criterion for the beamformer design.

In this chapter, we introduce two types of reduced-rank adaptive algorithms according to the CCM criterion for beamforming, namely, the JIO reduced-rank algorithms and the auxiliary vector filtering (AVF) reduced-rank algorithm. The JIO scheme consists

of a bank of full-rank adaptive filters, which constitutes the transformation matrix and an adaptive reduced-rank filter that operates at the output of the bank of full-rank filters. This scheme provides an iterative exchange of information between the transformation matrix and the reduced-rank filter for weight adaptation and thus leads to improved convergence and tracking performance. The scheme is investigated in both direct-form processor (DFP) and the generalized sidelobe canceller (GSC) [18] structures. For each structure, a family of computationally efficient reduced-rank SG and RLS type algorithms are derived based on the JIO scheme. The GS technique is employed in the proposed methods to reformulate the transformation matrix for further improving performance. An automatic rank selection technique is developed according to the CM criterion to determine the most adequate rank for the proposed algorithms. Besides, the complexity comparison between the existing and proposed methods are given and the convergence properties for the reduced-rank scheme are analyzed.

The other type of reduced-rank adaptive algorithm is based on the AVF scheme, which was reported in [29, 30] and developed for adaptive filtering in [31, 82]. Its application in adaptive beamforming has been given in [91]. In the AVF scheme, an auxiliary vector is calculated by maximizing the cross correlation between the outputs of the reference vector filter and the previously auxiliary vector filters. The weight vector is obtained by subtracting the scaling auxiliary vector from the reference vector. In [31], the AVF algorithm iteratively generates a sequence of filters that converge to the CMV filter with a small number of samples.

Motivated by the fact that the CCM-based beamformers outperform the CMV ones for the CM signals, we propose an AVF algorithm based on the CCM design for robust adaptive beamforming. The beamformer structure decomposes the adaptive filter into a constrained (reference vector filter) and an unconstrained component (auxiliary vector filter). The constrained component is initialized with the array response of the desired signal to start the iteration and to ensure the CCM constraint, and the auxiliary vector in the unconstrained component can be iterated to meet the CM criterion. The weight vector is computed by means of suppressing the scaling unconstrained component from the constrained part. The main difference from the existing AVF technique is that, in the proposed CCM-based algorithm, the auxiliary vector and the scalar factor depend on each other and are jointly calculated according to the CM criterion (subject to different constraints). The proposed method provides an iterative exchange of information between the auxiliary vector and the scalar factor and also exploits the information about the CM signals, which leads to an improved performance.

The remainder of this chapter is organized as follows: in Section 5.2, the preliminary

works with the CCM design are reviewed. Section 5.3 presents the reduced-rank CCM beamformer designs for the DFP and GSC structures. The proposed JIO reduced-rank scheme based on the CM criterion subject to different constraints is presented in Section 5.4, and the proposed adaptive algorithms based on the JIO scheme are detailed in Section 5.5. The complexity and convergence analysis of the proposed methods is given in Section 5.6. Simulation results are provided and discussed in Section 5.7. The proposed CCM adaptive algorithm based on the AVF scheme is introduced in Section 5.8 and the simulation results are given in Section 5.9. The conclusions about the reduced-rank CCM adaptive algorithms are drawn in Section 5.10.

5.2 Preliminary Works

In this section, based on the system model explained in chapter 2, the full-rank beamformer design according to the CM criterion subject to the constraint on the array response is introduced for the DFP and GSC structures.

5.2.1 Full-rank Beamformer Design for the DFP

The full-rank CCM linear receiver design for beamforming is equivalent to determining a filter $\boldsymbol{w}(i) = [w_1(i), \ldots, w_m(i)]^T \in \mathbb{C}^{m\times 1}$ that provides an estimate of the desired symbol $y(i) = \boldsymbol{w}^H(i)\boldsymbol{x}(i)$. The calculation of the weight vector is based on the minimization of the cost function given in (2.4).

The full-rank CCM design can be implemented by the DFP processor, whose diagram is depicted in Fig. 5.1 (a). The weight expression obtained from the DFP structure is [28]

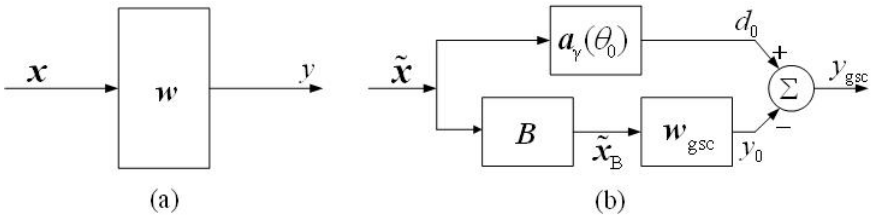

Fig. 5.1: (a) The full-rank DFP and (b) the full-rank GSC structures.

$$\boldsymbol{w}(i+1) = \boldsymbol{R}_{\text{yx}}^{-1}(i)\Big\{\boldsymbol{p}_{\text{yx}}(i) - \frac{\big[\boldsymbol{p}_{\text{yx}}^{H}(i)\boldsymbol{R}_{\text{yx}}^{-1}(i)\boldsymbol{a}(\theta_0) - \gamma\big]\boldsymbol{a}(\theta_0)}{\boldsymbol{a}^{H}(\theta_0)\boldsymbol{R}_{\text{yx}}^{-1}(i)\boldsymbol{a}(\theta_0)}\Big\}, \tag{5.1}$$

where $\boldsymbol{R}_{\text{yx}}(i) = \mathbb{E}[|y(i)|^2\boldsymbol{x}(i)\boldsymbol{x}^H(i)]$ and $\boldsymbol{p}_{\text{yx}}(i) = \mathbb{E}[y^*(i)\boldsymbol{x}(i)]$. Note that the expression in (5.1) is a function of previous values of $\boldsymbol{w}(i)$ (since $y(i) = \boldsymbol{w}^H(i)\boldsymbol{x}(i)$) and thus must be initialized to start the iteration for the solution. We keep the time index in $\boldsymbol{R}_{\text{yx}}(i)$ and $\boldsymbol{p}_{\text{yx}}(i)$ for the same reason.

5.2.2 Full-rank Beamformer Design for the GSC

The GSC structure converts the constrained optimization problem into an unconstrained one and adopts an alternative way to realize the beamformer design, as shown in Fig. 5.1 (b). The constraint in (2.4) is enclosed in the GSC processor. The full-rank CCM beamformer design of the GSC structure has been reported in [83]. Here, we employ an alternative way proposed in [84] to describe the design and simplify the derivation. The CM cost function of the GSC structure can be written as

$$J_{\text{cm-gsc}}\Big(\boldsymbol{w}(i)\Big) = \mathbb{E}\Big\{\big|\boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i) - \nu\big|^2\Big\}, \tag{5.2}$$

where $y_{\text{gsc}}(i) = \boldsymbol{w}^H(i)\boldsymbol{x}(i)$ and $\tilde{\boldsymbol{x}}(i) = y_{\text{gsc}}^*(i)\boldsymbol{x}(i)$. We set $\nu = 1$ in accordance with (2.4). Note that the expressions of $y_{\text{gsc}}(i)$ and $y(i)$ are similar but for the different structures. We regard $\tilde{\boldsymbol{x}}(i)$ as the new received vector to the full-rank CCM filter, as described in Fig. 1(b).

For the GSC structure, the output is $d_0(i) = \boldsymbol{a}_\gamma^H(\theta_0)\tilde{\boldsymbol{x}}(i)$, $\boldsymbol{a}_\gamma(\theta_0) = \gamma\boldsymbol{a}(\theta_0)$, γ is a positive scalar corresponding to that in (2.4), and the transformed vector $\tilde{\boldsymbol{x}}_B(i) \in \mathbb{C}^{(m-1)\times 1}$ is defined to be an output of the signal blocking matrix given by

$$\tilde{\boldsymbol{x}}_B(i) = \boldsymbol{B}\tilde{\boldsymbol{x}}(i), \tag{5.3}$$

where $\boldsymbol{B} \in \mathbb{C}^{(m-1)\times m}$ is the signal blocking matrix, which can be directly obtained by the singular value decomposition (SVD) or the QR decomposition algorithms [86]. Thus, $\boldsymbol{B}\boldsymbol{a}(\theta_0) = \boldsymbol{0}_{(m-1)\times 1}$ means that the term $\boldsymbol{B}$ effectively blocks any signal coming from the look direction θ_0. The GSC transformation can be concluded in an operator $\boldsymbol{S} = [\boldsymbol{a}_\gamma(\theta_0), \boldsymbol{B}^H]^H \in \mathbb{C}^{m\times m}$, which yields a transformed vector

$$\tilde{\boldsymbol{x}}_{\text{S}}(i) = \boldsymbol{S}\tilde{\boldsymbol{x}}(i) = \begin{bmatrix} \boldsymbol{a}_\gamma^H(\theta_0)\tilde{\boldsymbol{x}}(i) \\ \boldsymbol{B}\tilde{\boldsymbol{x}}(i) \end{bmatrix} = \begin{bmatrix} d_0(i) \\ \tilde{\boldsymbol{x}}_B(i) \end{bmatrix}, \tag{5.4}$$

and associated matrix $\boldsymbol{R}_{\tilde{x}_S}$ given by

$$\boldsymbol{R}_{\tilde{x}_S}(i) = \mathbb{E}\big[\tilde{\boldsymbol{x}}_S(i)\tilde{\boldsymbol{x}}_S^H(i)\big] = \boldsymbol{S}\boldsymbol{R}_{\tilde{x}}(i)\boldsymbol{S}^H = \begin{bmatrix} \gamma^2\sigma_0^2 & \tilde{\boldsymbol{p}}^H(i) \\ \tilde{\boldsymbol{p}}(i) & \boldsymbol{R}_{\tilde{x}_B}(i) \end{bmatrix}, \tag{5.5}$$

where $\boldsymbol{R}_{\tilde{x}}(i) = \mathbb{E}[\tilde{\boldsymbol{x}}(i)\tilde{\boldsymbol{x}}^H(i)] \in \mathbb{C}^{m\times m}$, $\sigma_0^2 = \boldsymbol{a}^H(\theta_0)\boldsymbol{R}_{\tilde{x}}(i)\boldsymbol{a}(\theta_0)$, $\tilde{\boldsymbol{p}}(i) = \mathbb{E}[d_0^*(i)\tilde{\boldsymbol{x}}_B(i)] \in \mathbb{C}^{(m-1)\times 1}$, and $\boldsymbol{R}_{\tilde{x}_B}(i) = \boldsymbol{B}\boldsymbol{R}_{\tilde{x}}(i)\boldsymbol{B}^H \in \mathbb{C}^{(m-1)\times(m-1)}$. The weight expression for the GSC structure in these transformed coordinates is $\boldsymbol{w}' = [1, -\boldsymbol{w}_{\text{gsc}}^H]^H \in \mathbb{C}^{m\times 1}$ with

$$\boldsymbol{w}_{\text{gsc}}(i+1) = \boldsymbol{R}_{\tilde{x}_B}^{-1}(i)\tilde{\boldsymbol{p}}_B(i), \tag{5.6}$$

and the weight solution is

$$\boldsymbol{w}(i+1) = \boldsymbol{S}^H\boldsymbol{w}'(i+1) = \boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{w}_{\text{gsc}}(i+1), \tag{5.7}$$

where $\tilde{\boldsymbol{p}}_B(i) = \mathbb{E}[(d_0^*(i) - 1)\tilde{\boldsymbol{x}}_B(i)] \in \mathbb{C}^{(m-1)\times 1}$. It is worth noticing that the expression in (4.6) is a function of previous values of $\boldsymbol{w}(i)$ and therefore must be initialized to start the iterations for the solution [85, 87].

The calculation of the weight solutions for the DFP and the GSC requires high complexity and suffers from numerical instability due to the matrix inversion appeared in (5.1) and (5.6), especially with large m. The SG and RLS-type algorithms can be employed to reduce the computational load but exhibit slow convergence and tracking performance when the dimension m is large.

5.3 Reduced-rank Beamformer Design

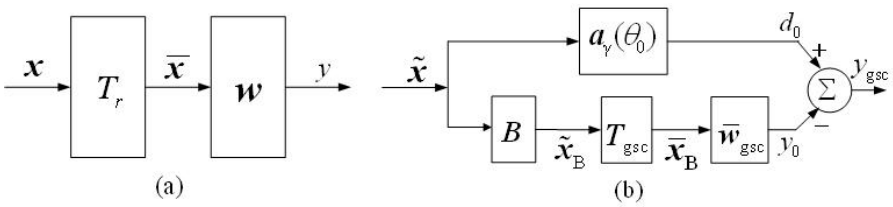

Fig. 5.2: (a) The reduced-rank DFP and (b) the reduced-rank GSC structures.

For large m, considering the high computational cost and poor performance associated with the full-rank filter, a number of works in the literature have been reported based on reduced-rank schemes [21]- [31], [88,89]. In this section, we will describe a reduced-rank framework that reduces the number of coefficients by mapping the received vector into a

lower dimensional subspace. The diagrams of the reduced-rank processors for the DFP and the GSC structures are depicted in Fig. 5.2(a) and Fig. 5.2(b), respectively.

5.3.1 Beamformer Design for the DFP

In the DFP structure, $\boldsymbol{T}_r \in \mathbb{C}^{m\times r}$ denotes the transformation matrix that includes a set of $m \times 1$ vectors for a r-dimensional subspace with $r \leq m$. The transformation matrix maps the received vector $\boldsymbol{x}(i)$ into its low-dimension version $\bar{\boldsymbol{x}}(i) \in \mathbb{C}^{r\times 1}$, which has been referred in chapter 5. We repeat it here for convenience:

$$\bar{\boldsymbol{x}}(i) = \boldsymbol{T}_r^H(i)\boldsymbol{x}(i), \tag{5.8}$$

where, in what follows, all r-dimensional quantities are denoted by an over bar. An adaptive reduced-rank CCM filter $\bar{\boldsymbol{w}}(i) \in \mathbb{C}^{r\times 1}$ follows the transformation matrix to produce the filter output $y(i) = \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{x}}(i)$.

The reduced-rank weight vector is expressed by [28]

$$\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{R}}^{-1}(i)\Big\{\bar{\boldsymbol{p}}(i) - \frac{\big[\bar{\boldsymbol{p}}^H(i)\bar{\boldsymbol{R}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_0) - \gamma\big]\bar{\boldsymbol{a}}(\theta_0)}{\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{R}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_0)}\Big\}, \tag{5.9}$$

where $\bar{\boldsymbol{R}}(i) = \mathbb{E}[|y(i)|^2\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{T}_r(i)] \in \mathbb{C}^{r\times r}$, $\bar{\boldsymbol{a}}(\theta_0) = \boldsymbol{T}_r^H\boldsymbol{a}(\theta_0) \in \mathbb{C}^{r\times 1}$, and $\bar{\boldsymbol{p}}(i) = \mathbb{E}[y^*(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)] \in \mathbb{C}^{r\times 1}$.

5.3.2 Beamformer Design for the GSC

The transformation matrix $\boldsymbol{T}_{\text{gsc}}(i) \in \mathbb{C}^{(m-1)\times r}$ for the GSC structure in Fig. 5.2(b) maps the transformed vector $\tilde{\boldsymbol{x}}_{\text{B}}(i)$ into a low-dimension version, as described by

$$\bar{\boldsymbol{x}}_B(i) = \boldsymbol{T}_{\text{gsc}}^H(i)\tilde{\boldsymbol{x}}_B(i). \tag{5.10}$$

The reduced-rank data vector $\bar{\boldsymbol{x}}_B(i)$ is processed by the filter $\bar{\boldsymbol{w}}_{\text{gsc}}(i) \in \mathbb{C}^{r\times 1}$ to get the sidelobe output $y_0(i) = \bar{\boldsymbol{w}}_{\text{gsc}}^H(i)\bar{\boldsymbol{x}}_B(i)$. The reduced-rank weight vector for the sidelobe of the GSC [18] is

$$\bar{\boldsymbol{w}}_{\text{gsc}}(i+1) = \bar{\boldsymbol{R}}_{\bar{x}_B}^{-1}(i)\bar{\boldsymbol{p}}_B(i), \tag{5.11}$$

where $\bar{\boldsymbol{R}}_{\bar{x}_B}(i) = \mathbb{E}[\boldsymbol{T}_{\text{gsc}}^H(i)\tilde{\boldsymbol{x}}_B(i)\tilde{\boldsymbol{x}}_B^H(i)\boldsymbol{T}_{\text{gsc}}(i)] \in \mathbb{C}^{r\times r}$ and $\bar{\boldsymbol{p}}_B(i) = \mathbb{E}[(d_0^*(i) -$

$1)\boldsymbol{T}_{\text{gsc}}^{H}(i)\tilde{\boldsymbol{x}}_B(i)] \in \mathbb{C}^{r\times 1}$.

The equivalent transformation operator for the GSC is $\bar{\boldsymbol{S}} = [\boldsymbol{a}_\gamma(\theta_0), \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}]^H \in \mathbb{C}^{(r+1)\times m}$ and the reduced-rank weight vector is $\bar{\boldsymbol{w}}' = [1, -\bar{\boldsymbol{w}}_{\text{gsc}}^H]^H \in \mathbb{C}^{(r+1)\times 1}$. The full-rank weight vector can be expressed as

$$\boldsymbol{w}(i+1) = \boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}(i+1)\bar{\boldsymbol{w}}_{\text{gsc}}(i+1). \tag{5.12}$$

From (5.9) and (5.11), the challenge left to us is how to efficiently design and calculate the transformation matrix $\boldsymbol{T}_r$. The PC method reported in [21] uses the eigenvectors of the interference-only covariance matrix corresponding to the eigenvalues of significant magnitude to construct the transformation matrix. The CS method [24], a counterpart of the PC method belonging to the eigen-decomposition family, forms the transformation matrix by using the eigenvectors which contribute the most towards maximizing the SINR and outperforms the PC method. Another family of adaptive reduced-rank filters such as the MSWF [26,27] and the AVF [29] generates a set of basis vectors as the transformation matrix that spans the same Krylov subspace [65, 80]. Despite the improved convergence and tracking performance achieved by these methods [26, 88], they require relatively high complexity and suffer from numerical problems.

5.4 Proposed CCM Reduced-rank Scheme

In this section, we introduce the proposed reduced-rank scheme based on the JIO approach. Two optimization problems according to the CM criterion subject to different constraints are described for the proposed scheme. Based on this scheme, we derive the expressions of the transformation matrix and the reduced-rank weight vector. For the sake of completeness, the proposed scheme is realized for both the DFP and the GSC structures.

5.4.1 Proposed CCM Reduced-rank Scheme for the DFP

Here we detail the principles of the proposed CCM reduced-rank scheme using a transformation based on adaptive filters. For the DFP structure depicted in Fig. 5.3(a), the proposed scheme employs a transformation matrix $\boldsymbol{T}_r(i) \in \mathbb{C}^{m\times r}$, which is responsible for the dimensionality reduction, to generate $\bar{\boldsymbol{x}}(i) \in \mathbb{C}^{r\times 1}$. The dimension

is reduced and the key features of the original signal are retained in $\bar{\boldsymbol{x}}(i)$ according to the CCM criterion. The transformation matrix is structured as a bank of r full-rank filters $\boldsymbol{t}_j(i) = [t_{1,j}(i), t_{2,j}(i), \ldots, t_{m,j}(i)]^T \in \mathbb{C}^{m\times 1}$, $(j = 1, \ldots, r)$ as given by $\boldsymbol{T}_r(i) = [\boldsymbol{t}_1(i), \boldsymbol{t}_2(i), \ldots, \boldsymbol{t}_r(i)]$. An adaptive reduced-rank filter $\bar{\boldsymbol{w}}(i) \in \mathbb{C}^{r\times 1}$ is then used to produce the output. The transformation matrix $\boldsymbol{T}_r(i)$ and the reduced-rank filter $\bar{\boldsymbol{w}}(i)$ are jointly optimized in the proposed scheme. The filter output is a function of the received vector, the transformation matrix, and the reduced-rank weight vector, which is

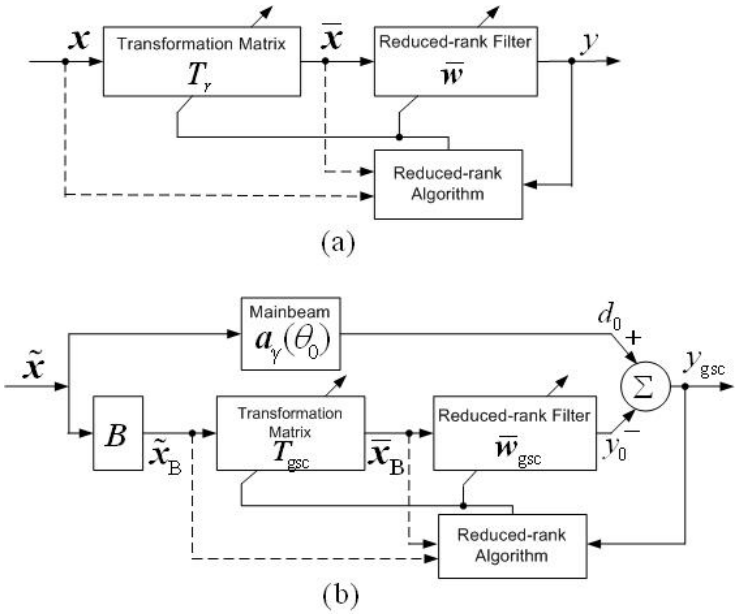

Fig. 5.3: Proposed reduced-rank scheme for (a) the DFP and (b) the GSC structures.

$$y(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i) = \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{x}}(i). \tag{5.13}$$

Note that for $r = 1$, the proposed scheme becomes a conventional full-rank filtering scheme with an additional weight parameter $\bar{w}_1$ that provides a gain. For $1 < r < m$, the signal processing tasks are changed, namely, the full-rank filters compute a subspace transformation matrix and the reduced-rank filter estimates the desired signal. For $r = m$ and $\boldsymbol{T}_r = \boldsymbol{I}_{m\times m}$, it is equivalent to the full-rank beamformer design. We find that DFP has the same structure as introduced in Fig. 4.1. The difference is that the transformation matrix and the reduced-rank adaptive filter are jointly optimized according to the different design criteria.

