www.ietdl.org

Published in IET Signal Processing
Received on 5th October 2009
Revised on 17th February 2010
doi: 10.1049/iet-spr.2009.0243

ISSN 1751-9675

Constrained adaptive filtering algorithms based on conjugate gradient techniques for beamforming

L. Wang R.C. de Lamare
Communications Research Group, Department of Electronics, The University of York, York YO10 5DD, UK
E-mail: lw517@york.ac.uk

Abstract: This article proposes constrained adaptive algorithms based on the conjugate gradient (CG) method for adaptive beamforming. The proposed algorithms are derived for the implementation of the beamformer according to the minimum variance and constant modulus criteria subject to a constraint on the array response. A CG-based weight vector strategy is created for enforcing the constraint and computing the weight expressions. The devised algorithms avoid the covariance matrix inversion and exhibit fast convergence with low complexity. A complexity analysis compares the proposed algorithms with the existing ones. The convergence properties of the CCM criterion are studied, conditions for convexity are established and a convergence analysis for the proposed algorithms is derived. Simulation results are conducted for both stationary and non-stationary scenarios, showing the convergence and tracking ability of the proposed algorithms.

1 Introduction

Linearly constrained adaptive filtering algorithms are important in many applications of signal processing and communications such as beamforming and interference suppression for code-division-multiple-access (CDMA) systems. In these applications, the linear constraints correspond to prior knowledge of certain parameters such as the direction of arrival (DOA) of the desired user in antenna array processing [1] and the signature sequence of the desired signal in CDMA interference suppression [2].

Numerous constrained adaptive algorithms have been reported in the last decades [3]. Among the existing algorithms, the stochastic gradient (SG) is a low complexity algorithm but converges slowly for correlated data inputs. The recursive least squares (RLS) [3] algorithm has fast convergence but needs high complexity and suffers from numerical instability. The conjugate gradient (CG) algorithm has an attractive trade-off between performance and complexity, since it enjoys a convergence comparable to the RLS with a computational requirement which is intermediate between the SG and RLS methods [4]. Another class of algorithms are those based on the multi-stage Wiener filter (MSWF) [5, 6] and the auxiliary-vector filtering (AVF) [7, 8]. The basis vectors of the CG, MSWF and AVF span the same Krylov subspace [9]. Their main differences lie on the computational cost, structure of adaptation and ease of implementation. Many adaptive algorithms based on the CG approach have been reported [10–12]. 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 desired user. For wireless communications, this array response is equivalent to a CDMA effective signature and to a spatial signature of a multiple-input and multiple-output (MIMO) system.

Another important issue that has been considered in a number of recent works is the design criterion. The most promising criteria found in the literature are the constrained minimum variance (CMV) [13, 14] and the constrained constant modulus (CCM) [15, 16] owing to their simplicity and effectiveness. The CMV criterion aims to minimise the beamformer output power while maintaining the array response on the DOA of the desired signal. The CCM criterion is a positive measure [17] of the average amount that the beamformer output deviates from a constant modulus condition subject to a constraint on the DOA of the desired signal.

In this paper, our first contribution is the development of CG adaptive filtering algorithms for beamforming. We derive two CG-based algorithms for the efficient implementation of the beamformer design according to the CMV and CCM criteria. The existing methods yield a system of equations that requires the costly matrix inversion for ensuring the constraint and obtaining the solution. The proposed algorithms are motivated by the need to circumvent this problem and for developing cost-effective algorithms. A CG-based weight vector strategy is devised to incorporate the constraint of the array response in the proposed algorithms. We create a simple relation between the CG-based weight vector and the matrix inversion and the array response of the desired user. We develop an iterative algorithm to calculate the CG-based weight vector. By substituting the CG-based weight vector into the proposed weight expression, we obtain the constrained weight solution without the matrix inversion. The proposed algorithms exhibit excellent performance with low complexity and address the numerical instability found in the RLS methods [3]. The second contribution of this work is the analysis of the proposed algorithms. We first study the convexity property of the CCM criterion and then show a complexity comparison of the existing and proposed algorithms. The convergence analysis of the proposed algorithms is also derived to explain the impact of the condition number [18] on the weight vector error. Simulations evaluate the performance of the proposed algorithms against the best known techniques.