We describe two optimization problems according to the CM cost function subject to

different constraints for the proposed reduced-rank scheme, which are given by

$$\begin{aligned} \text{Problem 1}: \ \min \ J_{\text{cm}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)\big) &= \mathbb{E}\Big\{ \big[|y(i)|^2 - 1\big]^2 \Big\} \\ \text{subject to } \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) &= \gamma, \end{aligned} \tag{5.14}$$

$$\begin{aligned} &\text{Problem 2}: \ \min \ J_{\text{cm}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)\big) = \mathbb{E}\Big\{ \big[|y(i)|^2 - 1\big]^2 \Big\} \\ &\text{subject to } \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = \gamma \text{ and } \boldsymbol{T}_r^H(i)\boldsymbol{T}_r(i) = \boldsymbol{I}. \end{aligned} \tag{5.15}$$

Compared with (5.14), the problem in (5.15) has an orthogonal constraint on the transformation matrix, which is to reformulate $\boldsymbol{T}_r(i)$ for performance improvement. The transformation matrix generated from (5.14) has vectors that may perform a similar operation (e.g., take the same information twice or more), thereby making poor use of the data and losing performance. The subspace computed with (5.15), which spans the same subspace as $\boldsymbol{T}_r(i)$, generates basis vectors that are orthogonal to each other and which does not affect the noise statistically. The reformulated transformation matrix performs an efficient operation to keep all useful information in the generated reduced-rank received vector, which is important to estimate the desired signal and improve the performance. In the following, we will derive the CCM expressions of $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}(i)$ for solving (5.14) and (5.15).

The cost function in (5.14) can be transformed by the method of Lagrange multipliers into an unconstrained one, which is

$$\begin{aligned} L_{\text{un}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)\big) = \mathbb{E}\Big\{ \big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)|^2 - 1\big]^2 \Big\} \\ + 2\Re\Big\{ \lambda\big[\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) - \gamma\big] \Big\}, \end{aligned} \tag{5.16}$$

where λ is a scalar Lagrange multiplier.

Assuming $\bar{\boldsymbol{w}}(i)$ is known, minimizing (5.16) with respect to $\boldsymbol{T}_r(i)$, equating it to a null matrix and solving for λ, we have

$$\begin{aligned} \boldsymbol{T}_r(i+1) = \boldsymbol{R}_{\text{yx}}^{-1}(i)\Big\{ &\boldsymbol{p}_{\text{yx}}(i)\bar{\boldsymbol{w}}^H(i) - \\ &\frac{\big[\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{p}^H(i)\boldsymbol{R}_{\text{yx}}^{-1}(i)\boldsymbol{a}(\theta_0) - \gamma\big]\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_0)\boldsymbol{R}_{\text{yx}}^{-1}(i)\boldsymbol{a}(\theta_0)} \Big\} \bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i), \end{aligned} \tag{5.17}$$

where $\boldsymbol{p}_{\text{yx}}(i) = \mathbb{E}[y^*(i)\boldsymbol{x}(i)] \in \mathbb{C}^{m\times 1}$, $\boldsymbol{R}_{\text{yx}}(i) = \mathbb{E}[|y(i)|^2\boldsymbol{x}(i)\boldsymbol{x}^H(i)] \in \mathbb{C}^{m\times m}$, and $\bar{\boldsymbol{R}}_{\bar{w}}(i) = \mathbb{E}[\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)] \in \mathbb{C}^{r\times r}$. $\boldsymbol{R}_{\text{yx}}(i)$ and $\boldsymbol{p}_{\text{yx}}(i)$ are functions of previous values of $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}(i)$ due to the presence of $y(i)$. Therefore, it is necessary to initialize $\boldsymbol{T}_r(i)$

and $\bar{\boldsymbol{w}}(i)$ to estimate $\boldsymbol{R}_{\text{yx}}(i)$ and $\boldsymbol{p}_{\text{yx}}(i)$, and start the iteration.

On the other hand, assuming $\boldsymbol{T}_r(i)$ is known, minimizing (5.14) with respect to $\bar{\boldsymbol{w}}(i)$, equating it to a null vector, and solving for λ, we obtain

$$\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{R}}^{-1}(i)\Big\{\bar{\boldsymbol{p}}(i) - \frac{\big[\bar{\boldsymbol{p}}^H(i)\bar{\boldsymbol{R}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_0) - \gamma\big]\bar{\boldsymbol{a}}(\theta_0)}{\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{R}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_0)}\Big\}, \tag{5.18}$$

where $\bar{\boldsymbol{R}}(i) = \mathbb{E}[|y(i)|^2\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{T}_r(i)] \in \mathbb{C}^{r\times r}$, $\bar{\boldsymbol{p}}(i) = \mathbb{E}[y^*(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)] \in \mathbb{C}^{r\times 1}$, and $\bar{\boldsymbol{a}}(\theta_0) = \boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0)$.

Note that the expressions in (5.17) for the transformation matrix and (5.18) for the reduced-rank weight vector can be applied to solve the optimization problem (5.14). The orthogonal constraint in (5.15) can be realized by the Gram-Schmidt (GS) technique, which will be illustrated in the next section.

5.4.2 Proposed CCM Reduced-rank Scheme for the GSC

For the GSC structure, as depicted in Fig. 5.3(b), the proposed scheme utilizes a transformation matrix $\boldsymbol{T}_{\text{gsc}}(i) \in \mathbb{C}^{(m-1)\times r}$ to map the new transformed vector $\tilde{\boldsymbol{x}}_B(i) \in \mathbb{C}^{(m-1)\times 1}$ into a lower dimension, say $\bar{\boldsymbol{x}}_B(i) = \boldsymbol{T}_{\text{gsc}}^H(i)\tilde{\boldsymbol{x}}_B(i) \in \mathbb{C}^{r\times 1}$. In our design, the transformation matrix $\boldsymbol{T}_{\text{gsc}}(i)$ and the reduced-rank weight vector $\bar{\boldsymbol{w}}_{\text{gsc}}(i)$ for the sidelobe of the GSC are jointly optimized by minimizing the cost function

$$J_{\text{cm-gsc}}\big(\boldsymbol{T}_{\text{gsc}}(i), \bar{\boldsymbol{w}}_{\text{gsc}}(i)\big) = \mathbb{E}\Big\{\Big|\big[\boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}(i)\bar{\boldsymbol{w}}_{\text{gsc}}(i)\big]^H\tilde{\boldsymbol{x}}(i) - 1\Big|^2\Big\}, \tag{5.19}$$

where the expression in (5.19) for the GSC is obtained by substituting (5.12) into (5.11). This is an unconstrained cost function that corresponds to (5.14). From Fig. 5.3 (b), this structure essentially decomposes the adaptive weight vector into constrained (array response) and unconstrained components (see also Eq. (5.12)). The unconstrained component can be adjusted to meet the CM criterion since the constrained component always ensures that the constrained condition is satisfied. Thus, the proposed GSC framework converts the constrained optimization problem into an unconstrained one.

Assuming $\bar{\boldsymbol{w}}_{\text{gsc}}(i)$ and $\boldsymbol{T}_{\text{gsc}}(i)$ are given, respectively, minimizing (5.19) with respect to $\boldsymbol{T}_{\text{gsc}}(i)$ and $\bar{\boldsymbol{w}}_{\text{gsc}}(i)$, and solving the equations yields

$$\boldsymbol{T}_{\text{gsc}}(i+1) = \boldsymbol{R}_{\tilde{x}_B}^{-1}(i)\tilde{\boldsymbol{p}}_B(i)\bar{\boldsymbol{w}}_{\text{gsc}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}_{\text{gsc}}}^{-1}(i), \tag{5.20}$$

$$\bar{\boldsymbol{w}}_{\text{gsc}}(i+1) = \bar{\boldsymbol{R}}_{\bar{x}_B}^{-1}(i)\bar{\boldsymbol{p}}_B(i), \tag{5.21}$$

where $\boldsymbol{R}_{\tilde{\boldsymbol{x}}_B}(i)$ and $\tilde{\boldsymbol{p}}_B(i)$ have been defined in the previous section, $\bar{\boldsymbol{R}}_{\bar{w}_{\text{gsc}}}(i) = \mathbb{E}[\bar{\boldsymbol{w}}_{\text{gsc}}(i)\bar{\boldsymbol{w}}_{\text{gsc}}^H(i)] \in \mathbb{C}^{r\times r}$, $\bar{\boldsymbol{R}}_{\bar{x}_B}(i) = \mathbb{E}[\boldsymbol{T}_{\text{gsc}}^H(i)\tilde{\boldsymbol{x}}_B(i)\tilde{\boldsymbol{x}}_B^H(i)\boldsymbol{T}_{\text{gsc}}(i)] \in \mathbb{C}^{r\times r}$ and $\bar{\boldsymbol{p}}_B(i) = \mathbb{E}[(d_0^*(i) - 1)\boldsymbol{T}_{\text{gsc}}^H(i)\tilde{\boldsymbol{x}}_B(i)] \in \mathbb{C}^{r\times 1}$ with $d_0(i)$ being the desired response of the SOI. Again, the orthogonal constraint on the transformation matrix can be enforced in the optimization problem (5.19) and the GS technique is employed to realize this.

Note that the filter expressions in (5.17) and (5.18) for the DFP and (5.20) and (5.21) for the GSC are not closed-form solutions. In the DFP structure, the expression of the transformation matrix in (5.16) is a function of $\bar{\boldsymbol{w}}(i)$ and the reduced-rank weight vector obtained from (5.18) depends on $\boldsymbol{T}_r(i)$. It is necessary to set initial values of $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}(i)$ for the iteration procedures. Thus, initialization of the transformation matrix and the reduced-rank weight vector is not only to get a beamformer output $y(i)$ for estimating $\boldsymbol{R}_{\text{yx}}(i)$ and $\bar{\boldsymbol{R}}(i)$, but to start the iteration of the proposed scheme. In the case of the GSC, we initialize $\boldsymbol{T}_{\text{gsc}}(i)$ and $\bar{\boldsymbol{w}}_{\text{gsc}}(i)$ with the same intention.

Unlike the MSWF [26] and the AVF [29] techniques in which the transformation matrix is computed independently from the reduced-rank filter, the proposed scheme provides an iterative exchange of information between the transformation matrix and the reduced-rank filter, which leads to improved convergence and tracking performance. The transformation matrix reduces the dimension of the received vector whereas the reduced-rank filter attempts to estimate the desired signal. The key strategy lies in the joint iterative optimization of the filters. In the next section, we will derive iterative solutions via simple adaptive algorithms and introduce an automatic rank selection technique for the adaptation of the rank r.

5.5 Adaptive Algorithms of the CCM Reduced-rank Scheme

We derive SG and RLS type algorithms for the proposed CCM reduced-rank scheme. The devised algorithms are described for the DFP and the GSC structures, respectively, to perform joint iterative updates of the transformation matrix and the reduced-rank weight vector. The proposed SG and RLS type algorithms are used to solve Problem 1. The Gram-Schmidt (GS) technique is employed in the proposed algorithms and imposes an orthogonal constraint on the transformation matrix for further performance improvement, which is to solve Problem 2. An automatic rank selection technique is introduced to determine the most adequate rank for the proposed methods.

5.5.1 Stochastic Gradient Algorithms

Here, we derive the SG algorithms for both the DFP and the GSC structures.

The SG Algorithm for the DFP

Assuming $\bar{\boldsymbol{w}}(i)$ and $\boldsymbol{T}_r(i)$ are known, respectively, taking the instantaneous gradient of (5.16) with respect to $\boldsymbol{T}_r(i)$ and $\bar{\boldsymbol{w}}(i)$, we obtain

$$\nabla L_{T_r(i)}(i) = 2e(i)y^*(i)\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i) + 2\lambda^*_{T_r}\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i), \tag{5.22}$$

$$\nabla L_{\bar{w}(i)}(i) = 2e(i)y^*(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i) + 2\lambda^*_{\bar{w}}\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0), \tag{5.23}$$

where $e(i) = |y(i)|^2 - 1$.

Following the gradient rules $\boldsymbol{T}_r(i+1) = \boldsymbol{T}_r(i) - \mu_{T_r}\nabla L_{\text{un}_{T_r(i)}}(i)$ and $\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{w}}(i) - \mu_{\bar{w}}\nabla L_{\text{un}_{\bar{w}(i)}}(i)$, substituting (5.22) and (5.23) into them, respectively, and solving the Lagrange multipliers λ_{T_r} and $\lambda_{\bar{w}}$ by employing the constraint in (5.14), we obtain the following iterative SG algorithm for the DFP:

$$\boldsymbol{T}_r(i+1) = \boldsymbol{T}_r(i) - \mu_{T_r}e(i)y^*(i)\big[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i), \tag{5.24}$$

$$\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{w}}(i) - \mu_{\bar{w}}e(i)y^*(i)\big[\boldsymbol{I} - \frac{\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)}{\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0)}\big]\bar{\boldsymbol{x}}(i), \tag{5.25}$$

where μ_{T_r} and $\mu_{\bar{w}}$ are the corresponding step size factors for the DFP, which are small positive values.

The transformation matrix $\boldsymbol{T}_r(i)$ and the reduced-rank weight vector $\bar{\boldsymbol{w}}(i)$ operate together and exchange information at each time instant. A summary of the proposed CCM reduced-rank SG algorithm based on the JIO scheme, which is denominated as JIO-CCM-SG, is shown in Table 5.1, where the initialization values are set to satisfy the constraint in (5.14).

Tab. 5.1: The JIO-CCM-SG algorithm for DFP

Initialization:

$\boldsymbol{T}_r(1) = [\boldsymbol{I}_{r\times r}\ \boldsymbol{0}_{r\times(m-r)}]^T$;

$\bar{\boldsymbol{w}}(1) = \boldsymbol{T}_r^H(1)\boldsymbol{a}_\gamma(\theta_0)/(\|\boldsymbol{T}_r^H(1)\boldsymbol{a}_\gamma(\theta_0)\|^2)$.

Update for each time instant i

$y(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)$

$e(i) = |y(i)|^2 - 1$

$\boldsymbol{T}_r(i+1) = \boldsymbol{T}_r(i) - \mu_{T_r}e(i)y^*(i)\big[\boldsymbol{I} - \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\big]\boldsymbol{x}(i)\bar{\boldsymbol{w}}^H(i)$

$y(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i+1)\boldsymbol{x}(i)$

$e(i) = |y(i)|^2 - 1$

$\bar{\boldsymbol{a}}(\theta_0) = \boldsymbol{T}_r^H(i+1)\boldsymbol{a}(\theta_0)$

$\bar{\boldsymbol{x}}(i) = \boldsymbol{T}_r^H(i+1)\boldsymbol{x}(i)$

$\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{w}}(i) - \mu_{\bar{w}}e(i)y^*(i)\big[\boldsymbol{I} - \frac{\bar{\boldsymbol{a}}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)}{\bar{\boldsymbol{a}}^H(\theta_0)\bar{\boldsymbol{a}}(\theta_0)}\big]\bar{\boldsymbol{x}}(i)$

The SG Algorithm for the GSC

For the GSC structure, assuming $\bar{\boldsymbol{w}}_{\text{gsc}}(i)$ and $\boldsymbol{T}_{\text{gsc}}(i)$ are given in (5.19), respectively, we get

$$\nabla J_{\text{cm-gsc}_{T_{\text{gsc}}(i)}}(i) = e_{\text{gsc}}^*(i)\tilde{\boldsymbol{x}}_B(i)\bar{\boldsymbol{w}}_{\text{gsc}}^H(i), \tag{5.26}$$

$$\nabla J_{\text{cm-gsc}_{\bar{w}_{\text{gsc}}(i)}}(i) = e_{\text{gsc}}^*(i)\bar{\boldsymbol{x}}_B(i), \tag{5.27}$$

where $e_{\text{gsc}}(i) = 1 - \boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i)$ and $\boldsymbol{w}(i)$ is obtained from (5.12).

Substituting (5.26) and (5.27) into the gradient rules, we obtain the following iterative SG algorithm for the GSC:

$$\boldsymbol{T}_{\text{gsc}}(i+1) = \boldsymbol{T}_{\text{gsc}}(i) - \mu_{T_{\text{gsc}}}e_{\text{gsc}}^*(i)\tilde{\boldsymbol{x}}_B(i)\bar{\boldsymbol{w}}_{\text{gsc}}^H(i), \tag{5.28}$$

$$\bar{\boldsymbol{w}}_{\text{gsc}}(i+1) = \bar{\boldsymbol{w}}_{\text{gsc}}(i) - \mu_{\bar{w}_{\text{gsc}}}e_{\text{gsc}}^*(i)\bar{\boldsymbol{x}}_B(i), \tag{5.29}$$

where $\mu_{T_{\text{gsc}}}$ and $\mu_{\bar{w}_{\text{gsc}}}$ are the corresponding step size factors for the GSC. A summary of the proposed CCM reduced-rank SG algorithm for the GSC structure is shown in Table 5.2.

5.5.2 Recursive Least Squares Algorithms

In this part, we derive the RLS algorithms for both the DFP and the GSC structures.

Tab. 5.2: The JIO-CCM-SG algorithm for GSC

Initialization:

$\boldsymbol{T}_{\text{gsc}}(1) = [\boldsymbol{I}_{r\times r}\ \boldsymbol{0}_{r\times(m-r)}]^T; \qquad \bar{\boldsymbol{w}}_{\text{gsc}}(1) = \boldsymbol{I}_{r\times 1}.$

Update for each time instant i

$\boldsymbol{w}(i) = \boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}(i)\bar{\boldsymbol{w}}_{\text{gsc}}(i)$

$y_{\text{gsc}}(i) = \boldsymbol{w}^H(i)\boldsymbol{x}(i)$

$\tilde{\boldsymbol{x}}(i) = y^*_{\text{gsc}}(i)\boldsymbol{x}(i)$

$\tilde{\boldsymbol{x}}_B(i) = \boldsymbol{B}\tilde{\boldsymbol{x}}(i)$

$e_{\text{gsc}}(i) = 1 - \boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i)$

$\boldsymbol{T}_{\text{gsc}}(i+1) = \boldsymbol{T}_{\text{gsc}}(i) - \mu_{T_r}e^*_{\text{gsc}}(i)\tilde{\boldsymbol{x}}_B(i)\bar{\boldsymbol{w}}^H_{\text{gsc}}(i)$

$\boldsymbol{w}(i) = \boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}(i+1)\bar{\boldsymbol{w}}_{\text{gsc}}(i)$

$y_{\text{gsc}}(i) = \boldsymbol{w}^H(i)\boldsymbol{x}(i)$

$\tilde{\boldsymbol{x}}(i) = y^*_{\text{gsc}}(i)\boldsymbol{x}(i)$

$\tilde{\boldsymbol{x}}_B(i) = \boldsymbol{B}\tilde{\boldsymbol{x}}(i)$

$e_{\text{gsc}}(i) = 1 - \boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i)$

$\bar{\boldsymbol{x}}_B(i) = \boldsymbol{T}^H_{\text{gsc}}(i+1)\tilde{\boldsymbol{x}}_B(i)$

$\bar{\boldsymbol{w}}_{\text{gsc}}(i+1) = \bar{\boldsymbol{w}}_{\text{gsc}}(i) - \mu_{\bar{w}_{\text{gsc}}}e^*_{\text{gsc}}(i)\bar{\boldsymbol{x}}_B(i)$

The RLS Algorithm for the DFP

Considering the DFP case, the unconstrained least squares (LS) cost function is given by

$$\begin{aligned} L_{\text{un}}\big(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)\big) = &\sum_{l=1}^{i}\alpha^{i-l}\big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(l)|^2 - 1\big]^2 \\ &+ 2\Re\Big\{\lambda\big[\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) - \gamma\big]\Big\}, \end{aligned} \tag{5.30}$$

where α is a forgetting factor chosen as a positive constant close to, but less than 1.

Assuming $\bar{\boldsymbol{w}}(i)$ is known in (5.30), we obtain

$$\boldsymbol{T}_r(i+1) = \hat{\boldsymbol{R}}^{-1}_{\text{yx}}(i)\Big\{\hat{\boldsymbol{p}}_{\text{yx}}(i) - \frac{\big[\hat{\boldsymbol{p}}^H_{\text{yx}}(i)\hat{\boldsymbol{R}}^{-1}_{\text{yx}}(i)\boldsymbol{a}(\theta_0) - \gamma\big]\boldsymbol{a}(\theta_0)}{\boldsymbol{a}^H(\theta_0)\hat{\boldsymbol{R}}^{-1}_{\text{yx}}(i)\boldsymbol{a}(\theta_0)}\Big\}\frac{\bar{\boldsymbol{w}}^H(i)}{\|\bar{\boldsymbol{w}}(i)\|^2}, \tag{5.31}$$

where $\hat{\boldsymbol{R}}_{\text{yx}}(i) = \sum_{l=1}^{i}\alpha^{i-l}|y(l)|^2\boldsymbol{x}(l)\boldsymbol{x}^H(l)$ with $y(i)$ expressed in (5.13) and $\hat{\boldsymbol{p}}_{\text{yx}}(i) = \sum_{l=1}^{i}\alpha^{i-l}y^*(l)\boldsymbol{x}(l)$. The details of the derivation are given in Appendix G. Note that $\hat{\boldsymbol{R}}_{\text{yx}}(i)$ is not invertible if $i < m$. It can be implemented by employing the diagonal loading technique [8], [32]. This same procedure is also used for the remaining matrices.

To avoid the matrix inversion and reduce the complexity, we employ the matrix inversion lemma [18] to update $\hat{\boldsymbol{R}}^{-1}_{\text{yx}}(i)$ iteratively. Defining $\hat{\boldsymbol{\Phi}}(i) = \hat{\boldsymbol{R}}^{-1}_{\text{yx}}(i)$ for concise

presentation, the recursive estimation procedures are given by

$$\boldsymbol{k}(i) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}{(1/|y(i)|^2) + \alpha^{-1}\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}, \tag{5.32}$$

$$\hat{\boldsymbol{\Phi}}(i) = \alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1) - \alpha^{-1}\boldsymbol{k}(i)\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1), \tag{5.33}$$

where $\boldsymbol{k}(i) \in \mathbb{C}^{m\times 1}$ is the Kalman gain vector. We set $\hat{\boldsymbol{\Phi}}(0) = \delta \boldsymbol{I}_{m\times m}$, where $\delta > 0$ is a scalar for numerical stability.