2 System model and problem statement

In this section, we describe a system model of a smart antenna system. Based on this model, the beamformer is designed according to the CMV and CCM criteria, and the problem statement is introduced.

2.1 System model

Let us consider a smart antenna system equipped with a uniform linear array (ULA) with m elements. For simplicity, we consider a ULA even though the problem can be extended to arbitrary antenna arrays. We focus on the adaptive filtering strategy for adjusting the beamformer's weights $w_{k'}$ $(k' = 1, \ldots, m)$. Suppose that $q (q \leq m)$ narrowband signals impinge on the ULA. The received vector $\boldsymbol{x}(i) \in \mathcal{C}^{m\times 1}$ of the sensor array at time instant i is

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

where $\boldsymbol{\theta} = [\theta_0, \ldots, \theta_{q-1}]^{\mathrm{T}} \in \mathcal{R}^{q\times 1}$ is the vector of the DOA, $\boldsymbol{A}(\boldsymbol{\theta}) = [\boldsymbol{a}(\theta_0), \ldots, \boldsymbol{a}(\theta_{q-1})] \in \mathcal{C}^{m\times q}$ is the complex matrix composed of the signal steering vectors $\boldsymbol{a}(\theta_k) = [1, \mathrm{e}^{-2\pi \mathrm{j}(d/\lambda_c)\cos\theta_k}, \ldots, \mathrm{e}^{-2\pi \mathrm{j}(m-1)(d/\lambda_c)\cos\theta_k}]^{\mathrm{T}} \in \mathcal{C}^{m\times 1}$ $(k = 0, \ldots, q-1)$, where λ_c is the wavelength and d $(d = \lambda_c/2$ in general) is the inter-element distance of the ULA. In the following derivation, we assume that θ_0 corresponds to the direction of the desired user with respect to the antenna array and is known beforehand by the receiver. For implementation, θ_0 can be estimated by DOA estimation algorithms [19]. The vector $\boldsymbol{s}(i) \in \mathcal{C}^{q\times 1}$ is the source data with its entries are uncorrelated with each other. The vector $\boldsymbol{n}(i) \in \mathcal{C}^{m\times 1}$ is the complex vector of sensor noise, which is assumed to be a zero-mean spatially and temporally white Gaussian process, N is the number of snapshots and $(\cdot)^{\mathrm{T}}$ stands for transpose. The output of the adaptive filter is given by

$$y(i) = \boldsymbol{w}^{\mathrm{H}}(i)\boldsymbol{x}(i) \tag{2}$$

where $\boldsymbol{w}(i) = [w_1(i), \ldots, w_m(i)]^{\mathrm{T}} \in \mathcal{C}^{m\times 1}$ is the complex weight vector and $(\cdot)^{\mathrm{H}}$ stands for Hermitian transpose.

2.2 Design criteria and problem statement

We present the design criteria that will be considered for the development of the proposed adaptive algorithms. The CMV criterion solves the following optimisation problem

$$\begin{aligned} &\min J_{\mathrm{mv}}(\boldsymbol{w}) = \boldsymbol{w}^{\mathrm{H}}\boldsymbol{R}\boldsymbol{w} \\ &\text{subject to} \quad \boldsymbol{w}^{\mathrm{H}}(i)\boldsymbol{a}(\theta_0) = \gamma, \quad i = 1, \ldots, N \end{aligned} \tag{3}$$

where $\boldsymbol{R} = E[\boldsymbol{x}(i)\boldsymbol{x}^{\mathrm{H}}(i)] \in \mathcal{C}^{m\times m}$ is the covariance matrix, γ is set to ensure the convexity of the cost function and $\boldsymbol{a}(\theta_0)$ is the steering vector of the desired user.