Assuming $\boldsymbol{T}_r(i)$ is known in (5.31), we obtain

$$\bar{\boldsymbol{w}}(i+1) = \hat{\bar{\boldsymbol{R}}}^{-1}(i)\Big\{\hat{\bar{\boldsymbol{p}}}(i) - \frac{\big[\hat{\bar{\boldsymbol{p}}}^H(i)\hat{\bar{\boldsymbol{R}}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_0) - \gamma\big]\bar{\boldsymbol{a}}(\theta_0)}{\bar{\boldsymbol{a}}^H(i)\hat{\bar{\boldsymbol{R}}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_0)}\Big\}, \tag{5.34}$$

where $\hat{\bar{\boldsymbol{p}}}(i) = \sum_{l=1}^{i}\alpha^{i-l}y^*(l)\bar{\boldsymbol{x}}(l)$ and $\hat{\bar{\boldsymbol{R}}}(i) = \sum_{l=1}^{i}\alpha^{i-l}|y(l)|^2\bar{\boldsymbol{x}}(l)\bar{\boldsymbol{x}}^H(l)$.

Defining $\hat{\bar{\boldsymbol{\Phi}}}(i) = \hat{\bar{\boldsymbol{R}}}^{-1}(i)$ and employing the matrix inversion lemma, we have

$$\bar{\boldsymbol{k}}(i) = \frac{\alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}{(1/|y(i)|^2) + \alpha^{-1}\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}, \tag{5.35}$$

$$\hat{\bar{\boldsymbol{\Phi}}}(i) = \alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}(i)\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1), \tag{5.36}$$

where $\bar{\boldsymbol{k}} \in \mathbb{C}^{r\times 1}$ is the reduced-rank gain vector and the recursive procedures are implemented by initializing $\hat{\bar{\boldsymbol{\Phi}}}(0) = \bar{\delta}\boldsymbol{I}_{r\times r}$ for $\bar{\delta} > 0$. In Table 5.3, we give the procedures of the proposed CCM reduced-rank RLS algorithm for the DFP structure.

The RLS Algorithm for the GSC

For the GSC structure, the LS cost function is given by

$$L_{\text{un-gsc}}(\boldsymbol{T}_{\text{gsc}}(i), \bar{\boldsymbol{w}}_{\text{gsc}}(i)) = \sum_{l=1}^{i}\alpha^{i-l}\Big\{\big[\boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}(i)\bar{\boldsymbol{w}}_{\text{gsc}}(i)\big]^H\tilde{\boldsymbol{x}}(l) - 1\Big\}^2. \tag{5.37}$$

Assuming the optimal reduced-rank weight vector $\bar{\boldsymbol{w}}_{\text{gsc}}$ is known, computing the gradient of (5.37) with respect to $\boldsymbol{T}_{\text{gsc}}(i)$, equating it equal to a zero matrix, and using the

Tab. 5.3: The JIO-CCM-RLS algorithm for DFP

Initialization:

$\boldsymbol{T}_r(1) = [\boldsymbol{I}_{r\times r}\ \boldsymbol{0}_{r\times(m-r)}]^T$;

$\bar{\boldsymbol{w}}(1) = \boldsymbol{T}_r^H(1)\boldsymbol{a}_\gamma(\theta_0)/(\|\boldsymbol{T}_r^H(1)\boldsymbol{a}_\gamma(\theta_0)\|^2)$;

$\hat{\boldsymbol{\Phi}}(0) = \delta\boldsymbol{I}_{m\times m}$, $\hat{\bar{\boldsymbol{\Phi}}}(0) = \bar{\delta}\boldsymbol{I}_{r\times r}$, $\hat{\boldsymbol{p}}_{\text{yx}}(0) = \boldsymbol{0}_{m\times 1}$,$\hat{\bar{\boldsymbol{p}}}(0) = \boldsymbol{0}_{r\times 1}$.

Update for each time instant i

$y(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)$

$\hat{\boldsymbol{p}}_{\text{yx}}(i) = \alpha\hat{\boldsymbol{p}}_{\text{yx}}(i-1) + y^*(i)\boldsymbol{x}(i)$

$\boldsymbol{k}(i) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}{(1/|y(i)|^2)+\alpha^{-1}\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1)\boldsymbol{x}(i)}$

$\hat{\boldsymbol{\Phi}}(i) = \alpha^{-1}\hat{\boldsymbol{\Phi}}(i-1) - \alpha^{-1}\boldsymbol{k}(i)\boldsymbol{x}^H(i)\hat{\boldsymbol{\Phi}}(i-1)$

$\boldsymbol{T}_r(i+1) = \hat{\boldsymbol{\Phi}}(i)\Big\{\hat{\boldsymbol{p}}_{\text{yx}}(i) - \frac{\left[\hat{\boldsymbol{p}}^H(i)\hat{\boldsymbol{\Phi}}(i)\boldsymbol{a}(\theta_0)-\gamma\right]\boldsymbol{a}(\theta_0)}{\boldsymbol{a}^H(\theta_0)\hat{\boldsymbol{\Phi}}(i)\boldsymbol{a}(\theta_0)}\Big\}\frac{\bar{\boldsymbol{w}}^H(i)}{\|\bar{\boldsymbol{w}}(i)\|^2}$

$y(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i+1)\boldsymbol{x}(i)$

$\bar{\boldsymbol{a}}(\theta_0) = \boldsymbol{T}_r^H(i+1)\boldsymbol{a}(\theta_0)$

$\bar{\boldsymbol{x}}(i) = \boldsymbol{T}_r^H(i+1)\boldsymbol{x}(i)$

$\hat{\bar{\boldsymbol{p}}}(i) = \alpha\hat{\bar{\boldsymbol{p}}}(i-1) + y^*(i)\bar{\boldsymbol{x}}(i)$

$\bar{\boldsymbol{k}}(i) = \frac{\alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}{(1/|y(i)|^2)+\alpha^{-1}\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1)\bar{\boldsymbol{x}}(i)}$

$\hat{\bar{\boldsymbol{\Phi}}}(i) = \alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}(i)\bar{\boldsymbol{x}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i-1)$

$\bar{\boldsymbol{w}}(i+1) = \hat{\bar{\boldsymbol{\Phi}}}(i)\Big\{\hat{\bar{\boldsymbol{p}}}(i) - \frac{\left[\hat{\bar{\boldsymbol{p}}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i)\bar{\boldsymbol{a}}(\theta_0)-\gamma\right]\bar{\boldsymbol{a}}(\theta_0)}{\bar{\boldsymbol{a}}^H(i)\hat{\bar{\boldsymbol{\Phi}}}(i)\bar{\boldsymbol{a}}(\theta_0)}\Big\}$

similar derivation as for (5.31), we have

$$\boldsymbol{T}_{\text{gsc}}(i+1) = \hat{\boldsymbol{R}}_{\tilde{x}_B}^{-1}(i)\hat{\bar{\boldsymbol{p}}}_B(i)\frac{\bar{\boldsymbol{w}}_{\text{gsc}}^H(i)}{\|\bar{\boldsymbol{w}}_{\text{gsc}}(i)\|^2}, \tag{5.38}$$

where $\hat{\boldsymbol{R}}_{\tilde{x}_B}(i) = \sum_{l=1}^{i}\alpha^{i-l}\boldsymbol{B}\tilde{\boldsymbol{x}}(l)\tilde{\boldsymbol{x}}^H(l)\boldsymbol{B}^H$ and $\hat{\bar{\boldsymbol{p}}}_B(i) = \sum_{l=1}^{i}\alpha^{i-l}[d_0^*(l)-1]\tilde{\boldsymbol{x}}_B(l)$.

Setting $\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i) = \hat{\boldsymbol{R}}_{\tilde{x}_B}^{-1}(i)$ and employing the matrix inversion lemma yields

$$\boldsymbol{k}_B(i) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1)\tilde{\boldsymbol{x}}_B(i)}{1+\alpha^{-1}\tilde{\boldsymbol{x}}_B^H(i)\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1)\tilde{\boldsymbol{x}}_B(i)}, \tag{5.39}$$

$$\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i) = \alpha^{-1}\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1) - \alpha^{-1}\boldsymbol{k}_B(i)\tilde{\boldsymbol{x}}_B^H(i)\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1), \tag{5.40}$$

where $\boldsymbol{k}_B(i) \in \mathbb{C}^{m\times 1}$ is the gain vector and $\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(0) = \delta\boldsymbol{I}_{m\times m}$ for $\delta > 0$. Substituting (5.39) and (5.40) into (5.38), the transformation matrix can be expressed by

$$\boldsymbol{T}_{\text{gsc}}(i+1) = \boldsymbol{T}_{\text{gsc}}(i) - \boldsymbol{k}_B(i)\boldsymbol{e}_{\bar{w}_{\text{gsc}}}(i), \tag{5.41}$$

where $\boldsymbol{e}_{\bar{w}_{\text{gsc}}}(i) = [1-\tilde{\boldsymbol{x}}^H(i)\boldsymbol{w}(i)]\frac{\bar{\boldsymbol{w}}_{\text{gsc}}^H(i)}{\|\bar{\boldsymbol{w}}_{\text{gsc}}(i)\|^2}$ and $\boldsymbol{w}(i)$ is defined by (4.12).

Tab. 5.4: The JIO-CCM-RLS algorithm for GSC

Initialization:

$\boldsymbol{T}_{\text{gsc}}(1) = [\boldsymbol{I}_{r\times r}\ \boldsymbol{0}_{r\times(m-r)}]^T,\ \bar{\boldsymbol{w}}_{\text{gsc}}(1) = \boldsymbol{I}_{r\times 1}$

$\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(0) = \delta \boldsymbol{I}_{m\times m},\ \hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(0) = \bar{\delta}\boldsymbol{I}_{r\times r}.$

Update for each time instant i

$\boldsymbol{w}(i) = \boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}(i)\bar{\boldsymbol{w}}_{\text{gsc}}(i), \qquad y_{\text{gsc}}(i) = \boldsymbol{w}^H(i)\boldsymbol{x}(i)$

$\tilde{\boldsymbol{x}}(i) = y^*_{\text{gsc}}(i)\boldsymbol{x}(i),\ \tilde{\boldsymbol{x}}_B(i) = \boldsymbol{B}\tilde{\boldsymbol{x}}(i),\ e_{\text{gsc}}(i) = 1 - \boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i)$

$\boldsymbol{k}_B(i) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1)\tilde{\boldsymbol{x}}_B(i)}{1+\alpha^{-1}\tilde{\boldsymbol{x}}_B^H(i)\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1)\tilde{\boldsymbol{x}}_B(i)}$

$\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i) = \alpha^{-1}\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1) - \alpha^{-1}\boldsymbol{k}_B(i)\tilde{\boldsymbol{x}}_B^H(i)\hat{\boldsymbol{\Phi}}_{\tilde{x}_B}(i-1)$

$\boldsymbol{e}_{\bar{w}_{\text{gsc}}}(i) = [1 - \tilde{\boldsymbol{x}}^H(i)\boldsymbol{w}(i)]\frac{\bar{\boldsymbol{w}}^H_{\text{gsc}}(i)}{\|\bar{\boldsymbol{w}}_{\text{gsc}}(i)\|^2}$

$\boldsymbol{T}_{\text{gsc}}(i+1) = \boldsymbol{T}_{\text{gsc}}(i) - \boldsymbol{k}_B(i)\boldsymbol{e}_{\bar{w}_{\text{gsc}}}(i)$

$\bar{\boldsymbol{x}}_B(i) = \boldsymbol{T}^H_{\text{gsc}}(i+1)\tilde{\boldsymbol{x}}_B(i)$

$\bar{\boldsymbol{k}}_B(i) = \frac{\alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1)\bar{\boldsymbol{x}}_B(i)}{1+\alpha^{-1}\bar{\boldsymbol{x}}_B^H(i)\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1)\bar{\boldsymbol{x}}_B(i)}$

$\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i) = \alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}_B(i)\bar{\boldsymbol{x}}_B^H(i)\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1)$

$\boldsymbol{w}(i) = \boldsymbol{a}_\gamma(\theta_0) - \boldsymbol{B}^H\boldsymbol{T}_{\text{gsc}}(i+1)\bar{\boldsymbol{w}}_{\text{gsc}}(i)$

$e_{\text{gsc}}(i) = 1 - \boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i)$

$\bar{\boldsymbol{w}}_{\text{gsc}}(i+1) = \bar{\boldsymbol{w}}_{\text{gsc}}(i) - e^*_{\text{gsc}}(i)\bar{\boldsymbol{k}}_B(i)$

On the other hand, considering $\boldsymbol{T}_{\text{gsc}}(i)$ is known in (5.37), we get

$$\bar{\boldsymbol{w}}_{\text{gsc}}(i+1) = \hat{\bar{\boldsymbol{R}}}^{-1}_{\bar{x}_B}(i)\hat{\bar{\boldsymbol{p}}}_B(i), \tag{5.42}$$

where $\hat{\bar{\boldsymbol{R}}}_{\bar{x}_B}(i) = \boldsymbol{T}^H_{\text{gsc}}(i)\hat{\boldsymbol{R}}_{\tilde{x}_B}(i)\boldsymbol{T}_{\text{gsc}}(i)$ and $\hat{\bar{\boldsymbol{p}}}_B(i) = \boldsymbol{T}^H_{\text{gsc}}(i)\hat{\tilde{\boldsymbol{p}}}_B(i)$. The recursive estimation of $\hat{\bar{\boldsymbol{R}}}^{-1}_{\bar{x}_B}(i)$ is

$$\bar{\boldsymbol{k}}_B(i) = \frac{\alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1)\bar{\boldsymbol{x}}_B(i)}{1+\alpha^{-1}\bar{\boldsymbol{x}}^H_B(i)\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1)\bar{\boldsymbol{x}}_B(i)}, \tag{5.43}$$

$$\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i) = \alpha^{-1}\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}_B(i)\bar{\boldsymbol{x}}^H_B(i)\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i-1), \tag{5.44}$$

where $\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(i) = \hat{\bar{\boldsymbol{R}}}^{-1}_{\bar{x}_B}(i)$, $\bar{\boldsymbol{k}}_B(i) \in \mathbb{C}^{r\times 1}$ is the reduced-rank gain vector, and $\hat{\bar{\boldsymbol{\Phi}}}_{\bar{x}_B}(0) = \bar{\delta}\boldsymbol{I}_{r\times r}$ for $\bar{\delta} > 0$.

Substituting (5.43) and (5.44) into (5.42), we get a recursive expression of the reduced-rank weight vector, which is given by

$$\bar{\boldsymbol{w}}_{\text{gsc}}(i+1) = \bar{\boldsymbol{w}}_{\text{gsc}}(i) - e^*_{\text{gsc}}(i)\bar{\boldsymbol{k}}_B(i), \tag{5.45}$$

where $e_{\text{gsc}}(i) = 1 - \boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i)$. A summary of the proposed CCM reduced-rank RLS algorithm for the GSC is given in Table 5.4.

5.5.3 Gram-Schmidt Technique for Problem 2

As mentioned before, the transformation matrix $\boldsymbol{T}_r(i+1)$ for the DFP is constituted by a bank of full-rank filters $\boldsymbol{t}_j(i+1)$ $(j = 1, \ldots, r)$. According to the optimization problem in (5.15), by orthogonalizing vectors $\boldsymbol{t}_j(i+1)$ and reformulating the transformation matrix, the proposed algorithms could reach further improved performance. The reformulated transformation matrix is composed of r orthogonal vectors and thus keeps the useful information efficiently in the generated reduced-rank received vector, which is important to the parameter estimation. Another important aspect of this technique is that the orthonormal projection maintains the original statistical characteristics of the noise at the array output. The orthogonal procedure is realized by the Gram-Schmidt (GS) technique [19]. Specifically, after the iterative processes in (5.24) for SG and (5.31) for RLS, the GS technique is performed to modify the columns of the transformation matrix as follows:

$$\boldsymbol{t}_{j,\text{ort}}(i+1) = \boldsymbol{t}_j(i+1) - \sum_{l=1}^{j-1} \text{proj}_{\boldsymbol{t}_{l,\text{ort}}(i+1)} \boldsymbol{t}_j(i+1), \tag{5.46}$$

where $\boldsymbol{t}_{j,\text{ort}}(i+1)$ is the normalized orthogonal vector after the GS process. The projection operator is $\text{proj}_{\boldsymbol{t}_{l,\text{ort}}(i+1)} \boldsymbol{t}_j(i+1) = [\boldsymbol{t}_{l,\text{ort}}^H(i+1)\boldsymbol{t}_{l,\text{ort}}(i+1)]^{-1}[\boldsymbol{t}_{l,\text{ort}}^H(i+1)\boldsymbol{t}_j(i+1)]\boldsymbol{t}_{l,\text{ort}}(i+1)$.

The reformulated transformation matrix $\boldsymbol{T}_{r,\text{ort}}(i+1)$ is constructed after we obtain a set of orthogonal $\boldsymbol{t}_{j,\text{ort}}(i+1)$. By employing $\boldsymbol{T}_{r,\text{ort}}(i+1)$ to compute the reduced-rank weight vectors in (5.25) for SG and (5.34) for RLS, the proposed algorithms could achieve further improved performance. Following the same procedures, we can also apply the GS technique to the proposed algorithms for the GSC structure. We denominate the GS version of the SG and RLS algorithms as JIO-CCM-GS and JIO-CCM-RGS, respectively.

5.5.4 Automatic Rank Selection

The selection of the rank r impacts the performance of the proposed reduced-rank algorithms. Here, we introduce an adaptive method for selecting the rank according to the constant modulus criterion. We describe a rank selection method based on the CM criterion computed by the filters $\boldsymbol{T}_r^{(r)}(i)$ and $\bar{\boldsymbol{w}}^{(r)}(i)$, where the superscript $(\cdot)^{(r)}$ denotes the rank used for the adaptation at each time instant. We consider the rank adaptation technique for both the DFP and the GSC structures. Specifically, in the DFP structure, the rank is automatically selected for the proposed algorithms based on the exponentially-weighted a *posteriori* least-squares cost function according to the CM criterion, which

is

$$J_{\text{pcm}}\big(\boldsymbol{T}_r^{(r)}(i-1), \bar{\boldsymbol{w}}^{(r)}(i-1)\big) = \sum_{l=1}^{i} \varrho^{i-l}\big[|\bar{\boldsymbol{w}}^{(r)H}(l-1)\boldsymbol{T}_r^{(r)}(l-1)\boldsymbol{x}(l)|^2 - 1\big]^2, \tag{5.47}$$

where ϱ is the exponential weight factor that is required as the optimal rank r can change as a function of the time instant i. The rank adaptation scheme is similar to that given in chapter 5 and the key quantities to be updated for the rank adaptation are the transformation matrix $\boldsymbol{T}_r(i)$, the reduced-rank weight vector $\bar{\boldsymbol{w}}(i)$, the associated reduced-rank steering vector $\bar{\boldsymbol{a}}(\theta_0)$ and the matrix $\hat{\bar{\boldsymbol{\Phi}}}(i)$ (for RLS only).

For each time instant i, $\boldsymbol{T}_r^{(r)}(i)$ and $\bar{\boldsymbol{w}}^{(r)}(i)$ are updated along with the associated quantities $\bar{\boldsymbol{a}}(\theta_0)$ and $\hat{\bar{\boldsymbol{\Phi}}}(i)$ for a selected r according to the minimization of the cost function in (5.47). The developed automatic rank selection method is given by

$$r_{\text{opt}} = \arg \min_{r_{\min} \le j \le r_{\max}} J_{\text{pcm}}\big(\boldsymbol{T}_r^{(j)}(i-1), \bar{\boldsymbol{w}}^{(j)}(i-1)\big), \tag{5.48}$$

where j is an integer ranging between $r_{\min}$ and $r_{\max}$. Note that a smaller rank may provide faster adaptation during the initial stages of the estimation procedure and a slightly larger rank tends to yield a better steady-state performance. Our studies reveal that the range for which the rank r of the proposed algorithms have a positive impact on the performance is very limited, being from $r_{\min} = 3$ to $r_{\max} = 7$. With the case of large m, the rank r is significantly smaller than m and the additional computations do not increase the computational cost significantly.

The proposed algorithms with the rank adaptation technique can increase the convergence rate and improve the output performance, and r can be made fixed once the algorithms reach the steady-state. Simulation results will show how the developed rank adaptation technique works. Note that the same idea can be employed in the algorithms for the GSC structure. We omit this part for simplification and readability.

5.6 Analysis of the Proposed Algorithms

In this section, we provide a complexity analysis of the proposed reduced-rank algorithms and compare them with existing algorithms. An analysis of the optimization problem for the proposed reduced-rank scheme is also carried out.

5.6.1 Complexity Analysis

We evaluate the computational complexity of the proposed reduced-rank algorithms and compare them with the existing full-rank and reduced-rank algorithms based on the MSWF and the AVF techniques for the DFP and the GSC structures. With respect to each algorithm, we consider the CMV and the CCM design criteria. The computational requirements are described in terms of the number of complex arithmetic operations, namely, additions and multiplications. The complexity of the proposed and existing algorithms for the DFP is depicted in Table 5.5 and for the GSC in Table 5.6, where JIO-CMV-SG and JIO-CMV-RLS are the algorithms proposed in chapter 5. Since we did not consider the AVF technique for the GSC structure, we put its complexity for the DFP in both tables for comparison.

For the DFP structure, we can say that the complexity of the proposed reduced-rank SG type and extended GS version algorithms increases linearly with rm. The parameter m is more influential since r is selected around a small range that is much less than m for large arrays. The complexity of the proposed reduced-rank RLS type and GS version algorithms is higher than that of the SG type and quadratic with m and r. For the GSC structure, the complexity of the SG type algorithms has extra m^2 terms as compared to the DFP structure in terms of additions and multiplications due to the blocking matrix in the sidelobe canceller. There is no significant difference in complexity of the RLS type algorithms due to the presence of the blocking matrix since (5.41) and (5.45) are recursive expressions and, as compared to non-recursive versions, reduce the complexity.

In order to illustrate the main trends in what concerns the complexity of the proposed algorithms, we show in Fig. 5.4 and Fig. 5.5 the complexity of both the DFP and the GSC structures in terms of additions and multiplications versus the length of the filter m. Since the complexity of the current algorithms according to the CMV criterion is a little less than that of the CCM criterion, we only plot the curves for the CCM criterion for simplification. It is clear that the proposed SG type and extended GS version algorithms have a complexity slightly higher than the full-rank SG algorithm but much less than the existing algorithms based on the MSWF and the AVF techniques for both the DFP and the GSC structures. The curves of the proposed RLS type and GS version algorithms are situated between the full-rank RLS and the MSWF RLS algorithms in both figures. This complexity is less than those of the MSWF and the AVF based methods.

Tab. 5.5: Computational complexity of algorithms for DFP

Algorithm	Additions	Multiplications
FR-CMV-SG	$3m-1$	$4m+1$
FR-CCM-SG	$3m$	$4m+3$
FR-CMV-RLS	$4m^2-m-1$	$5m^2+5m-1$
FR-CCM-RLS	$5m^2+2m-1$	$6m^2+6m+3$
MSWF-CMV-SG	$rm^2+(r+1)m+2r-2$	$rm^2+2rm+5r+2$
MSWF-CCM-SG	$rm^2+(r+1)m+4r-2$	$rm^2+2rm+4r+4$
MSWF-CMV-RLS	$rm^2+(r+1)m+4r^2-3r-1$	$(r+1)m^2+2rm+5r^2+4r$
MSWF-CCM-RLS	$rm^2+(r+1)m+5r^2-r$	$(r+1)m^2+2rm+6r^2+7r+3$
AVF	$(4r+5)m^2+(r-1)m-2r-1$	$(5r+8)m^2+(3r+2)m$
JIO-CMV-SG	$4rm+m+2r-3$	$4rm+m+7r+3$
JIO-CMV-GS	$7rm-m-1$	$7rm-2m+8r+2$
JIO-CCM-SG	$4rm+m+2r-2$	$4rm+m+7r+6$
JIO-CCM-GS	$7rm-m$	$7rm-2m+8r+5$
JIO-CMV-RLS	$4m^2+(2r-1)m+4r^2-4r-1$	$5m^2+(3r+3)m+6r^2+4r$
JIO-CMV-RGS	$4m^2+(5r-3)m+4r^2-6r+1$	$5m^2+6rm+6r^2+5r-1$
JIO-CCM-RLS	$5m^2+rm+5r^2+3r-1$	$6m^2+(2r+6)m+5r^2+9r+3$
JIO-CCM-RGS	$5m^2+(4r-2)m+5r^2+r+1$	$6m^2+(5r+3)m+5r^2+10r+2$

Tab. 5.6: Computational complexity of algorithms for GSC

Algorithm	Additions	Multiplications
FR-CMV-SG	m^2+m-2	m^2+2m-1
FR-CCM-SG	m^2+m-1	m^2+2m+1
FR-CMV-RLS	$4m^2-6m+4$	$5m^2-4m$
FR-CCM-RLS	$4m^2-6m+2$	$5m^2-3m$
MSWF-CMV-SG	$(r+1)m^2-2rm+2r-1$	$(r+2)m^2-(r+2)m+2r+2$
MSWF-CCM-SG	$(r+1)m^2-2rm+2r$	$(r+2)m^2-(r+2)m+2r+4$
MSWF-CMV-RLS	$(r+1)m^2-2rm+3r^2+r-1$	$(r+2)m^2-(r+2)m+4r^2+4r$
MSWF-CCM-RLS	$(r+1)m^2-2rm+3r^2+r-1$	$(r+2)m^2-(r+1)m+4r^2+4r+1$
AVF	$(4r+5)m^2+(r-1)m-2r-1$	$(5r+8)m^2+(3r+2)m$
JIO-CMV-SG	$m^2+2rm-m-r$	$m^2+2rm+r+2$
JIO-CMV-GS	$m^2+5rm-3m-6r+4$	$m^2+5rm-3m-r+4$
JIO-CCM-SG	$m^2+2rm-m-r+1$	$m^2+2rm+r+4$
JIO-CCM-GS	$m^2+5rm-3m-6r+5$	$m^2+5rm-3m-r+6$
JIO-CMV-RLS	$4m^2+(2r-8)m+5r^2-2r+4$	$5m^2+(2r-6)m+7r^2+3r+2$
JIO-CMV-RGS	$4m^2+(5r-10)m+5r^2-7r+8$	$5m^2+(5r-9)m+7r^2+r+4$
JIO-CCM-RLS	$4m^2+(2r-7)m+5r^2-4r+3$	$5m^2+(2r-4)m+7r^2+2r+1$
JIO-CCM-RGS	$4m^2+(5r-9)m+5r^2-9r+7$	$5m^2+(5r-7)m+7r^2+3$

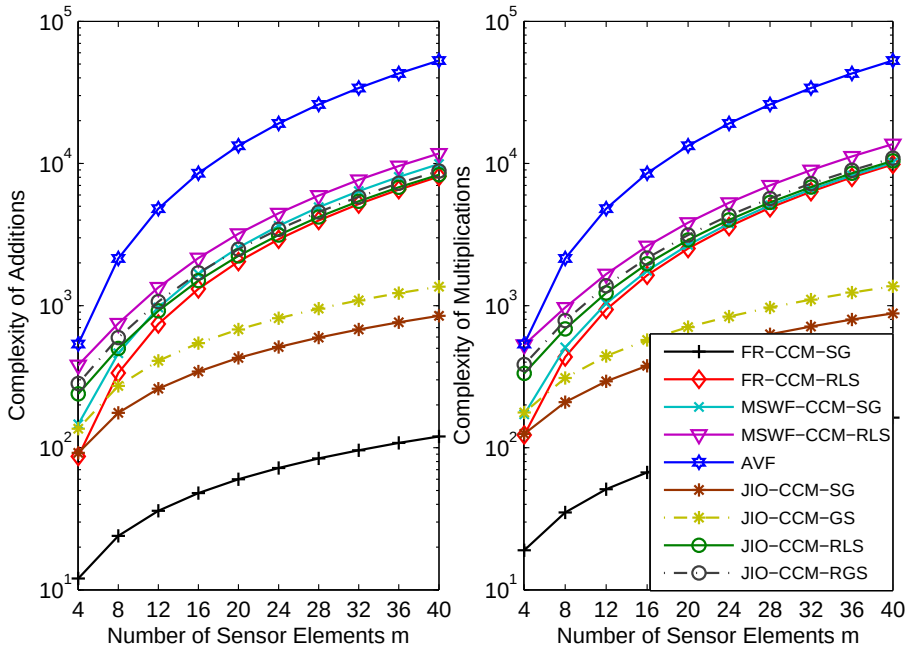

Fig. 5.4: Complexity in terms of arithmetic operations versus the length of the filter m for the DFP structure.

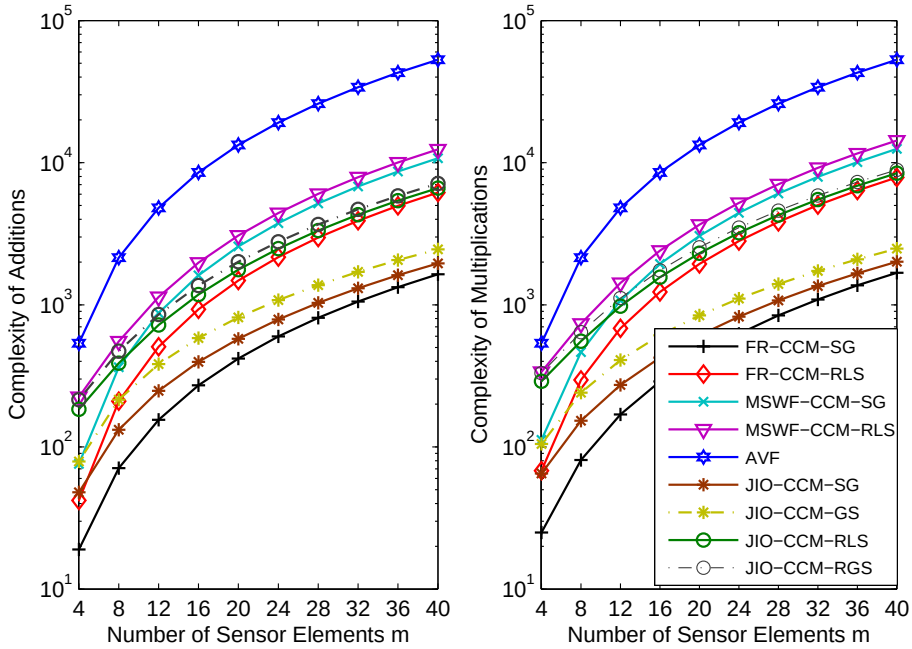

Fig. 5.5: Complexity in terms of arithmetic operations versus the length of the filter m for the GSC structure.

5.6.2 Analysis of the Optimization Problem

Unlike the prior work reported in [90] that considers the CCM criterion as a function of the weight vector, the analysis here is carried out for the proposed reduced-rank scheme according to the CCM criterion in (5.14), which depends on the transformation matrix $\boldsymbol{T}_r(i)$ and the reduced-rank weight vector $\bar{\boldsymbol{w}}(i)$. Our approach is based on expressing the output of the proposed scheme in a convenient form that renders itself to analysis. Note that the GSC structure is an alternative realization of the DFP beamformer design [15] and thus we only describe the convexity analysis of the proposed scheme for the DFP.

In the following, the reduced-rank CM cost function is considered first and the constraint is enforced during the analysis. In order to proceed, let us express $y(i)$ in an alternative and more convenient form as

$$\begin{aligned} y(i) &= \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i) \\ &= \bar{\boldsymbol{w}}^H(i) \begin{bmatrix} \boldsymbol{x}(i) & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{x}(i) & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{x}(i) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{t}_1^*(i) \\ \boldsymbol{t}_2^*(i) \\ \vdots \\ \boldsymbol{t}_r^*(i) \end{bmatrix} \\ &= \bar{\boldsymbol{w}}^H(i)\boldsymbol{\Psi}^T(i)\boldsymbol{\tau}^*(i), \end{aligned} \tag{5.49}$$

where $\boldsymbol{\Psi}(i) \in \mathbb{C}^{rm\times r}$ is a block diagonal matrix with the received vector $\boldsymbol{x}(i)$ and $\boldsymbol{\tau}^*(i) \in \mathbb{C}^{rm\times 1}$ is a vector with the columns of $\boldsymbol{T}_r(i)$ stacked on top of each other.

In order to analyze the proposed reduced-rank scheme, we define $\boldsymbol{f}(i) = [\bar{\boldsymbol{w}}^H(i)\ \boldsymbol{\tau}^T(i)]^T \in \mathbb{C}^{r(m+1)\times 1}$. Then, $y(i)$ can be expressed as

$$\begin{aligned} y(i) &= \boldsymbol{f}^H(i) \begin{bmatrix} \mathbf{0}_{r\times r} & \mathbf{0}_{r\times rm} \\ \boldsymbol{\Psi}(i) & \mathbf{0}_{rm\times rm} \end{bmatrix} \boldsymbol{f}(i) \\ &= \boldsymbol{f}^H(i)\boldsymbol{\Omega}(i)\boldsymbol{f}(i), \end{aligned} \tag{5.50}$$

where $\boldsymbol{\Omega}(i) \in \mathbb{C}^{r(m+1)\times r(m+1)}$ is a matrix which contains $\boldsymbol{\Psi}(i)$.

According to (5.50) and the definitions of $\boldsymbol{f}(i)$ and $\boldsymbol{\Omega}(i)$, we have

$$\begin{aligned} |y(i)|^2 &= \boldsymbol{f}^H(i)\boldsymbol{\Omega}(i)\boldsymbol{f}(i)\boldsymbol{f}^H(i)\boldsymbol{\Omega}^H(i)\boldsymbol{f}(i) \\ &= \sum_{j=1}^{r}\sum_{k=1}^{r} \boldsymbol{t}_{\bar{w}_j}^H(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{t}_{\bar{w}_k}(i), \end{aligned} \tag{5.51}$$

where $\boldsymbol{t}_{\bar{w}_j}(i) = \bar{w}_j(i)\boldsymbol{t}_j(i) \in \mathbb{C}^{m\times 1}$, $(j = 1, \ldots, r)$. The received vector can be expressed as $\boldsymbol{x}(i) = \sum_{l=0}^{q-1} C_l d_l(i)\boldsymbol{a}(\theta_l) + \boldsymbol{n}(i)$ with C_l being the signal amplitude and $d_l(i)$ is the transmitted bit of the lth user at time instant i, respectively. Note that s_l in (2.1) is replaced by $C_l d_l$.

For the sake of analysis, we will follow the assumption in [90] and consider a noise free case. For small noise variance σ_n^2, this assumption can be considered as a small perturbation and the analysis will still be applicable. For large σ_n^2, we remark that the term γ can be adjusted to ensure the convexity of the cost function. Under this assumption, we write the received vector as $\boldsymbol{x}(i) = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{C}\boldsymbol{d}(i)$, where $\boldsymbol{A}(\boldsymbol{\theta})$, as before, denotes the signature matrix, $\boldsymbol{C}(i) = \text{diag}[C_0, \ldots, C_{q-1}] \in \mathbb{C}^{q\times q}$, and $\boldsymbol{d}(i) = [d_0(i), \ldots, d_{q-1}(i)]^T \in \mathbb{C}^{q\times 1}$.

For simplicity, we drop the time instant in the quantities. Letting $\varsigma_l^j = C_l \boldsymbol{t}_{\bar{w}_j}^H \boldsymbol{a}(\theta_l)$ $(l = 0, \ldots, q-1,\ j = 1, \ldots, r)$, and $\boldsymbol{\varsigma}^j = [\varsigma_0^j, \ldots, \varsigma_{q-1}^j]^T \in \mathbb{C}^{q\times 1}$ and considering that d_l are independent random variables, the expectation of (5.51) can be written as

$$\mathbb{E}[\boldsymbol{t}_{\bar{w}_j}^H \boldsymbol{x}\boldsymbol{x}^H \boldsymbol{t}_{\bar{w}_k}] = \mathbb{E}[{\boldsymbol{\varsigma}^j}^H \boldsymbol{d}\boldsymbol{d}^H \boldsymbol{\varsigma}^k] = \mathbb{E}[\sum_{l=0}^{q-1} |d_l|^2 \varsigma_l^{j^*} \varsigma_l^k], \tag{5.52}$$

$$\mathbb{E}[|y|^2] = \mathbb{E}[\sum_{j=1}^{r}\sum_{k=1}^{r}\sum_{l=0}^{q-1} |d_l|^2 \varsigma_l^{j^*} \varsigma_l^k]. \tag{5.53}$$

Now, we consider the constraint in (5.14). To enforce the constraint in the analysis, we write it as

$$\bar{\boldsymbol{w}}^H \boldsymbol{T}_r^H \boldsymbol{a}(\theta_0) = \sum_{j=1}^{r} \boldsymbol{t}_{\bar{w}_j}^H \boldsymbol{a}(\theta_0). \tag{5.54}$$

From the definition of ς_l^j and (5.48), it is interesting to find

$$\sum_{j=1}^{r} \varsigma_0^j = C_0 \sum_{j=1}^{r} \boldsymbol{t}_{\bar{w}_j}^H \boldsymbol{a}(\theta_0) = C_0\gamma. \tag{5.55}$$

According to (5.55), the constraint can be enforced in the expression of (5.53), which

is given by

$$\begin{aligned}\mathbb{E}[|y|^2] &= \mathbb{E}[|d_0|^2 C_0^2\gamma^2] + \mathbb{E}[\sum_{j=1}^{r}\sum_{l=1}^{q-1}\varsigma_l^{j*}\varsigma_l^j] + \mathbb{E}[\sum_{j=1}^{r}\sum_{k\neq j}^{r}\sum_{l=1}^{q-1}\varsigma_l^{j*}\varsigma_l^k] \\ &= \mathbb{E}[|d_0|^2 C_0^2\gamma^2] + \mathbb{E}[\sum_{j=1}^{r}\tilde{\boldsymbol{\varsigma}}^{jH}\tilde{\boldsymbol{\varsigma}}^j] + \mathbb{E}[\sum_{j=1}^{r}\sum_{k\neq j}^{r}\tilde{\boldsymbol{\varsigma}}^{jH}\tilde{\boldsymbol{\varsigma}}^k],\end{aligned} \tag{5.56}$$

where $\tilde{\boldsymbol{\varsigma}}^j = [\varsigma_1^j, \ldots, \varsigma_{q-1}^j]^T \in \mathbb{C}^{(q-1)\times 1}$.

We can examine the convexity of (5.14) by substituting (5.56) into (5.14) and computing the Hessian matrix ($\boldsymbol{H}_j$, $j = 1, \ldots, r$) with respect to $\tilde{\boldsymbol{\varsigma}}^j$ using the expression [95]

$$\boldsymbol{H}_j = \frac{\partial}{\partial \tilde{\boldsymbol{\varsigma}}^{jH}}\frac{\partial J(\tilde{\boldsymbol{\varsigma}}^j)}{\partial \tilde{\boldsymbol{\varsigma}}^j}, \tag{5.57}$$

and testing if the terms are positive semi-definite. Specifically, $\boldsymbol{H}_j$ is positive semi-definite if $\boldsymbol{v}^H\boldsymbol{H}_j\boldsymbol{v} \geq 0$ for all nonzero $\boldsymbol{v} \in \mathbb{C}^{(q-1)\times(q-1)}$ [19]. Thus, the optimization problem is convex if the Hessian $\boldsymbol{H}_j$ is positive semi-definite for $j = 1, \ldots, r$. Note that $J(\tilde{\boldsymbol{\varsigma}}^j)$ is a constrained cost function since the constraint is enforced in (5.56).

Evaluating the partial differentiation in the expression given in (5.57) yields

$$\begin{aligned}\boldsymbol{H}_j = 2\mathbb{E}\Big\{ &\big[\sum_{j=1}^{r}\tilde{\boldsymbol{\varsigma}}^{jH}\tilde{\boldsymbol{\varsigma}}^j\boldsymbol{I} + \sum_{j=1}^{r}\tilde{\boldsymbol{\varsigma}}^j\tilde{\boldsymbol{\varsigma}}^{jH}\big] + \big[|d_0|^2C_0^2\gamma^2 - 1\big]\boldsymbol{I} \\ &+ \big[\sum_{j=1}^{r}\sum_{k\neq j}^{r}\tilde{\boldsymbol{\varsigma}}^{jH}\tilde{\boldsymbol{\varsigma}}^k\boldsymbol{I} + \sum_{j=1}^{r}\sum_{k\neq j}^{r}\tilde{\boldsymbol{\varsigma}}^j\tilde{\boldsymbol{\varsigma}}^{kH}\big]\Big\}.\end{aligned} \tag{5.58}$$

For different $j = 1, \ldots, r$, the results of Hessian $\boldsymbol{H}_j$ are the same. Thus, the convexity of the optimization problem can be ensured if $\boldsymbol{H}_j$ is positive semi-definite. Apparently, the terms in the first square brackets yield positive semi-definite matrices. The Hessian $\boldsymbol{H}_j$ depends on the remaining terms and in principle does not look like a positive semi-definite. However, there is a term γ^2 which is a design parameter and can be set to enforce the positive semi-definiteness of $\boldsymbol{H}_j$, thereby enforcing the convexity. The term γ is set to enforce the convexity and ensure the proposed methods work well, as illustrated in the simulations that follow.

5.7 Simulations

In this section, we evaluate the output SINR performance of the proposed adaptive reduced-rank algorithms and compare them with the existing methods. Specifically, we compare the proposed SG and RLS type algorithms with the full-rank (FR) SG and RLS and reduced-rank methods based on the MSWF and the AVF techniques for both the DFP and the GSC structures. For each algorithm, we consider the CMV and the CCM criteria for the beamformer design. We assume that the DOA of the desired user is known by the receiver. In each experiment, a total of $K = 1000$ runs are carried out to get the curves. For all simulations, the source power (including the desired user and interferers) is $\sigma_{\rm s}^2 = \sigma_{\rm i}^2 = 1$ with white Gaussian noise, and $\gamma = 1$. Simulations are performed by an ULA containing $m = 32$ sensor elements with half-wavelength interelement spacing.

In Fig. 5.6, we compare the proposed and existing algorithms according to the CMV and the CCM criteria for the DFP structure of the beamformer design. The simulation, which includes two experiments, shows the input SNR versus the output SINR. The input SNR is varied between -10 dB and 10 dB. The number of users is $q = 5$ with one desired user. The number of snapshots is fixed $N = 500$. Fig. 5.6 (a) plots the curves of the SG type algorithms based on the full-rank, MSWF, AVF and the proposed reduced-rank scheme, and Fig. 5.6 (b) shows the corresponding RLS type algorithms. The parameters used to obtain these curves are given and the rank r is selected to optimize the algorithms. From Fig. 5.6 (a), the output SINR of all SG type methods increases following the increase of the input SNR. The algorithms based on the CCM beamformer design outperform those based on the CMV since the CCM criterion is a positive measure of the beamformer output deviating from a constant modulus, which provides more information than the CMV for the parameter estimation of constant modulus constellations. The proposed CCM algorithms achieve better performance than the existing full-rank, MSWF and AVF ones. By employing the GS technique to reformulate the transformation matrix, the GS version algorithms achieve improved performance. Fig. 5.6 (b) verifies the same fact but for the RLS type algorithms. It is clear that the RLS type algorithms superior to the SG type ones for all input SNR values.

This simulation verifies that the performance of the adaptive algorithms based on the CCM beamformer design has a similar trend but is better than that based on the CMV for constant modulus constellations. Considering this fact, we will only compare the CCM based adaptive algorithms in the following part for simplification. Note that all the methods in this simulation are for the DFP structure. The algorithms for the GSC structure show a similar performance, which is given in the next part.

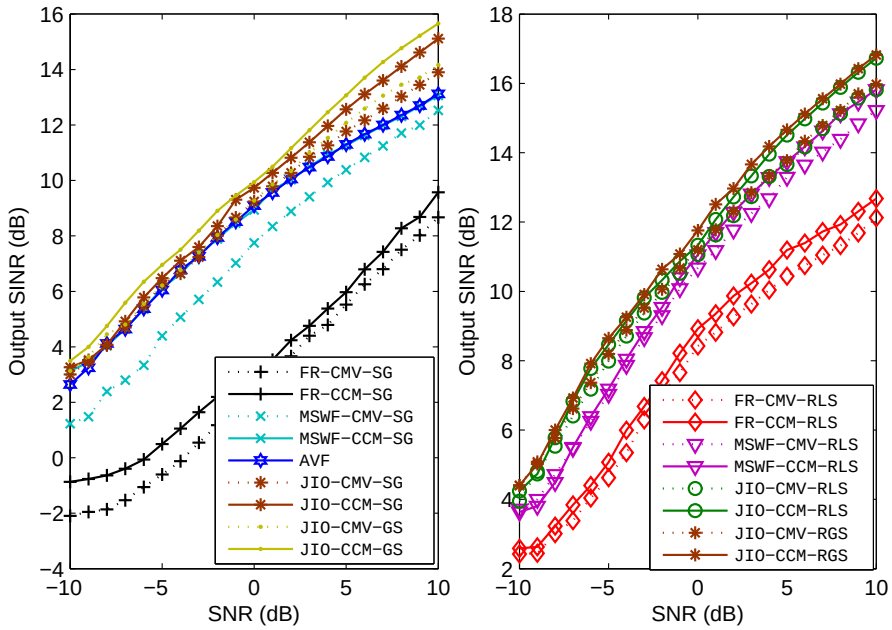

Fig. 5.6: Output SINR versus input SNR with $m = 32$, $q = 5$, SNR$= 10$ dB, (a) $\mu_{T_r} = 0.002$, $\mu_{\bar{w}} = 0.001$, $r = 5$ for SG, $\mu_{T_r} = 0.003$, $\mu_{\bar{w}} = 0.0007$, $r = 5$ for GS; (b) $\alpha = 0.998$, $\delta = \bar{\delta} = 0.03$, $r = 5$ for RLS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.028$, $r = 5$ for RGS of the proposed CCM reduced-rank scheme.