The CCM criterion solves the following optimisation problem

$$\begin{aligned} &\min J_{\mathrm{cm}}(\boldsymbol{w}) = E[(|y(i)|^p - \delta_p)^2] \\ &\text{subject to} \quad \boldsymbol{w}^{\mathrm{H}}(i)\boldsymbol{a}(\theta_0) = \gamma, \quad i = 1, \ldots, N \end{aligned} \tag{4}$$

where the constant δ_p is suitably chosen to guarantee that the weight solution is close to the global minimum [20] and in general, $p = 2$ is selected to consider the cost function as the expected deviation of the squared modulus of the beamformer output to a constant, say $\delta_p = 1$.

In order to solve (3) and (4), we resort to the method of Lagrange multipliers [3]. The weight expressions are

$$\boldsymbol{w}_{\mathrm{cmv}} = \frac{\gamma \boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)}{\boldsymbol{a}^{\mathrm{H}}(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)} \tag{5}$$

$$\boldsymbol{w}_{\mathrm{ccm}} = \frac{\gamma \boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0)}{\boldsymbol{a}^{\mathrm{H}}(\theta_0)\boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0)} \tag{6}$$

where $\boldsymbol{R}_y = E[e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^{\mathrm{H}}(i)] \in \mathcal{C}^{m\times m}$ is a matrix with cross-correlations between $y(i)$ and $\boldsymbol{x}(i)$, and $e_y = ||y(i)|^2 - 1|$. It is worth noting that (6) is not a solution for the CCM criterion since $\boldsymbol{R}_y$ depends on $y(i)$, which is a function of $\boldsymbol{w}_{\mathrm{ccm}}$. A solution can be obtained by setting an initial value of $\boldsymbol{w}_{\mathrm{ccm}}$ and running an iterative procedure, which will be shown in the following part.

Considering the expressions in (5) and (6), we can manipulate and organise them into the following systems of equations

$$(\boldsymbol{a}^{\mathrm{H}}(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0))\boldsymbol{R}\boldsymbol{w}_{\mathrm{cmv}} = \gamma\boldsymbol{a}(\theta_0) \quad (7)$$

$$(\boldsymbol{a}^{\mathrm{H}}(\theta_0)\boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0))\boldsymbol{R}_y\boldsymbol{w}_{\mathrm{ccm}} = \gamma\boldsymbol{a}(\theta_0) \quad (8)$$

where the terms $\boldsymbol{a}^{\mathrm{H}}(\theta_0)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)$ and $\boldsymbol{a}^{\mathrm{H}}(\theta_0)\boldsymbol{R}_y^{-1}\boldsymbol{a}(\theta_0)$ 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 [3, 18, 19, 21]. Among them, the CG is an efficient method with very attractive trade-off between performance and complexity. Here, we plan to use the CG-based algorithm to enforce the constraint and adjust the beamformer's coefficients.

3 Proposed adaptive algorithms

In this section, we devise a CG-based weight vector strategy to incorporate the constraint of the array response and the matrix inversion in (5) and (6), and propose two adaptive filtering algorithms, that is, the conventional conjugate gradient (CCG) and the modified conjugate gradient (MCG), for the beamformer design according to the CMV and CCM criteria, respectively. The proposed methods avoid the matrix inversion and exhibit excellent performance with low complexity.