We evaluate the output SINR performance of the proposed and existing algorithms against the number of snapshots for both the DFP and the GSC structures in Fig. 5.7 and Fig. 5.8, respectively. The number of snapshots is $N = 500$ to ensure that the considered methods reach their steady-state. There are $q = 7$ users in the system, including one desired user. The input SNR is 10 dB. In Fig. 5.7, the convergence of the proposed SG type and extended GS version algorithms is close to the RLS type algorithm based on the MSWF, and the steady-state performance is better than other SG type methods based on the full-rank, MSWF and AVF. The convergence of the proposed RLS type and GS version algorithms is slightly slower than the AVF, but much faster than that of other existing and proposed methods. Its steady-state performance outperforms the more complex MSWF and AVF-based algorithms.

Fig. 5.8 is carried out for the GSC structure under the same scenario as in Fig. 5.7. The curves of the considered algorithms for the GSC show nearly the same convergence and steady-state performance as those for the DFP. It implies that the GSC structure is an alternative way for the CCM beamformer design. The difference is that the GSC processor incorporates the constraint in the structure and thus converts the constrained optimization problem into an unconstrained one. The adaptive methods of the GSC beamformer design are different from those of the DFP but the performance is similar. The following

simulations are carried out for the DFP structure to simplify the presentation.

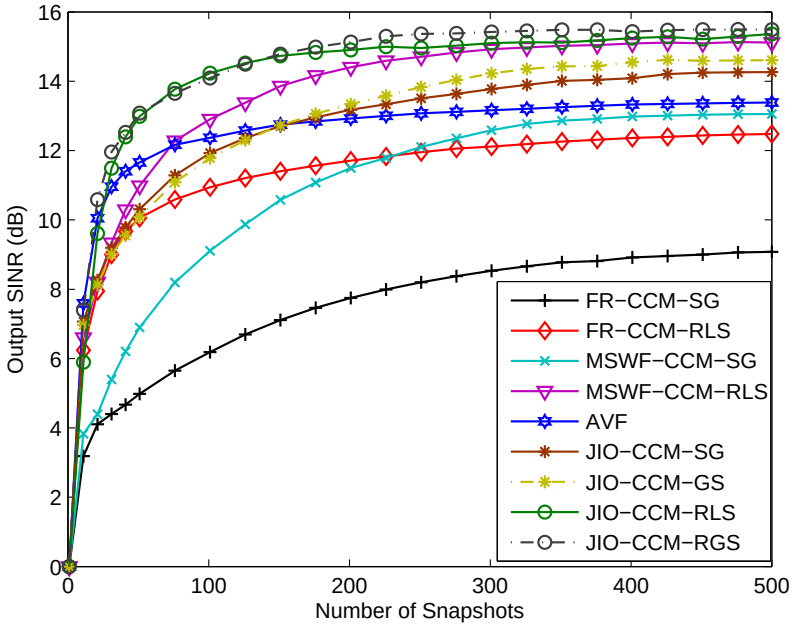

Fig. 5.7: Output SINR versus the number of snapshots with $m = 32$, $q = 7$, SNR$= 10$ dB, $\mu_{T_r} = 0.003$, $\mu_{\bar{w}} = 0.003$, $r = 5$ for SG, $\mu_{T_r} = 0.0023$, $\mu_{\bar{w}} = 0.003$, $r = 5$ for GS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.025$, $r = 5$ for RLS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.02$, $r = 5$ for RGS of the DFP structure.

In the next two experiments, we assess the output SINR performance of the proposed and analyzed algorithms versus their associated rank r, and check the effectiveness of the automatic rank selection technique. The experiment in Fig. 5.9 is intended for setting the adequate rank r of the reduced-rank schemes for a given input SNR and number of snapshots. The scenario is the same as that in Fig. 5.7 except that the number of snapshots is fixed to be $N = 500$ and the rank r is varied between 1 and 16. The result indicates that the best performance of the proposed SG and RLS type algorithms is obtained with rank $r = 5$ for the proposed reduced-rank scheme. The performance of the full-rank methods is invariant with the change of the rank r. For the MSWF technique, its SG and RLS type algorithms achieve their best performance with ranks $r = 6$ and $r = 7$, respectively. For the AVF-based algorithm, the best rank is found to be $r = 7$. It is interesting to note that the best r is usually much smaller than the number of elements m, which leads to significant computational savings. For the proposed and analyzed algorithms, the range of r that has the best performance is concentrated between $r_{\min} = 3$ and $r_{\max} = 7$. This range is used in the next experiment to check the performance of the proposed algorithms with the automatic rank selection technique.

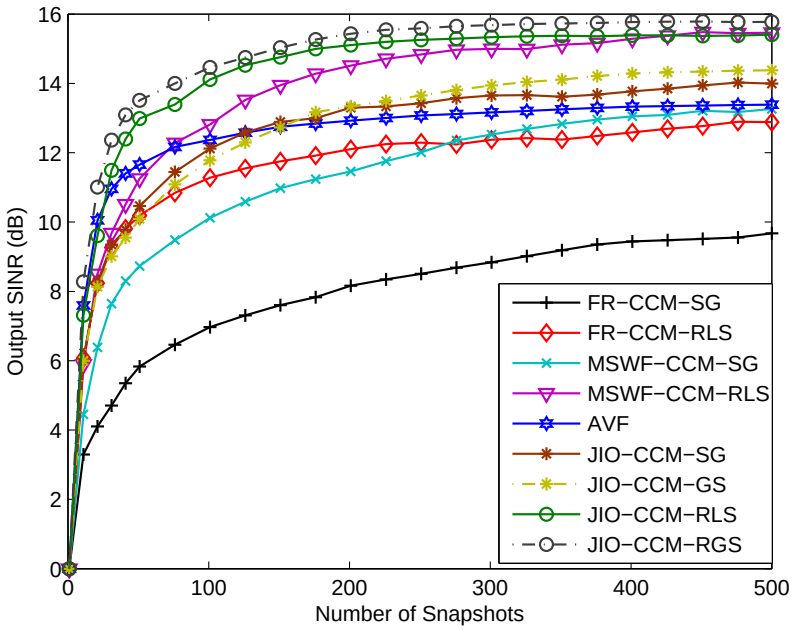

Fig. 5.8: Output SINR versus input SNR with $m = 32$, $q = 7$, SNR$= 10$ dB, $\mu_{T_r} = 0.0025$, $\mu_{\bar{w}_{\rm gsc}} = 0.002$, $r = 5$ for SG, $\mu_{T_r} = 0.003$, $\mu_{\bar{w}_{\rm gsc}} = 0.003$, $r = 5$ for GS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.01$, $r = 5$ for RLS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.0093$, $r = 5$ for RGS of the GSC structure.

Since the performance of the proposed reduced-rank algorithms was found in our studies to be a function of the rank r and other parameters such as the step size and the forgetting factor, we need to consider their impacts on the performance of the system. Specifically, we assume that the step size of the SG type algorithms and the forgetting factor of the RLS type algorithms are adequately chosen and we focus on the developed automatic rank selection technique introduced in the previous section.

In Fig. 5.10, the proposed reduced-rank algorithms utilize fixed values for their rank and also the automatic rank selection technique. We consider the presence of $q = 10$ users (one desired) in the system. The input SNR is 10 dB. The results show that with a lower rank $r = 3$ the reduced-rank algorithms usually converge faster but achieve lower steady-state SINR values. Conversely, with a higher rank $r = 7$ the proposed algorithms converge relatively slower than with a lower rank but reach higher steady-state SINR values. The developed automatic rank selection technique allows the proposed algorithms to circumvent the tradeoff between convergence and steady-state performance for a given rank, by adaptively choosing the best rank for a given number of snapshots which provides both fast convergence and improved steady-state performance.

In the last experiment, we evaluate the performance of the proposed and analyzed al-

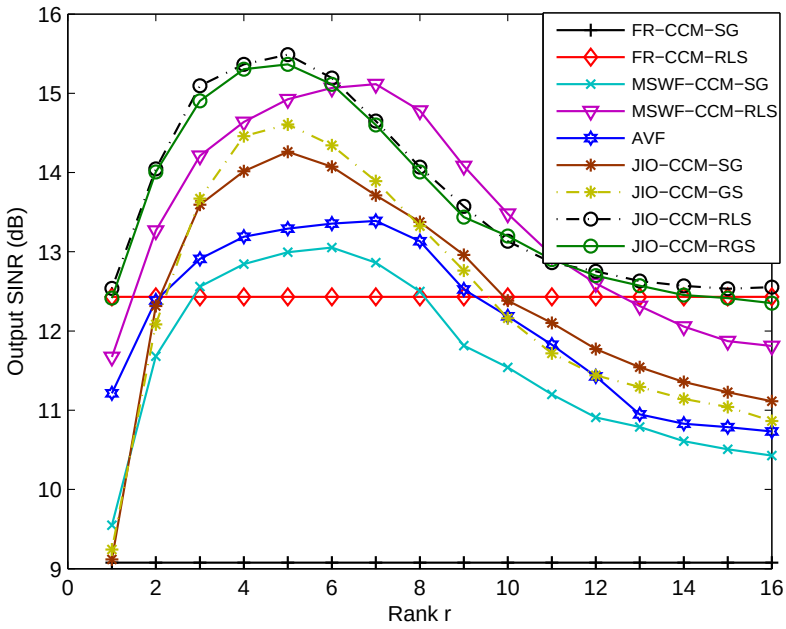

Fig. 5.9: Output SINR versus rank r with $m = 32$, $q = 7$, SNR$= 10$ dB.

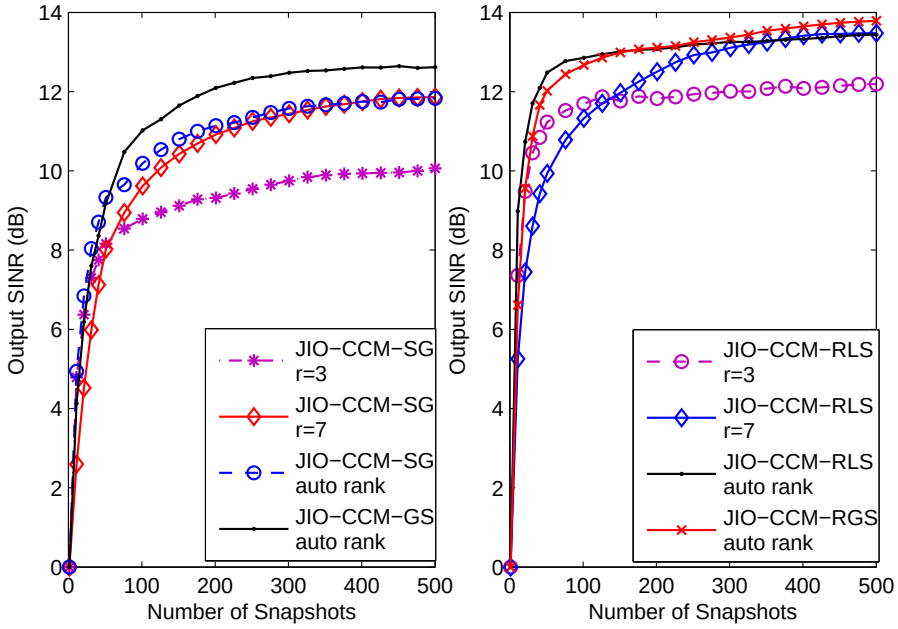

Fig. 5.10: Output SINR versus the number of snapshots with $m = 32$, $q = 10$, SNR$= 10$ dB, (a) $\mu_{T_r} = 0.003$, $\mu_{\bar{w}} = 0.004$ for SG, $\mu_{T_r} = 0.003$, $\mu_{\bar{w}} = 0.001$ for GS; (b) $\alpha = 0.998$, $\delta = \bar{\delta} = 0.03$ for RLS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.026$, $r = 5$ for RGS with the automatic rank selection technique.

gorithms in a non-stationary scenario, namely, when the number of users changes. The automatic rank selection technique is employed, and the step size and the forgetting factor are set to ensure that the considered algorithms converge quickly to the steady-state. The input SNR is 10 dB. In this experiment, the scenario starts with $q = 8$ users including one desired user. From the first stage (first 500 snapshots) of Fig. 5.11, the convergence and steady-state performance of the proposed SG type algorithms is superior to other SG type methods with the full-rank, MSWF and AVF. The proposed RLS type algorithm has a convergence rate a little slower than the AVF but faster than the other analyzed methods, and the steady-state performance better than the existing ones. Three more interferers enter the system at time instant $i = 500$. This change makes the output SINR reduce suddenly and degrades the performance of all methods. Note that the output SINR values of all the methods at $N = 500$ are set around 0 dB since it is convenient to show the convergence behaviors. The proposed SG and RLS type algorithms keep faster convergence and better steady-state performance in comparison with the corresponding SG and RLS type methods based on the full-rank and MSWF techniques. The convergence of the AVF method is fast but the steady-state performance is inferior to the proposed methods.

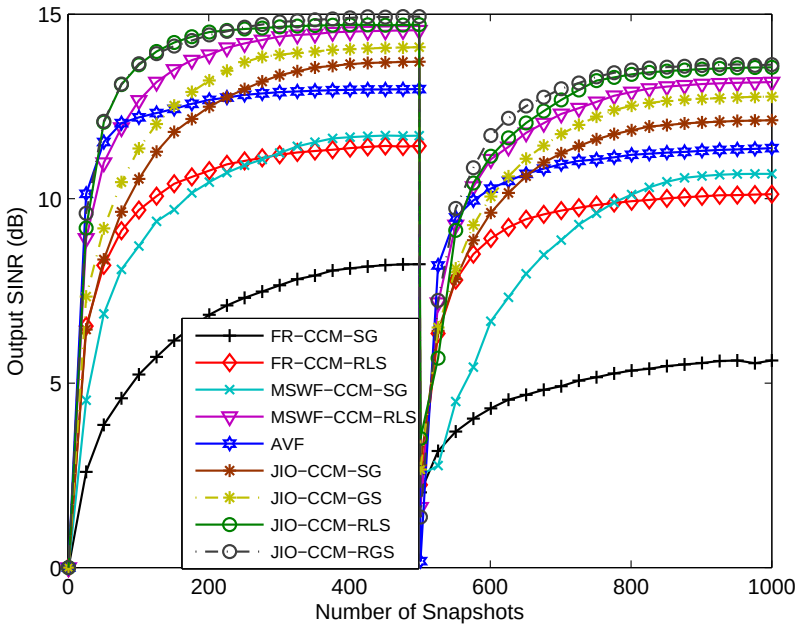

Fig. 5.11: Output SINR versus input SNR with $m = 32$, $q_1 = 8$, $q_2 = 11$, SNR$= 10$ dB, $\mu_{T_r} = 0.003$, $\mu_{\bar{w}} = 0.0038$, $r = 5$ for SG, $\mu_{T_r} = 0.003$, $\mu_{\bar{w}} = 0.001$, $r = 5$ for GS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.033$, $r = 5$ for RLS, $\alpha = 0.998$, $\delta = \bar{\delta} = 0.028$, $r = 5$ for RGS of the proposed CCM reduced-rank scheme.

5.8 Proposed CCM-AVF Algorithm

In this section, we will introduce another CCM reduced-rank adaptive algorithm based on the AVF scheme for beamforming.

5.8.1 Proposed CCM-AVF Scheme

We define the cost function for the beamformer design, which is

$$J_{\text{av}}\big(\boldsymbol{w}(i)\big) = \mathbb{E}\big\{\big|\boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i) - \nu\big|^2\big\}, \tag{5.59}$$

where $\tilde{\boldsymbol{x}}(i) = y^*(i)\boldsymbol{x}(i)$ can be viewed as a new received vector to the beamformer and $\nu = 1$ is set.

To obtain the weight solution for the time index i, we start the iteration by initializing the weight vector

$$\boldsymbol{w}_0(i) = \boldsymbol{a}(\theta_0)/\|\boldsymbol{a}(\theta_0)\|^2. \tag{5.60}$$

Then, we subtract a scaling auxiliary vector (unconstrained component) that is orthogonal to $\boldsymbol{a}(\theta_0)$ from $\boldsymbol{w}_0(i)$ (constrained component) and obtain

$$\boldsymbol{w}_1(i) = \boldsymbol{w}_0(i) - \mu_1(i)\boldsymbol{g}_1(i), \tag{5.61}$$

where $\boldsymbol{g}_1(i) \in \mathbb{C}^{m\times 1}$ with $\boldsymbol{g}_1^H(i)\boldsymbol{a}(\theta_0) = 0$, and $\mu_1(i)$ is a scalar factor to control the weight of $\boldsymbol{g}_1(i)$. The auxiliary vector is supposed to capture the signal components in $\tilde{\boldsymbol{x}}(i)$ that are not from the direction θ_0. The aim of (5.61) is to suppress the disturbance of the unconstrained component while maintaining the contribution of the SOI. The cost function in (5.59) appears in unconstrained form since the constraint has been incorporated in the weight adaptation.

5.8.2 Proposed CCM-AVF Algorithm

From (5.61), it is necessary to determine the auxiliary vector $\boldsymbol{g}_1(i)$ and the scalar factor $\mu_1(i)$ for the calculation of $\boldsymbol{w}_1(i)$. Assuming $\boldsymbol{g}_1(i)$ is known, $\mu_1(i)$ can be obtained by minimizing $\mathbb{E}\{[\boldsymbol{w}_1(i)\tilde{\boldsymbol{x}}(i) - 1]^2\}$. Substituting (5.61) into this cost function, computing

the gradient with respect to $\mu_1(i)$ and equating it to zero, we obtain

$$\mu_1(i) = \frac{\boldsymbol{g}_1^H(i)\tilde{\boldsymbol{R}}(i)\boldsymbol{w}_0(i) - \boldsymbol{g}_1^H(i)\tilde{\boldsymbol{p}}(i)}{\boldsymbol{g}_1^H(i)\tilde{\boldsymbol{R}}(i)\boldsymbol{g}_1(i)}, \tag{5.62}$$

where $\tilde{\boldsymbol{R}}(i) = \mathbb{E}[\tilde{\boldsymbol{x}}(i)\tilde{\boldsymbol{x}}^H(i)] \in \mathbb{C}^{m\times m}$ and $\tilde{\boldsymbol{p}}(i) = \mathbb{E}[\tilde{\boldsymbol{x}}(i)] \in \mathbb{C}^{m\times 1}$. Note that the situation that $\mu_1(i) = 0$, i.e., $\tilde{\boldsymbol{R}}(i)\boldsymbol{w}_0(i) = \tilde{\boldsymbol{p}}(i)$, needs to be avoided here.

On the other hand, the calculation of the auxiliary vector $\boldsymbol{g}_1(i)$ should take the conditions $\boldsymbol{g}_1^H(i)\boldsymbol{a}(\theta_0) = 0$ and $\boldsymbol{g}_1^H(i)\boldsymbol{g}_1(i) = 1$ into account. The constrained minimization problem with respect to $\boldsymbol{g}_1(i)$ can be transformed by the method of Lagrange multipliers into

$$J_{\mathrm{L}}\big(\boldsymbol{w}_1(i)\big) = \mathbb{E}\big\{\big[\boldsymbol{w}_1^H(i)\tilde{\boldsymbol{x}}(i) - 1\big]^2\big\} - 2\,\Re\big\{\lambda_1\big[\boldsymbol{g}_1^H(i)\boldsymbol{g}_1(i) - 1\big] - \lambda_2\boldsymbol{g}_1^H(i)\boldsymbol{a}(\theta_0)\big\}, \tag{5.63}$$

where λ_1 and λ_2 are scalar Lagrange multipliers. For the sake of mathematical accuracy, we note that the cost function to be minimized is phase invariant, namely, if $\boldsymbol{g}_1(i)$ is the solution of the minimization problem, so does $\boldsymbol{g}_1(i)e^{j\phi}$ for any phase ϕ. To avoid any ambiguity, we assume that only one auxiliary vector can be obtained.

Following the procedure to get $\mu_1(i)$, the auxiliary vector can be expressed by

$$\boldsymbol{g}_1(i) = \frac{\mu_1^*(i)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) - \lambda_2\boldsymbol{a}(\theta_0)}{\lambda_1}, \tag{5.64}$$

where $\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) = \mathbb{E}\big[\big(1 - \tilde{y}(i)\big)^*\tilde{\boldsymbol{x}}(i)\big] \in \mathbb{C}^{m\times 1}$ and $\tilde{y}(i) = \boldsymbol{w}^H(i)\tilde{\boldsymbol{x}}(i)$. We keep the time index i in $\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)$ since it is a function of $\boldsymbol{w}(i)$, which must be initialized to provide an estimation of $\tilde{y}(i)$ and to start the iteration.

The expression of $\boldsymbol{g}_1(i)$ is utilized to enforce the constraints and solve for λ_1 and λ_2. Indeed, we have

$$\lambda_1 = \left\| \mu_1^*(i)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) - \frac{\mu_1^*(i)\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)}{\|\boldsymbol{a}(\theta_0)\|^2}\boldsymbol{a}(\theta_0) \right\|, \tag{5.65}$$

$$\lambda_2 = \frac{\mu_1^*(i)\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)}{\|\boldsymbol{a}(\theta_0)\|^2}, \tag{5.66}$$

where $\|\cdot\|$ denotes the Euclidean norm. Substitution of λ_1 and λ_2 back in (5.64) leads to $\boldsymbol{g}_1(i)$ that satisfies the constraints and minimizes (with $\mu_1(i)$) the squared deviation of

$\tilde{y}(i)$ from the CM condition, yielding

$$\boldsymbol{g}_1(i) = \frac{\mu_1^*(i)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) - \frac{\mu_1^*(i)\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)}{\|\boldsymbol{a}(\theta_0)\|^2}\boldsymbol{a}(\theta_0)}{\left\|\mu_1^*(i)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) - \frac{\mu_1^*(i)\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)}{\|\boldsymbol{a}(\theta_0)\|^2}\boldsymbol{a}(\theta_0)\right\|}. \tag{5.67}$$

So far, we have detailed the first iteration of the proposed CCM-AVF algorithm for time index i, i.e., $\boldsymbol{w}_0(i)$ in (5.60), $\boldsymbol{w}_1(i)$ in (5.61), $\mu_1(i)$ in (5.62), and $\boldsymbol{g}_1(i)$ in (5.67), respectively. In this procedure, $\tilde{\boldsymbol{x}}(i)$ can be viewed as a new received vector that is processed by the adaptive filter $\boldsymbol{w}_1(i)$ $\big($first estimation of $\boldsymbol{w}(i)\big)$ to generate the output $\tilde{y}(i)$, in which, $\boldsymbol{w}_1(i)$ is determined by minimizing the mean squared error between the output and the desired CM condition. This principle is suitable to the following iterations with $\boldsymbol{w}_2(i), \boldsymbol{w}_3(i), \ldots$.

Now, we consider the iterations one step further and express the adaptive filter as

$$\boldsymbol{w}_2(i) = \boldsymbol{w}_0(i) - \sum_{k=1}^{2}\mu_k(i)\boldsymbol{g}_k(i) = \boldsymbol{w}_1(i) - \mu_2(i)\boldsymbol{g}_2(i), \tag{5.68}$$

where $\mu_2(i)$ and $\boldsymbol{g}_2(i)$ will be calculated based on the previously identified $\mu_1(i)$ and $\boldsymbol{g}_1(i)$. $\mu_2(i)$ $\big(\mu_2(i) \neq 0\big)$ is chosen to minimize the cost function $\mathbb{E}\{[\boldsymbol{w}_2^H(i)\tilde{\boldsymbol{x}}(i) - 1]^2\}$ under the assumption that $\boldsymbol{g}_2(i)$ is known beforehand. Thus, we have

$$\mu_2(i) = \frac{\boldsymbol{g}_2^H(i)\tilde{\boldsymbol{R}}(i)\boldsymbol{w}_1(i) - \boldsymbol{g}_2^H(i)\tilde{\boldsymbol{p}}(i)}{\boldsymbol{g}_2^H(i)\tilde{\boldsymbol{R}}(i)\boldsymbol{g}_2(i)}, \tag{5.69}$$

The auxiliary vector $\boldsymbol{g}_2(i)$ is calculated by the minimization of the cost function subject to the constraints $\boldsymbol{g}_2^H(i)\boldsymbol{a}(\theta_0) = 0$ and
$\boldsymbol{g}_2^H(i)\boldsymbol{g}_2(i) = 1$, which is

$$\boldsymbol{g}_2(i) = \frac{\mu_2^*(i)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) - \frac{\mu_2^*(i)\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)}{\|\boldsymbol{a}(\theta_0)\|^2}\boldsymbol{a}(\theta_0)}{\left\|\mu_2^*(i)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) - \frac{\mu_2^*(i)\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)}{\|\boldsymbol{a}(\theta_0)\|^2}\boldsymbol{a}(\theta_0)\right\|}. \tag{5.70}$$

The above iterative procedures are taken place at time index i to generate a sequence of filters $\boldsymbol{w}_k(i)$ with $k = 0, 1, \ldots$ being the iteration number. Generally, there exists a maximum (or suitable) value of k, i.e., $k_{\max} = K$, that is determined by a certain rule to stop iterations and achieve satisfactory performance. One simple rule, which is adopted in the proposed CCM-AVF algorithm, is to terminate the iteration if $\boldsymbol{g}_k(i) \cong \boldsymbol{0}$ is achieved. Alternative and more complicated selection rules can be found in [92]. Until now, the weight solution at time index i can be given by $\boldsymbol{w}(i) = \boldsymbol{w}_K(i)$. The proposed CCM-AVF algorithm for the design of the CCM beamformer is summarized in Table 5.7.

Tab. 5.7: Proposed CCM-AVF algorithm.

For the time index $i = 1, 2, \ldots, N$**.**
Initialization:
$\boldsymbol{w}(i) = \boldsymbol{w}_0(i) = \frac{\boldsymbol{a}(\theta_0)}{\|\boldsymbol{a}(\theta_0)\|^2}$; $\mu_0(i) =$ small positive value.
Iterative procedures:
For $k = 1, 2, \ldots, K$
$\boldsymbol{g}_k(i) = \mu_{k-1}^*(i)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i) - \frac{\mu_{k-1}^*(i)\boldsymbol{a}^H(\theta_0)\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)}{\|\boldsymbol{a}(\theta_0)\|^2}\boldsymbol{a}(\theta_0)$
if $\boldsymbol{g}_k(i) = \mathbf{0}$ **then EXIT.**
$\mu_k(i) = \frac{\boldsymbol{g}_k^H(i)\tilde{\boldsymbol{R}}(i)\boldsymbol{w}_{k-1}(i) - \boldsymbol{g}_k^H(i)\tilde{\boldsymbol{p}}(i)}{\boldsymbol{g}_k^H(i)\tilde{\boldsymbol{R}}(i)\boldsymbol{g}_k(i)}$
$\boldsymbol{w}_k(i) = \boldsymbol{w}_{k-1}(i) - \mu_k\boldsymbol{g}_k(i)$
Weight expression:
$\boldsymbol{w}(i) = \boldsymbol{w}_K(i)$.

5.8.3 Interpretations about Proposed CCM-AVF Algorithm

There are several points we need to interpret in Table 5.7. First of all, initialization is important to the realization of the proposed method. $\boldsymbol{w}(i)$ is set to estimate $\tilde{y}(i)$ and so $\tilde{\boldsymbol{R}}(i)$, $\tilde{\boldsymbol{p}}(i)$, and $\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)$. $\boldsymbol{w}_0(i)$ is for the activation of the weight adaptation. Note that the calculation of the scalar factor, e.g., in (5.62), is a function of $\boldsymbol{g}_1(i)$ and the auxiliary vector obtained from (5.67) depends on $\mu_1(i)$. It is necessary to initialize one of these quantities to start the iteration. We usually set a small positive scalar value $\mu_0(i)$ for simplicity. Under this condition, the subscript of the scalar factor for the calculation of $\boldsymbol{g}_k(i)$ should be replaced by $k-1$ instead of k, as shown in Table 5.7.

Second, the expected quantities $\tilde{\boldsymbol{R}}(i)$, $\tilde{\boldsymbol{p}}(i)$, and $\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)$ are not available in practice. We use a sample-average approach to estimate them, i.e.,

$$\hat{\tilde{\boldsymbol{R}}}(i) = \frac{1}{i}\sum_{l=1}^{i}\tilde{\boldsymbol{x}}(l)\tilde{\boldsymbol{x}}^H(l); \quad \hat{\tilde{\boldsymbol{p}}}(i) = \frac{1}{i}\sum_{l=1}^{i}\tilde{\boldsymbol{x}}(l); \quad \hat{\tilde{\boldsymbol{p}}}_{\mathrm{y}}(i) = \frac{1}{i}\sum_{l=1}^{i}\big(1-\tilde{y}(l)\big)^*\tilde{\boldsymbol{x}}(i). \tag{5.71}$$

where $\tilde{\boldsymbol{R}}(i)$, $\tilde{\boldsymbol{p}}(i)$, and $\tilde{\boldsymbol{p}}_{\mathrm{y}}(i)$ are substituted by their estimates in the iterative procedure to generate $\boldsymbol{w}_k(i)$. To improve the estimation accuracy, the quantities in (5.71) can be refreshed or further regularized during the iterations. Specifically, we use $\boldsymbol{w}_k(i)$ in the iteration step instead of $\boldsymbol{w}(i)$ in the initialization to generate $y(i)$, and related $\tilde{\boldsymbol{x}}(i)$ and $\tilde{y}(i)$, which are employed to update the estimates $\hat{\tilde{\boldsymbol{R}}}(i)$, $\hat{\tilde{\boldsymbol{p}}}(i)$, and $\hat{\tilde{\boldsymbol{p}}}_{\mathrm{y}}(i)$. Compared with $\boldsymbol{w}(i) = \boldsymbol{a}(\theta_0)/\|\boldsymbol{a}(\theta_0)\|^2$, $\boldsymbol{w}_k(i)$ is more efficient to evaluate the desired signal. Thus, the refreshment of the estimates based on the current $\boldsymbol{w}_k(i)$ is valuable to calculate the subsequent scalar factor and the auxiliary vector.

Third, we drop the normalization of the auxiliary vector [31, 82, 92]. Note that the calculated auxiliary vectors $\boldsymbol{g}_k(i)$ are constrained to be orthogonal to $\boldsymbol{a}(\theta_0)$. The orthogonality among the auxiliary vectors is not imposed. Actually, the successive auxiliary vectors do satisfy the orthogonality as verified in our numerical results. We leave the analysis about this characteristic for the future work.

The proposed CCM-AVF beamformer efficiently measures the expected deviation of the beamformer output from the CM value and provide useful information for the proposed algorithm for dealing with parameter estimation in many severe scenarios including low signal-to-noise ratio (SNR) or steering vector mismatch. The proposed CCM-AVF algorithm employs an iterative procedure to adjust the weight vector for each time instant. The matrix inversion appeared in (5.1) is avoided and thus the computational cost is limited. Since the scalar factor and the auxiliary vector depend on each other, the proposed algorithm provides an iterative exchange of information between them, which are jointly employed to update the weight vector. This scheme leads to an improved convergence and the steady-state performance that will be shown in the simulations.

5.9 Simulations

Simulations are performed for a ULA containing $m = 40$ sensor elements with half-wavelength interelement spacing. We compare the proposed algorithm (CCM-AVF) with the SG [18], RLS [93, 94], MSWF [28], and AVF [31] methods. For each method, we consider the CMV and the CCM criteria for beamforming. A total of 1000 runs are used to get the curves. In all experiments, BPSK sources' powers (desired user and interferers) are $\sigma_{\rm S}^2 = \sigma_{\rm I}^2 = 1$ and the input SNR$= 0$ dB with spatially and temporally white Gaussian noise.

Fig. 5.12 includes two experiments. There are $q = 10$ users, including one desired user in the system. We set $\mu_0(i) = 0.01$. In Fig.5.12 (a), the exact DOA of the SOI is known at the receiver. All output SINR values increase to the steady-state following the snapshots. The RLS-type algorithms enjoy faster convergence and better steady-state performance than the SG-type methods. The proposed CCM-AVF algorithm converges rapidly and reaches the steady-state with superior performance. The CCM-based MSWF technique with the RLS implementation has comparative fast convergence rate but the steady-state performance is not as good as that of the proposed method. In Fig. 5.12 (b), we set the DOA of the SOI estimated by the receiver to be 1^o away from the actual direction. It indicates that the mismatch induces performance degradation to all the analyzed algorithms.

The CCM-based methods are more robust to this scenario than the CMV-based ones since the CCM deviation between the beamformer output and the constant modulus provides more information for the parameter estimation. The proposed CCM-AVF algorithm has faster convergence and better steady-state performance than the other analyzed methods.

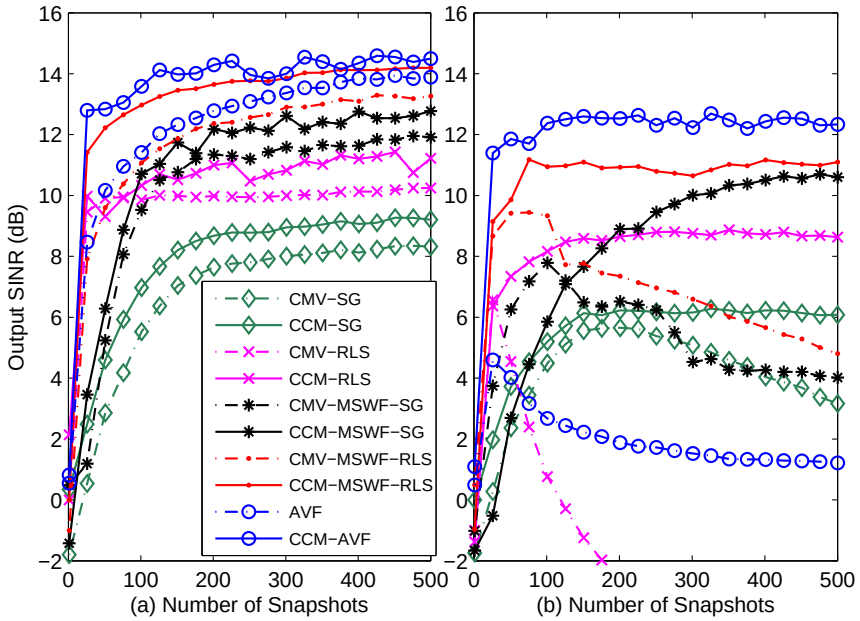

Fig. 5.12: Output SINR versus the number of snapshots for (a) ideal steering vector; (b) steering vector mismatch 1^o.

In Fig. 5.13, we keep the same scenario as that in Fig. 5.12 (a) and check the iteration number for the existing and proposed methods. The number of snapshots is fixed to $N = 500$. The most adequate iteration number for the proposed CCM-AVF algorithm is $K = 3$, which is less than those of other analyzed algorithms, but reach the preferable performance. We also checked that this value is rather insensitive to the number of users in the system, to the number of sensor elements, and work efficiently for the studied scenarios.

5.10 Conclusions

In this chapter, we proposed reduced-rank adaptive algorithms according to the CCM criterion for beamforming. The proposed algorithms can be divided into the JIO scheme based and the AVF scheme based methods. In the JIO scheme, the dimension of the received vector is reduced by the adaptive transformation matrix that is formed by a bank of full-rank adaptive filters, and the transformed received vector is processed by the reduced-

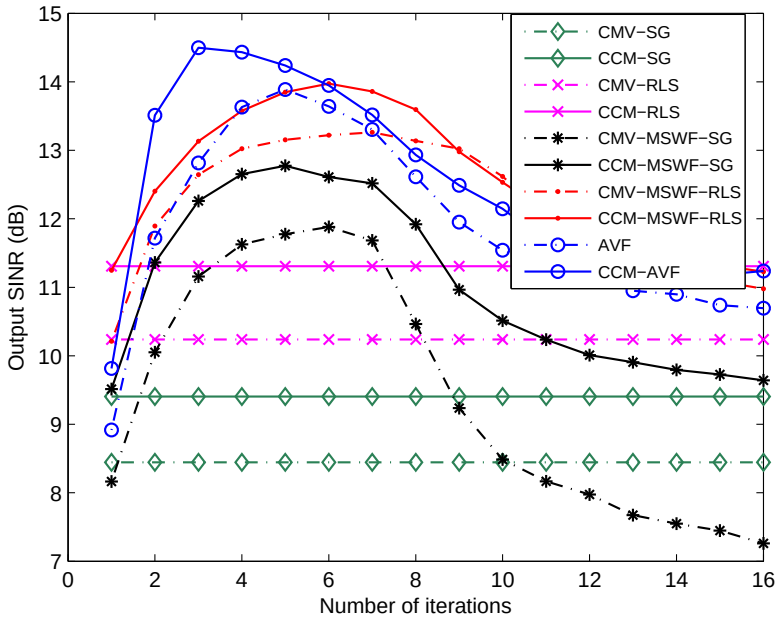

Fig. 5.13: Output SINR versus the number of iterations.

rank adaptive filter for estimating the desired signal. The JIO scheme was developed for both DFP and GSC structures. We derived the CCM expressions for the transformation matrix and the reduced-rank weight vector, and developed SG and RLS type algorithms for their calculations. The complexity and convexity analysis of the proposed algorithms was carried out. In the AVF scheme, the weight solution is iterated by jointly calculating the auxiliary vector and the scalar factor. The auxiliary vector and the scalar factor exchange information between each other and lead to a good performance.

6. CONCLUSIONS AND FUTURE WORK

Contents

6.1 Summary of Work

In this thesis, we have investigated array processing algorithms and their applications to beamforming and DOA estimation. According to the dimension processed, these algorithms can be categorized into full-rank and reduced-rank. For the full-rank techniques, we employed two adaptive step size mechanisms to develop the SG algorithms (Chapter 2) and derived constrained CG-based adaptive algorithms (Chapter 3) for the beamformer design. For the reduced-rank techniques, we proposed the JIO reduced-rank scheme for both beamforming (Chapter 4, 5) and DOA estimation (Chapter 4) and presented SG and RLS-based algorithms. The proposed array processing algorithms were derived according to the CMV and/or the CCM design criteria.

In Chapter 2, we started to introduce the adaptive SG algorithm with the CCM criterion for beamforming and pointed out the drawback of the performance degradation due to the inappropriate selection of the step size. Then, we presented two adaptive step size mechanisms according to the CCM criterion to adjust the step size. We compared the complexity of the developed mechanisms with the existing fixed and adaptive step size SG methods. Furthermore, we investigated the characteristics of the proposed SG algorithms. First, we studied the sufficient condition for the convergence of the mean weight vector. Then, the steady-state mean and mean-square expressions of the step size were derived. On the basis of these step size expressions, we employed the energy conservation relation to give the steady-state and tracking analyses of the proposed methods.

In Chapter 3, we developed two CG adaptive algorithms according to the CMV and CCM criteria. The existing CG algorithms incorporate the constraints with significant

high computational cost to solve the system of equations. The proposed algorithms were motivated to circumvent this problem. We introduced a CG-based weight vector to incorporate the constraint of the array response in the proposed methods for beamforming. A simple relation between the CG-based weight vector and the matrix inversion and the array response of the SOI was established. The weight solution can be obtained by iteratively computing the CG-based weight vector. We derived the convexity condition for the global convergence of the CCM criterion and compared the complexity of the proposed algorithms with the existing SG, RLS, MSWF, and AVF methods. The convergence properties of the proposed methods were also considered here.

In Chapters 4, we introduced a robust JIO reduced-rank scheme according to the CMV criterion and developed adaptive algorithms. The JIO reduced-rank scheme was motivated to deal with the beamformer design with large arrays. The essence of the JIO scheme is to change the role of adaptive filters. It includes a bank of adaptive filters that constitute a transformation matrix and a reduced-rank filter with the reduced-rank weight vector. The transformation matrix maps the received vector into a lower dimension subspace to obtain the reduced-rank received vector, which is processed by the reduced-rank weight vector to compute the output. The transformation matrix and the reduced-rank weight vector are joint optimized according to the design criterion. The reduced-rank scheme provided a iterative exchange of information between the transformation matrix and the reduced-rank filter for weight adaptation and thus leads to an improved convergence and tracking performance. We also presented a DOA estimation algorithm based on the JIO scheme. We derived adaptive RLS algorithms to iteratively update the transformation matrix and the reduced-rank weight vector according to the minimum variance criterion for plotting the output power spectrum over the possible scanning angles. The objective of the proposed algorithms was to resolve the DOA estimation problem with large arrays and a small number of snapshots. The forward and backward SS technique was employed in the LS type algorithm to deal with the highly correlated sources' problem. These proposed methods exhibit a dominance when many sources are present.

Chapter 5 was devoted to the CCM reduced-rank schemes and algorithms in the application of the beamformer design. We introduced the CCM-based JIO reduced-rank scheme. This scheme was investigated in both DFP and GSC structures. For each structure, a family of computationally efficient reduced-rank SG and RLS type algorithms were derived. The GS technique was employed in the proposed methods to reformulate the transformation matrix for further improving the performance. An automatic rank selection technique based on the constant modulus criterion was developed there. The complexity comparison between the existing and proposed algorithms was given and the convergence properties for the CCM reduced-rank scheme was analyzed. Besides, we

proposed a CCM-based AVF algorithm for robust adaptive beamforming in this chapter. The proposed beamformer decomposes the adaptive filter into a constrained (reference vector filters) and an unconstrained (auxiliary vector filters) components. The weight vector is iteratively computed by subtracting the scaling auxiliary vector from the reference vector. The scalar factor and the auxiliary vector are jointly calculated according to the CCM criterion. The proposed algorithm shows the robustness in severe conditions.

6.2 Future Work

For the future work, we plan to pay our efforts to the development of array processing algorithms for beamforming and DOA estimation. Some suggestions for future work based on this thesis are given below.

In Chapter 4 and Chapter 5, we introduced the JIO reduced-rank scheme with the CMV and CCM criteria for the beamformer design and developed SG and RLS adaptive algorithms. For the SG algorithm, it is possible to employ the adaptive step size mechanisms reported in Chapter 2 to adjust the step size. One possibility to further develop the reduced-rank scheme is to utilize the combination of the adaptive filters [104]. This scheme tries to use a smart way to combine two adaptive filters $\boldsymbol{w}_1(i)$ and $\boldsymbol{w}_2(i)$. The key to the scheme is the selection of a scalar mixing parameter $\eta(i)$ for combining the two filter outputs. Several approaches have been reported to adapt $\eta(i)$ [104]- [106]. A combination of adaptive filters can lead to fast convergence and good steady-state performance. Current works on the combination focus on the full-rank adaptive filters, like LMS, NLMS, or RLS. It brings us an idea to use this combination for the reduced-rank processing, namely, two JIO reduced-rank branches can work together by using the convex combination [105] or affine combination [106]. In each reduced-rank branch, a transformation matrix and a reduced-rank adaptive filter operate to compute the output. Another point we could consider in this topic is the stability of the combination. The combination of two adaptive filters of same type is relatively simple. However, if two filters does not belong to the same type, how can we efficiently adapt the scalar factor for combination? This is a choke point of current works, which is also an interesting topic we can examine in the future.

Regarding another kind of reduced-rank technique, i.e., the CCM-AVF proposed in Chapter 5, we want to investigate more about its properties. In Chapter 5, we developed the adaptive algorithm and checked the impact of the selection of the number of iteration K to the output performance. Actually, K was selected based on the simulation experi-

ence. The proposed algorithm would be more powerful if it can be selected automatically according to the design criterion. Besides, the convergence analysis will be an important point to keep the integrity of the proposed method and establish formal conditions of convergence. Different from the convergence analysis presented in [31], our analysis should consider the impact of the covariance matrix, which does not purely depend on the received vector but is a function of the weight vector.

In the development of the full-rank and reduced-rank adaptive algorithms, we always assumed that the locations of the users are fixed. This assumption is weak in practice. Although the DOA estimation algorithm can be employed to determine the locations of the users, it always causes the time delay and extra computational load. Besides, we considered the AWGN channel in simulations. For the application of the proposed algorithms in wireless communications, it is necessary to consider the impacts of the time varying channel. These topics are what we should investigate in the future.

For the DOA estimation, polynomial rooting can be considered in the proposed reduced-rank scheme. By using this technique, we could roughly locate the directions of the sources and thus reduce the search range. The proposed DOA estimation algorithm would be operated under this limited scanning angles to plot the output power spectrum and determine the final DOAs. This technique is an efficient way to avoid the exhaustive search through all possible angles to estimate the DOAs. A challenge to utilize polynomial rooting in the proposed scheme is that the calculation of the output power with respect to each scanning angle is a function of both the transformation matrix and the reduced-rank weight vector, as can be seen in (4.49). It is difficult to transfer this expression into an alternative one with the summation of only the DOAs and the received vector, as the polynomial rooting for the Capon's method. We plan to use the property of the pseudo-inverse to simplify the expression and make an equivalence between the proposed output power equation and the the Capon's one, and then employ the polynomial rooting technique.

Another topic motivated from the DOA estimation is to use subspace tracking algorithms to estimate the signal subspace, avoiding the eigen-decomposition. Actually, the estimation of the signal subspace can be viewed as a constrained or unconstrained optimization problem, for which the introduction of a projection approximation hypothesis leads to fast subspace tracking methods, e.g., PAST [97], NIC [98], or API [100]. By using different optimization problems and realizations of the projection matrix, it is possible to reach a trade-off between the complexity and estimation accuracy of the signal subspace. How to efficiently estimate this signal subspace is what we plan to investigate in the future.

APPENDIX

A. DERIVATION OF (2.28)

In the steady-state environment, applying assumption i) in Chapter 3 to (2.27) and defining $B = \xi_{\text{min}} + \xi_{\text{ex}}(\infty)$, we have

$$\mathbb{E}[v^2(i)] = (1-\beta)^2[B^2 + \beta^2 B^2 + \ldots + \beta^{2(i-1)}B^2]. \tag{A.1}$$

In a compact way, we also define $C = [B^2 + \beta^2 B^2 + \ldots + \beta^{2(i-1)}B^2]$. By multiplying β^2 on both sides, we obtain

$$\beta^2 C = \beta^2 B^2 + \beta^4 B^2 + \ldots + \beta^{2(i-1)}B^2 + \beta^{2i}B^2. \tag{A.2}$$

Since $0 < \beta < 1$, if $i \to \infty$, the term β^{2i} in (A.2) can be ignored and then

$$\beta^2 C = C - B^2. \tag{A.3}$$

Rearranging (A.3) to get C, substituting it into (A.1), and recalling assumption ii), we find

$$\mathbb{E}[v^2(i)] = \frac{(1-\beta)^2 B^2}{1-\beta^2} = \frac{(1-\beta)(\xi_{\text{min}} + \xi_{\text{ex}}(\infty))^2}{1+\beta} \approx \frac{(1-\beta)\xi_{\text{min}}^2}{1+\beta}. \tag{A.4}$$

B. CONVEXITY CONDITION FOR THE CCM CRITERION

We consider the cost function (2.4), which can be written as

$$\begin{aligned} J_{\text{cm}} &= \mathbb{E}[|y(i)|^4 - 2|y(i)|^2 + 1] \\ &= \mathbb{E}[|\boldsymbol{w}^H(i)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{w}(i)|^2] - 2\mathbb{E}[|\boldsymbol{w}^H(i)\boldsymbol{x}(i)|^2] + 1, \end{aligned} \tag{B.1}$$

where $\boldsymbol{x}(i) = \sum_{k=0}^{q-1} B_k d_k \boldsymbol{a}(\theta_k) + \boldsymbol{n}(i)$ with B_k being the signal amplitude, d_k is the transmitted bit of the kth user, and k ($k = 0, \ldots, q-1$) is the user number.

For the sake of analysis, we will follow the assumption in [90] and consider a noise free case. For small noise variance σ_n^2, this assumption can be considered as a small perturbation and the analysis will still be applicable. For large σ_n^2, we remark that the term γ can be adjusted to make (2.4) convex, as pointed out in [90]. Under this assumption, we write the received vector as $\boldsymbol{x}(i) = \boldsymbol{A}\boldsymbol{B}\boldsymbol{d}(i)$, where $\boldsymbol{A}$, as before, denotes the signature matrix, $\boldsymbol{B} = \text{diag}[B_0, \ldots, B_{q-1}]$, and $\boldsymbol{d}(i) = [d_0(i), \ldots, d_{q-1}(i)]^T$. For simplicity, we remove the time instant i in the quantities. Letting $r_k = B_k \boldsymbol{w}^H \boldsymbol{a}(\theta_k)$ and $\boldsymbol{r} = [r_0, \ldots, r_{q-1}]^T$, (B.1) can be written as

$$J_{\text{cm}} = \mathbb{E}[\boldsymbol{r}^H \boldsymbol{d}\boldsymbol{d}^H \boldsymbol{r}\boldsymbol{r}^H \boldsymbol{d}\boldsymbol{d}^H \boldsymbol{r}] - 2\mathbb{E}[\boldsymbol{r}^H \boldsymbol{d}\boldsymbol{d}^H \boldsymbol{r}] + 1. \tag{B.2}$$

Since d_k are independent random variables, the evaluation of the first two terms in the brackets of (B.2) reads

$$\begin{aligned} \boldsymbol{r}^H \boldsymbol{d}\boldsymbol{d}^H \boldsymbol{r}\boldsymbol{r}^H \boldsymbol{d}\boldsymbol{d}^H \boldsymbol{r} &= \sum_{k=0}^{q-1} \sum_{j=0}^{q-1} |d_k|^2 |d_j|^2 r_k^* r_k r_j^* r_j \\ \boldsymbol{r}^H \boldsymbol{d}\boldsymbol{d}^H \boldsymbol{r} &= \sum_{k=0}^{q-1} |d_k|^2 r_k^* r_k. \end{aligned} \tag{B.3}$$

Substituting (B.3) into (B.2) and using the constrained condition $\boldsymbol{w}^H \boldsymbol{a}(\theta_0) = \gamma$, we

have

$$\begin{aligned} J_{\text{ccm}} =& \mathbb{E}\big[|d_0|^2 B_0^2\gamma^2 + \sum_{k=1}^{q-1} |d_k|^2 r_k^* r_k\big]^2 \\ & - 2\mathbb{E}\big[|d_0|^2 B_0^2\gamma^2 + \sum_{k=1}^{q-1} |d_k|^2 r_k^* r_k\big] + 1, \end{aligned} \tag{B.4}$$

where d_0 and B_0 denote the transmitted bit and amplitude relevant to the desired signal, and thus

$$J_{\text{ccm}} = \mathbb{E}\big[|d_0|^2 B_0^2\gamma^2 + \bar{\boldsymbol{r}}^H \bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^H \bar{\boldsymbol{r}}\big]^2 - 2\mathbb{E}\big[|d_0|^2 B_0^2\gamma^2 + \bar{\boldsymbol{r}}^H \bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^H \bar{\boldsymbol{r}}\big] + 1, \tag{B.5}$$

where $\bar{\boldsymbol{d}} = [d_1, \ldots, d_{q-1}]^T$ and $\bar{\boldsymbol{r}} = [r_1, \ldots, r_{q-1}]^T$. To examine the convexity property of (B.5), we compute Hessian $\boldsymbol{H}$ with respect to $\bar{\boldsymbol{r}}^H$ and $\bar{\boldsymbol{r}}$, that is $\boldsymbol{H} = \frac{\partial}{\partial \bar{\boldsymbol{r}}^H}\frac{\partial J_{\text{ccm}}}{\partial \bar{\boldsymbol{r}}}$ yields

$$\boldsymbol{H} = 2\mathbb{E}\big[(|d_0|^2 B_0^2\gamma^2 - 1)\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^H + \bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^H \bar{\boldsymbol{r}}\bar{\boldsymbol{r}}^H \bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^H + \bar{\boldsymbol{r}}^H \bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^H \bar{\boldsymbol{r}}\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^H\big], \tag{B.6}$$

where $\boldsymbol{H}$ should be positive semi-definite to ensure the convexity of the optimization problem. The second and third term in (B.6) yield positive semi-definite matrices, while the first term provides the condition $|d_0|^2 B_0^2\gamma^2 - 1 \geq 0$ to ensure the convexity of J_{ccm}. Since $\bar{\boldsymbol{r}}$ can be expressed as a linear function of $\boldsymbol{w}$, i.e., $\bar{\boldsymbol{r}} = \boldsymbol{C}\boldsymbol{w}$, where $\boldsymbol{B}^{'} = \text{diag}(B_1, \ldots, B_{q-1}) \in \mathbb{R}^{(q-1)\times(q-1)}$, $\boldsymbol{A}^{'} = [\boldsymbol{a}(\theta_1), \ldots, a(\theta_{q-1})] \in \mathbb{C}^{m\times(q-1)}$, and $\boldsymbol{C} = \boldsymbol{B}^{'H}\boldsymbol{A}^{'H} \in \mathbb{C}^{(q-1)\times m}$. This expression shows that $J_{\text{ccm}}(\boldsymbol{w})$ is a convex function of $\boldsymbol{w}$ as $J_{\text{ccm}}(\bar{\boldsymbol{r}}) = J_{\text{ccm}}(\boldsymbol{C}\boldsymbol{w})$ is of $\bar{\boldsymbol{r}}$ when

$$\gamma^2 \geq \frac{1}{|d_0|^2 B_0^2}. \tag{B.7}$$

The optimization problem is convex if the condition in (B.7) is satisfied. Note that this condition is suitable to all constant modulus constellations.

C. PRESERVATION OF MV AND EXISTENCE OF MULTIPLE SOLUTIONS

In this Appendix, we discuss the conditions for which the MV obtained for the full-rank filter is preserved and the existence of multiple solutions in the proposed optimization method. Given a transformation matrix $\boldsymbol{T}_r(i) \in \mathbb{C}^{m\times r}$, where $r \leq m$, the MV is achieved if and only if $\boldsymbol{w}$ which minimizes (2.3) belongs to the Rang$\{\boldsymbol{T}_r(i)\}$, i.e., $\boldsymbol{w}(i)$ lies in the subspace generated by $\boldsymbol{T}_r(i)$. In this case, we have

$$\mathrm{MV}\big(\bar{\boldsymbol{w}}(i)\big) = \frac{1}{\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)}. \tag{C.1}$$

For a general $\boldsymbol{T}_r(i)$, we have

$$\mathrm{MV}\big(\bar{\boldsymbol{w}}(i)\big) \geq \frac{1}{\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)}. \tag{C.2}$$

From the above relations, we can conclude that there exists multiple solutions to the proposed optimization problem.

D. ANALYSIS OF THE OPTIMIZATION OF THE JIO CMV SCHEME

We carry out an analysis of the proposed JIO scheme with the CMV criterion and its optimization. Our approach is based on expressing the output of the proposed scheme and the proposed constraint in a convenient form that renders itself to analysis. Let us rewrite the proposed constrained optimization method in (4.7) using the method of Lagrange multipliers and express it by the Lagrangian

$$L_{\text{un}} = \mathbb{E}\big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i)|^2\big] + 2\,\Re[\lambda(\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) - 1)]. \tag{D.1}$$

In order to proceed, let us express $y(i)$ in an alternative and more convenient form as

$$\begin{aligned} y(i) &= \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(i) \\ &= \bar{\boldsymbol{w}}^H(i) \begin{bmatrix} \boldsymbol{x}(i) & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{x}(i) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \boldsymbol{x}(i) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{t}_1^*(i) \\ \boldsymbol{t}_2^*(i) \\ \vdots \\ \boldsymbol{t}_r^*(i) \end{bmatrix} \\ &= \bar{\boldsymbol{w}}^H(i)\boldsymbol{\Phi}^T(i)\boldsymbol{\tau}^*(i), \end{aligned} \tag{D.2}$$

where $\boldsymbol{\Phi}(i) \in \mathbb{C}^{rm\times r}$ is a block diagonal matrix with the input data vector $\boldsymbol{x}(i)$, and $\boldsymbol{\tau}^*(i) \in \mathbb{C}^{rm\times 1}$ is a vector with the columns of $\boldsymbol{T}_r(i)$ stacked on top of each other.

In order to analyze the proposed joint optimization procedure, we can rearrange the terms in $y(i)$ and define a single parameter vector $\boldsymbol{f}(i) = [\bar{\boldsymbol{w}}^T(i)\ \boldsymbol{\tau}^T(i)]^T \in \mathbb{C}^{r(m+1)\times 1}$. We can therefore further express $y(i)$ as

$$\begin{aligned} y(i) &= \boldsymbol{f}^H(i) \begin{bmatrix} \mathbf{0}_{r\times r} & \mathbf{0}_{r\times rm} \\ \boldsymbol{\Phi}(i) & \mathbf{0}_{rm\times rm} \end{bmatrix} \boldsymbol{f}(i) \\ &= \boldsymbol{f}^H(i)\boldsymbol{\Omega}(i)\boldsymbol{f}(i), \end{aligned} \tag{D.3}$$

where $\boldsymbol{\Omega}(i) \in \mathbb{C}^{r(m+1)\times r(m+1)}$ is a matrix which contains $\boldsymbol{\Phi}(i)$. Now let us perform a similar linear algebra transformation with the proposed constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = 1$

and express it as

$$\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = \boldsymbol{f}^H(i)\boldsymbol{A}(\theta_0)\boldsymbol{f}(i), \tag{D.4}$$

where $\boldsymbol{A}(\theta_0) \in \mathbb{C}^{r(m+1)\times r(m+1)}$ is structured as

$$\boldsymbol{A}(\theta_0) = \begin{bmatrix} \mathbf{0}_{r\times r} & \mathbf{0}_{r\times rm} \\ \boldsymbol{\Phi}_{\boldsymbol{a}(\theta_0)} & \mathbf{0}_{rm\times rm} \end{bmatrix},$$

and the block diagonal matrix $\boldsymbol{\Phi}_{\boldsymbol{a}(\theta_0)}(i) \in \mathbb{C}^{rm\times r}$ with the steering vector $\boldsymbol{a}(\theta_0)$ constructed as

$$\boldsymbol{\Phi}_{\boldsymbol{a}(\theta_0)} = \begin{bmatrix} \boldsymbol{a}(\theta_0) & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{a}(\theta_0) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \boldsymbol{a}(\theta_0) \end{bmatrix}. \tag{D.5}$$

At this point, we can alternatively express the Lagrangian as

$$L_{\text{un}} = \mathbb{E}\big[|\boldsymbol{f}^H(i)\boldsymbol{\Omega}(i)\boldsymbol{f}(i)|^2\big] + 2\Re[\lambda(\boldsymbol{f}^H(i)\boldsymbol{A}(\theta_0)\boldsymbol{f}(i) - 1)]. \tag{D.6}$$

We can examine the convexity of the above Lagrangian by computing the Hessian ($\boldsymbol{H}$)with respect to $\boldsymbol{f}(i)$ using the expression [95]

$$\boldsymbol{H} = \frac{\partial}{\partial \boldsymbol{f}^H(i)} \frac{\partial(L_{\text{un}})}{\partial \boldsymbol{f}(i)}, \tag{D.7}$$

and testing if the terms are positive semi-definite. Specifically, $\boldsymbol{H}$ is positive semi-definite if $\boldsymbol{v}^H\boldsymbol{H}\boldsymbol{v} \geq 0$ for all nonzero $\boldsymbol{v} \in \boldsymbol{C}^{r(m+1)\times r(m+1)}$ [19]. Therefore, the optimization problem is convex if the Hessian $\boldsymbol{H}$ is positive semi-definite.

Evaluating the partial differentiation in the expression given in (D.7) yields

$$\begin{aligned} \boldsymbol{H} = \mathbb{E}\big[&\boldsymbol{f}^H(i)\boldsymbol{\Omega}(i)\boldsymbol{f}(i)\boldsymbol{\Omega}(i) + \boldsymbol{\Omega}(i)\boldsymbol{f}(i)\boldsymbol{f}^H(i)\boldsymbol{\Omega}(i) \\ &+ \boldsymbol{\Omega}(i)\boldsymbol{f}^H(i)\boldsymbol{\Omega}(i)\boldsymbol{f}(i) + \boldsymbol{f}^H(i)\boldsymbol{\Omega}(i)\boldsymbol{\Omega}(i)\boldsymbol{f}(i) + 2\lambda\boldsymbol{A}(\theta_0)\big]. \end{aligned} \tag{D.8}$$

By examining $\boldsymbol{H}$, we verify that the second and fourth terms are positive semi-definite, whereas the first and the third terms are indefinite. The fifth term depends on the constraint, which is typically positive in the proposed scheme as verified in our studies, yielding a positive semi-definite matrix. Therefore, the optimization problem can not be classified as convex. It is however important to remark that our studies indicate that there are no local minima and there exists multiple solutions (which are possibly identical).

In order to support this claim, we have checked the impact on the proposed algorithms of different initializations. This study confirmed that the algorithms are not subject to

performance degradation due to the initialization although we have to bear in mind that the initialization $\boldsymbol{T}_r(0) = \mathbf{0}_{m\times r}$ annihilates the signal and must be avoided. We have also studied a particular case of the proposed scheme when $m = 1$ and $r = 1$, which yields the Lagrangian $L_{\text{un}}(\bar{\boldsymbol{w}}, \boldsymbol{T}_r) = \mathbb{E}\big[|\bar{w}T_r x|^2\big] + 2\Re\big[\lambda(\bar{w}T_r a(\theta_0) - 1)\big]$. Choosing T_r (the "scalar" projection) fixed with r equal to 1, it is evident that the resulting function $L_{\text{un}}(\bar{w}, T_r = 1, r) = |w^*\, x|^2 + 2\Re\big[\lambda(\bar{w}a(\theta_0) - 1)\big]$ is a convex one. In contrast to that, for a time-varying projection T_r the plots of the function indicate that the function is no longer convex but it also does not exhibit local minima. This problem can be generalized to the vector case, however, we can no longer verify the existence of local minima due to the multi-dimensional surface. This remains as an interesting open problem.

E. DERIVATION OF TRANSFORMATION MATRIX

In this appendix, we detail the derivation of $\boldsymbol{T}_r(i)$ and the simplification shown in (4.17) for reducing the computational complexity. Let us consider the derivation of $\boldsymbol{T}_r(i)$ obtained from the minimization of the Lagrangian

$$L_{\text{un}}(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i)) = \sum_{l=1}^{i} \alpha^{i-l} |\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{x}(l)|^2 + 2\,\Re[\lambda(\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) - 1)]. \tag{E.1}$$

Taking the gradient terms of the above expression with respect to $\boldsymbol{T}_r^*(i)$, we get

$$\begin{aligned} \nabla L_{\text{un}}(\boldsymbol{T}_r(i), \bar{\boldsymbol{w}}(i))_{T_r^*(i)} &= \sum_{l=1}^{i} \alpha^{i-l} \boldsymbol{x}(l)\boldsymbol{x}^H(l)\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i) + 2\lambda\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i) \\ &= \boldsymbol{R}(i)\boldsymbol{T}_r(i)\bar{\boldsymbol{R}}_{\bar{w}}(i) + 2\lambda\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i). \end{aligned} \tag{E.2}$$

Making the above gradient terms equal to zero yields

$$\boldsymbol{T}_r(i) = \boldsymbol{R}^{-1}(i)(-2\lambda)\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}. \tag{E.3}$$

Using the proposed constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = 1$ and substituting the above filter expression, we obtain the Lagrange multiplier $\lambda = -1/2(\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0))^{-1}$. Substituting λ into (E.3), we get

$$\boldsymbol{T}_r(i) = \frac{\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0)}. \tag{E.4}$$

The above expression for the matrix filter $\boldsymbol{T}_r(i)$ can be simplified by observing the quantities involved and making use of the proposed constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = 1$. Let us consider the term $\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)$ in the denominator of (E.4) and multiply it by the proposed constraint as follows:

$$\begin{aligned} \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i) &= \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) \\ &= \bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = 1. \end{aligned} \tag{E.5}$$

Now let us consider the term $\boldsymbol{a}^H(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)$ and rewrite it as follows:

$$\begin{aligned}\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i) &= \boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0)\\ &= \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\\ &= \boldsymbol{a}(\theta_0)\boldsymbol{a}^H(\theta_0)\boldsymbol{T}_r(i) = \boldsymbol{a}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0).\end{aligned} \tag{E.6}$$

Using the relations obtained in (E.5) and (E.6) into the expression in (E.4), we can get a simpler expression for the projection matrix as given by

$$\begin{aligned}\boldsymbol{T}_r(i) &= \frac{\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0)} = \frac{\boldsymbol{R}^{-1}(i)\overbrace{\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}^{\boldsymbol{a}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)}}{\underbrace{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)}_{1}\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0)}\\ &= \frac{\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0)\bar{\boldsymbol{a}}^H(\theta_0)}{\boldsymbol{a}^H(\theta_0)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_0)}.\end{aligned} \tag{E.7}$$

This completes the derivation and the simplification.

F. DERIVATION OF (5.31)

In this appendix, we show the details of the derivation of the expression for the transformation matrix in (5.31). Assuming $\bar{\boldsymbol{w}}(i) \neq \mathbf{0}$ is known, taking the gradient terms of (5.30) with respect to $\boldsymbol{T}_r(i)$, we get

$$\begin{aligned}\nabla L_{\text{un}_{\boldsymbol{T}_r(i)}} &= 2\sum_{l=1}^{i}|y(l)|^2\boldsymbol{x}(l)\boldsymbol{x}^H(l)\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i) - 2\sum_{l=1}^{i}\boldsymbol{x}(l)\boldsymbol{x}^H(l)\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i) + 2\lambda\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i) \\ &= 2\hat{\boldsymbol{R}}(i)\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i) - 2\hat{\boldsymbol{p}}(i)\bar{\boldsymbol{w}}^H(i) + 2\lambda\boldsymbol{a}(\theta_0)\bar{\boldsymbol{w}}^H(i).\end{aligned} \tag{F.1}$$

Making the above gradient terms equal to the zero matrix, right-multiplying the both sides by $\bar{\boldsymbol{w}}(i)$, and rearranging the expression, it becomes

$$\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i) = \hat{\boldsymbol{R}}^{-1}(i)\big[\hat{\boldsymbol{p}}(i) - \lambda\boldsymbol{a}(\theta_0)\big]. \tag{F.2}$$

If we define $\hat{\boldsymbol{p}}_{\hat{R}}(i) = \hat{\boldsymbol{R}}^{-1}(i)\big[\hat{\boldsymbol{p}}(i) - \lambda\boldsymbol{a}(\theta_0)\big]$, the solution of $\boldsymbol{T}_r(i)$ in (F.2) can be regarded to find the solution to the linear equation

$$\boldsymbol{T}_r(i)\bar{\boldsymbol{w}}(i) = \hat{\boldsymbol{p}}_{\hat{R}}(i). \tag{F.3}$$

Given a $\bar{\boldsymbol{w}}(i) \neq \mathbf{0}$, there exists multiple $\boldsymbol{T}_r(i)$ satisfying (F.3) in general. Therefore, we derive the minimum Frobenius-norm solution for stability. Let us express the quantities involved in (F.3) by

$$\boldsymbol{T}_r(i) = \begin{bmatrix}\bar{\boldsymbol{\rho}}_1(i) \\ \bar{\boldsymbol{\rho}}_2(i) \\ \vdots \\ \bar{\boldsymbol{\rho}}_m(i)\end{bmatrix};\ \ \hat{\boldsymbol{p}}_{\hat{R}}(i) = \begin{bmatrix}\hat{p}_{\hat{R},1}(i) \\ \hat{p}_{\hat{R},2}(i) \\ \vdots \\ \hat{p}_{\hat{R},m}(i)\end{bmatrix}. \tag{F.4}$$

The search for the minimum Frobenius-norm solution of (F.3) is reduced to the following m subproblems ($j = 1, \ldots, m$):

$$\min \|\bar{\boldsymbol{\rho}}_j(i)\|^2 \quad \text{subject to} \quad \bar{\boldsymbol{\rho}}_j(i)\bar{\boldsymbol{w}}(i) = \hat{p}_{\hat{R},j}(i). \tag{F.5}$$

The solution to (F.5) is the projection of $\bar{\boldsymbol{\rho}}(i)$ onto the hyperplane $\mathcal{H}_j(i) = \left\{\bar{\boldsymbol{\rho}}(i) \in \mathbb{C}^{1\times r} : \ \bar{\boldsymbol{\rho}}(i)\bar{\boldsymbol{w}}(i) = \hat{p}_{\hat{R},j}(i)\right\}$, which is given by

$$\bar{\boldsymbol{\rho}}_j(i) = \hat{p}_{\hat{R},j}(i)\frac{\bar{\boldsymbol{w}}^H(i)}{\|\bar{\boldsymbol{w}}(i)\|^2}. \tag{F.6}$$

Hence, the minimum Frobenius-norm solution of the transformation matrix is given by

$$\boldsymbol{T}_r(i) = \hat{\boldsymbol{p}}_{\hat{R}}(i)\frac{\bar{\boldsymbol{w}}^H(i)}{\|\bar{\boldsymbol{w}}(i)\|^2}. \tag{F.7}$$

Substituting the definition of $\hat{\boldsymbol{p}}_{\hat{R}}(i)$ into (F.7), we have

$$\boldsymbol{T}_r(i) = \hat{\boldsymbol{R}}^{-1}(i)\left[\hat{\boldsymbol{p}}(i) - \lambda\boldsymbol{a}(\theta_0)\right]\frac{\bar{\boldsymbol{w}}^H(i)}{\|\bar{\boldsymbol{w}}(i)\|^2}. \tag{F.8}$$

The multiplier λ can be obtained by incorporating (F.3) with the constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{T}_r^H(i)\boldsymbol{a}(\theta_0) = \gamma$, which is

$$\lambda = \frac{\hat{\boldsymbol{p}}(i)\hat{\boldsymbol{R}}^{-1}(i)\boldsymbol{a}(\theta_0) - \gamma}{\boldsymbol{a}^H(\theta_0)\hat{\boldsymbol{R}}^{-1}(i)\boldsymbol{a}(\theta_0)}. \tag{F.9}$$

Therefore, the expression of the transformation matrix in (5.30) can be obtained by substituting (F.9) into (F.8).

BIBLIOGRAPHY

[1] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing*, McGraw-Hill, 2005.

[2] S. R. Saunders, *Antennas and Propagation for Wireless Communication Systems, 2nd Ed.*, John Wiley Sons, 2007.

[3] J. Boccuzzi, *Signal Processing for Wireless Communications*, McGraw-Hill, 2007.

[4] M. A. Richards, *Fundamentals of Radar Signal Processing*, McGraw-Hill, 2005.

[5] R. E. Blabut, W. Millerm, and C. H. Wilcox, *Radar and Sonar, Part I*, Springer-Verlag, New York, 1991.

[6] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, *Wireless Sensor Networks*, Springer Science, 2004.

[7] P. S. Naidu, *Sensor Array Signal Processing, 2nd Ed.*, CRC, 2009.

[8] J. Li and P. Stoica, *Robust adaptive beamforming*, John Wiley, Inc. Hoboken, New Jersey, 2006.

[9] J. C. Liberti, Jr. and T. S. Rappaport, *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*, Prentice Hall PTR, 1999.

[10] Y. Zou, Z. L. Yu, and Z. Lin, "A robust algorithm for linearly constrained adaptive beamforming," *IEEE Signal Processing Letters*, vol. 11, pp. 26-29, Jan. 2004.

[11] C. Farsakh and J. A. Nossek, "Spatial covariance based downlink beamforming in an SDMA mobile radio system," *IEEE Trans. Commun.*, vol. 46, pp. 1497-1506, Nov. 1998.

[12] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, pp. 4-24, April. 1988.

[13] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," *IEEE Magazine Signal Processing*, pp. 30-39, Mar. 2002.

[14] O. L. Frost, "An algortihm for linearly constrained adaptive array processing," *IEEE Proc.*, AP-30, pp. 27-34, 1972.

[15] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas and Propagation*, vol. 30, pp. 27-34, Jan. 1982.

[16] B. D. Van Veen, "Adaptive convergence of linearly constrained beamformers based on the sample covariance matrix," *IEEE Trans. Signal Processing*, vol. 39, pp. 1470-1473, June 1991.

[17] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "A fast least-squares algorithm for linearly constrained adaptive filtering," *IEEE Trans. Signal Processing*, vol. 44, pp. 1168-1174, May 1996.

[18] S. Haykin, *Adaptive Filter Theory,* 4^{th} *edition*, NJ: Prentice-Hall, 2002.

[19] G. H. Golub and C. F. Van Loan, *Matrix Comuputations 3rd Ed.*, Baltimore, MD: Johns Hopkins Univ. Press, 1996.

[20] P. S. Chang and A. N. Willson, Jr., "Adaptive filtering using modified conjugate gradient," *Proc. 38th Midwest Symp. Circuits Syst.*, Rio de Janeiro, Brazil, pp. 243-246, Aug. 1995.

[21] A. M. Haimovich and Y. Bar-Ness, "An eigenanalysis interference canceler," *IEEE Trans. Signal Processing*, vol. 39, pp. 76-84, Jan. 1991.

[22] K. A. Byerly and R. A. Roberts, "Output power based partial adaptive array design," in *Twenty-Third Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, pp. 576-580, Oct. 1989.

[23] X. Wang and H. V. Poor, "Blind multiuser detection: A subspace approach," *IEEE Trans. Information Theory*, vol. 44, pp. 677-690, Mar. 1998.

[24] J. S. Goldstein and I. S. Reed, "Reduced-rank adaptive filtering," *IEEE Trans. Signal Processing*, vol. 45, pp. 492-496, Feb. 1997.

[25] J. S. Goldstein and I. S. Reed, "Subspace selection for partially adaptive sensor array processing," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, pp. 539-544, Apr. 1997.

[26] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the Wiener filter based on orthogonal projections," *IEEE Trans. Information Theory*, vol. 44, pp. 2943-2959, Nov. 1998.

[27] M. L. Honig and J. S. Goldstein, "Adaptive reduced-rank interference suppression based on the multistage Wiener filter," *IEEE Trans. Communications*, vol. 50, pp. 986-994, June 2002.

[28] 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 Trans. Signal Proc.*, vol. 56, pp. 2470-2482, Jun. 2008.

[29] D. A. Pados and S. N. Batalama, "Low-complexity blind detection of DS/CDMA signals: Auxiliary-vector receivers," *IEEE Trans. Commun.*, vol. 45, pp. 1586-1594, Dec. 1997.

[30] D. A. Pados and S. N. Batalama, "Joint space-time auxiliary-vector filtering for DS/CDMA systems with antenna arrays," *IEEE Trans. Commun.*, vol. 47, pp. 1406-1415, Sep. 1999.

[31] D. A. Pados and G. N. Karystinos, "An iterative algorithm for the computation of the MVDR filter," *IEEE Trans. Signal Processing*, vol. 49, pp. 290-300, Feb. 2001.

[32] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, John Wiley Sons, 2002.

[33] J. Capon, "High resolution frequency-wavenumber spectral analysis," *Proc. of the IEEE*, vol. 57, pp. 1408-1418, Aug. 1969.

[34] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. on Antennas and Propagation*, vol. 34, pp. 276-280, Aug. 1986.

[35] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *Optical Engineering*, vol. 29, pp. 296-313, Apr. 1990.

[36] R. Grover, D. A. Pados, and M. J. Medley, "Subpace direction finding with an auxiliary-vector analysis," *IEEE Trans. Signal Processing*, vol. 55, pp. 758-763, Feb. 2007.

[37] H. Semira, H. Belkacemi, and S. Marcos, "High-resolution source localization algorithm based on the conjugate gradient," *EURASIP Journal on Advance in Signal Processing*, vol. 2007, pp. 1-9, Mar. 2007.

[38] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1553-1560, Oct. 1988.

[39] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Trans. Acoust., Speech ,Signal Processing*, vol. 37, pp. 720-741, Feb. 1989.

[40] A. H. Sayed and M. Rupp, "A time-domain feedback analysis of adaptive algorithms via the small gain theorem," *Proc. SPIE*, vol. 2563, pp. 458-469, May 1995.

[41] A. J. Barabell, "Improving the resolution performance of eigenstructure-based direction finding algorithms," *Proc. of the IEEE Int'l. Conf. on Acoustics, Speech, and Signal Processing-83*, pp. 336-339, 1983.

[42] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Process.*, vol. 41, pp. 2075-2087, Jun. 1993.

[43] H. J. Kushner and J. Yang, "Analysis of adaptive step-size sa algorithms for parameter tracking," *IEEE Trans. Autom. Control*, vol. 40, pp. 1403-1410, Aug. 1995.

[44] J. Chung, C. S. Hwang, K. Kim, and Y. K. Kim, "A random beamforming technique in MIMO systems exploiting multiuser diversity," *IEEE J. Sel. Areas Commun.*, vol. 21, pp. 848-855, Jun. 2003.

[45] A. B. Gershman, "Robust adaptive beamforming in sensor arrays," *Int. J. Electron. Commun.*, vol. 53, pp. 1365-1376, Dec. 1999.

[46] S. Anderson, M. Millnert, M. Viberg, and B. Wahlberg, "An adaptive array for mobile communication systems," *IEEE Trans. Vehicular Technology*, vol. 40, pp. 230-236, Feb. 1991.

[47] H. J. Kushner and J. Yang, "Analysis of adaptive step-size sa algorithms for parameter tracking," *IEEE Trans. Autom. Control*, vol. 40, pp. 1403-1410, Aug. 1995.

[48] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 1633-1642, July 1992.

[49] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-Type algorithm: analysis and simulations," *IEEE Trans. Signal Processing*, vol. 45, pp. 631-639, Mar. 1997.

[50] R. C. de Lamare and R. Sampaio-Neto, "Low-complexity variable step-size mechanisms for stochastic gradient algorithms in minimum variance CDMA receivers," *IEEE Trans. Signal Processing*, vol. 54, pp. 2302-2317, Jun. 2006.

[51] Johnson R. Jr., Schniter P., Endres T. J. and Behm J. D., "Blind equalization using the constant modulus criterion: a review," *IEEE Proceedings*, vol. 86, pp. 1927-1950, Oct. 1998.

[52] Z. Y. Xu and P. Liu, "Code-constrained blind detection of CDMA signals in multipath channels," *IEEE Signal Processing Letters*, vol. 9, pp. 389-392, Dec. 2002.

[53] D. N. Godard, "Self-recovering equalization and carrier tracking in two-demensional data communication systems," *IEEE Trans. Commun.*, vol. 28, pp. 1867-1875, Nov. 1980.

[54] H. H. Zeng, L. Tong, and C. R. Johnson, "Relationships between the constant modulus and Wiener receivers," *IEEE Trans. Inf. Theory*, vol. 44, pp. 1523-1538, July 1998.

[55] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[56] A. V. Keerthi, A. Mathur, and J. Shynk, "Misadjustment and tracking analysis of the constant modulus array," *IEEE Trans. Signal Processing*, vol. 46, pp. 51-58, Jan. 1998.

[57] O. W. Kwon, C. K. Un, and J. C. Lee, "Performance of constant modulus adaptive digital filters for interference cancellation," *IEEE Trans. Signal Processing*, vol. 26, pp. 185-196, Feb. 1992.

[58] J. Mai and A. H. Sayed, "A feedback approach to the steady-state performance of fractionally spaced blind adaptive equalizers," *IEEE Trans. Signal Processing*, vol. 48, pp. 80-91, Jan. 2000.

[59] M. Rupp and A. H. Sayed, "A time-domain feedback analysis of filtered-error adaptive gradient algorithms," *IEEE Trans. Signal Processing*, vol. 44, pp. 1428-1439, June 1996.

[60] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyses of adaptive filters," *IEEE Trans. Signal Processing*, vol. 49, pp. 314-324, Feb. 2001.

[61] J. B. Whitehead and F. Takawira, "Performance analysis of the linearly constrained constant modulus algorithm-based multiuser detector," *IEEE Trans. Signal Processing*, vol. 53, pp. 643-653, Feb. 2005.

[62] S. Verdu, *Multiuser Detection*. Cambridge, UK: Cambridge Univ. Press, 1998.

[63] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley Sons, NY, 2003.

[64] G. K. Boray and M. D. Srinath, "Conjugate gradient techniques for adaptive filtering," *IEEE Trans. Circuits Syst. I*, vol. 39, pp. 1-10, Jan. 1992.

[65] S. Burykh and K. Abed-Meraim, "Reduced-rank adaptive filtering using krylov subspace," *EURASIP Journal on Applied Signal Processing*, pp. 1387-1400, 2002.

[66] T. Bose and M. Q. Chen, “Conjugate gradient method in adaptive bilinear filtering,” *IEEE Trans. Signal Processing*, vol. 43, pp. 1503-1508, Jan. 1995.

[67] Z. Fu and E. M. Dowling, “Conjugate gradient eigenstructure tracking for adaptive spectral estimation,” *IEEE Trans. Signal Processing*, vol. 43, pp. 1151-1160, May. 1995.

[68] J. A. Apolinário, Jr., M. L. R. de Campos, and C. P. Bernal O., “The constrained conjugate gradient algorithm”, *IEEE Letters Signal Processing*, vol. 7, pp. 351-354, Dec. 2000.

[69] H. Cox, R. M. Zeskind, and M. M. Owen, ”Robust adaptive beamforming,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 35, pp. 1365-1376, Otc. 1987.

[70] Y. S. Hwu and M. D. Srinath, ”A neural network approach to design of smart antennas for wireless communication systems,” *Proc. of the 31th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 145-148, 1998.

[71] S. Fiori, ”Neural minor component analysis approach to robust constrained beamforming,” *IEE Proc. of Version, Image and Signal Processing*, vol. 150, pp. 205-218, Aug. 2003.

[72] M. E. Weippert, J. D. Hiemstra, J. S. Goldstein, and M. D. Zoltowski, ”Insights from the relationship between the multistage Wiener filter and the method of conjugate gradients,” *Proc. of IEEE Workshop on Sensor Array and Multichannel Signal Processing*, pp. 388-392, Aug. 2002.

[73] L. L. Scharf, E. K. P. Chong, M. D. Zoltowski, J. S. Goldstein, and I. S. Reed, ”Subspace Expansion and the equivalence of conjugate direction and multistage Wiener filters,” *IEEE Trans. Signal Processing*, vol. 56, pp. 5013-5019, Oct. 2008.

[74] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison-Wesley, 1984.

[75] D. S. Watkins, *Fundamentals of Matrix Computations, 2nd Ed.*, Wiley-Interscience, May. 2002.

[76] P. S. Chang and A. N. Willson, Jr., “Analysis of conjugate gradient algorithms for adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 48, pp. 409-418, Feb. 2000.

[77] M. Al-Baali, “Descent property and global convergence of the Fletcher-Reeves method with inexact line search,” *IMA J. Number. Anal.*, vol. 5, pp. 121-124, Jan. 1985.

[78] R. Fletcher, *Practical Methods of Optimization 2nd Ed.*, Chichester, U.K.: Wiley, 1987.

[79] D. F. Shanno, "Conjugate gradient methods with inexact searches," *Math. Oper. Res.*, vol. 3, pp. 244-256, Aug. 1978.

[80] W. Chen, U. Mitra, and P. Schniter, "On the equivalence of three reduced rank linear estimators with applications to DS-CDMA," *IEEE Trans. Information Theory*, vol. 48, pp. 2609-2614, Sep. 2002.

[81] R. C. de Lamare, "Adaptive reduced-rank LCMV beamforming algorithms based on joint iterative optimization of filters," *Electronics Letters*, vol. 44, no. 9, Apr. 2008.

[82] H. Qian and S. N. Batalama, "Data record-based criteria for the selection of an auxiliary vector estimator of the MMSE/MVDR filter," *IEEE Trans. Communications*, vol. 51, pp. 1700-1708, Oct. 2003.

[83] M. J. Rude and L. J. Griffiths, "Incorporation of linear constraints into the constant modulus algorithm," *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, vol. 2, pp. 968-971, May 1989.

[84] S. J. Chern, C. H. Sun, and C. C. Chang, "Blind adaptive DS-CDMA receivers with sliding window constant modulus GSC-RLS algorithm," *Intelligent Signal Processing and Communication System, 2006, ISPACS-06., 2006 International Conference on*, pp. 979-982, Dec. 2006.

[85] S. J. Chern and C. Y. Chang, "Adaptive MC-CDMA receiver with constrained constant modulus IQRD-RLS algorithm for MAI suppression," *Signal Processing*, vol. 83, pp. 2209-2226, Oct. 2003.

[86] J. S. Goldstein and I. S. Reed, "Theory of partially adaptive radar," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, pp. 1309-1325, Oct. 1997.

[87] V. Nagesha and S. Kay, "On frequency estimation with IQML algorithm," *IEEE Trans. Signal Processing*, vol. 42, pp. 2509- 2513, Sep. 1994.

[88] J. R. Guerci, J. S. Goldstein, and I. S. Reed, "Optimal and adaptive reduced-rank STAP," *IEEE Trans. Aerospace and Electronic Systems*, vol. 36, pp. 647-663, Apr. 2000.

[89] R. Y. M. Wong and R. Adve, "Reduced-Rank Adaptive Filtering Using Localized Processing for CDMA Systems," *IEEE Trans. Vehicular Technology*, vol. 56, pp. 3846-3856, Nov. 2007.

[90] C. J. Xu, G. Z. Feng, and K. S. Kwak, "A modified constrained constant modulus approach to blind adaptive multiuser detection," *IEEE Trans. Communications*, vol. 49, pp. 1642-1648, Sep. 2001.

[91] B. L. Mathews, L. Mili, and A. I. Zaghloul, "Auxiliary vector selection algorithms for adaptive beamforming," *Antennas and Propagation Society International Symposium*, pp. 271-274, July 2005.

[92] B. L. Mathews, L. Mili, and A. I. Zaghloul, "Auxiliary vector selection algorithms for adaptive beamforming," *Antennas and Propagation Society International Symposium*, pp. 271-274, July 2005.

[93] R. C. de Lamare and R. Sampaio-Neto, "Blind adaptive code-constrained constant modulus RLS algorithm for CDMA receivers in frequency selective channels," *IEEE Vehicular Technology Conf.*, vol. 3, pp. 1708-1711, May. 2004.

[94] R. C. de Lamare and R. Sampaio-Neto, "Blind adaptive code-constrained constant modulus algorithms for CDMA interference suppression in multipath," *IEEE Communications Letters*, vol. 9, pp. 334-336 Apr. 2005.

[95] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd Ed., 1999.

[96] H. Krim and M. Viberg, "Two decades of array signal processing research," *IEEE Signal Processing Magazine*, vol.13, pp. 67-94, July 1996.

[97] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol. 44, pp. 95-107, Jan. 1995.

[98] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Trans. Signal Processing*, vol. 46, pp. 1967-1979, Jul. 1998.

[99] P. Strobach, "Fast recursive subspace adaptive ESPRIT algorithms," *IEEE Trans. Signal Processing*, vol. 46, pp. 2413-2430, Sep. 1998.

[100] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Processing*, vol. 53, pp. 2931-2941, Aug. 2005.

[101] E. E. Tyrtyshnikov, *A Brief Introduction to Numerical Analysis*, Birkhäuser Boston, 1997.

[102] J. E. Evans, J. R. Johnson, and D. F. Sun, "High resolution angular spectrum estimation techniques for terrain scattering analysis and angle of arrival estimation in ATC navigation and surveillance system," M.I.T. Lincoln Lab, Lexington, MA, Rep. 582, 1982.

[103] S. U. Pillai and B. H. Kwon, "Forward/Backword spatial smoothing techniques for coherent signal identification," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, pp. 8-15, Jan. 1989.

[104] M. Martinez-Ramon, J. Arenas-Garcia, A. Navia-Vasquez, and A. R. Figueiras-Vidal, "An adaptive combination of adaptive filters for plant identification," in *Proc. 14th Int. Conf. Ditial Signal Processing*, Santorini, Greece, 2002, pp. 1195-1198.

[105] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Processing*, vol. 54, pp. 1078-1090, Mar. 2006.

[106] N. J. Bershad, J. C. M. Bermudez, and J. Tourneret, "An affine combination of two LMS adaptive filters-transient mean-square analysis," *IEEE Trans. Signal Processing*, vol. 56, pp. 1853-1864, May 2008.

[107] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*, 1st ed. Chapman Hall/CRC, 2002.

[108] R. W. Hamming, *Numerical Methods for Scientists and Engineers*, 2nd, Courier Dover Publications, 1986.