3.1 CG algorithm

The CG algorithm can be employed for solving optimisation problems of the form [22, 23]

$$\min \mathcal{J}(\boldsymbol{v}) = \boldsymbol{v}^{\mathrm{H}}\boldsymbol{R}\boldsymbol{v} - 2\Re\{\boldsymbol{b}^{\mathrm{H}}\boldsymbol{v}\} \quad (9)$$

where $\boldsymbol{R} \in \mathcal{C}^{m\times m}$ is the covariance matrix of $\boldsymbol{x}(i)$, $\boldsymbol{b} \in \mathcal{C}^{m\times 1}$ is the cross-correlation vector between $\boldsymbol{x}(i)$ and the desired response $d(i)$, and $\boldsymbol{v} \in \mathcal{C}^{m\times 1}$ is the CG weight vector. The operator $\Re(\cdot)$ selects the real part of the argument. The CG algorithm solves (9) by iteratively updating the CG weight vector as

$$\boldsymbol{v}_k = \boldsymbol{v}_{k-1} + \alpha_k\boldsymbol{p}_k \quad (10)$$

where $\boldsymbol{p}_k$ is the direction vector with conjugacy, that is, $\boldsymbol{p}_k^{\mathrm{H}}\boldsymbol{R}\boldsymbol{p}_l = 0$ for $k \neq l$, α_k is calculated by substituting (10) into (9) and then minimising 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 [24].

The direction vector $\boldsymbol{p}_k$ is obtained by a linear combination of the previous direction vector and the negative gradient vector $\boldsymbol{g}_k = \boldsymbol{b} - \boldsymbol{R}\boldsymbol{v}_k$ [22], which can be expressed as

$$\boldsymbol{p}_{k+1} = \boldsymbol{g}_k + \beta_k\boldsymbol{p}_k \quad (11)$$

where β_k is chosen to provide conjugacy for the direction vectors. The CG algorithm and its properties can be found in [18, 24, 25].

3.2 CG-based weight vector strategy

According to the CG optimisation problem, by taking the gradient of (9) with respect to $\boldsymbol{v}$, equating it to a null vector and rearranging the expression, we obtain $\boldsymbol{v} = \boldsymbol{R}^{-1}\boldsymbol{b}$. This expression is equivalent to the numerator of (5) if $\boldsymbol{b}$ is replaced by $\boldsymbol{a}(\theta_0)$ and the preset γ is not considered, which is

$$\boldsymbol{v} = \boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0) \quad (12)$$

where $\boldsymbol{b} = E[d^*(i)\boldsymbol{x}(i)] = \boldsymbol{a}(\theta_0)$ if we assume that the source data and the noise are mutually independent [26]. It should be remarked that $\boldsymbol{a}(\theta_0)$ is a constant known beforehand by the receiver unlike $\boldsymbol{b}$ that has to be estimated in practice. From (12), we can employ the CG algorithm to calculate $\boldsymbol{v}$ iteratively. Expression (12) is fundamental for this work as it creates a new CG-based weight vector $\boldsymbol{v}$. We devise a simple relation with the matrix inversion and the steering vector of the desired user. In the proposed algorithm, $\boldsymbol{v}$ is iterated and then substituted into (5) to replace $\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_0)$ for the calculation of $\boldsymbol{w}_{\mathrm{cmv}}$. Note that $\boldsymbol{v}$ in (12) is not the weight solution for (3) but regarded as an intermediate weight vector for enforcing the constraint and avoiding the matrix inversion. Besides, the numerical problems caused by the matrix inversion can be addressed in the proposed strategy.

3.3 Proposed CG algorithms

In this part, we introduce the proposed CCG and the MCG algorithms with respect to the CMV and CCM criteria, respectively. For the CCG algorithm, the iteration procedure for the CG-based weight vector $\boldsymbol{v}$ is executed per snapshot. For the ith snapshot, $\boldsymbol{R}$ and $\boldsymbol{R}_y$ are replaced by their recursive forms [3], which are fixed throughout the K iterations of the CCG operation. Since the CG algorithm was elaborated in [18], we omit the details here and summarise the procedures for the CMV and CCM criteria in Figs. 1 and 2, respectively, where λ and λ_y are the forgetting factors, and δ and δ_y are the regularisation parameters to initialise the covariance matrix.

The CCG algorithm operates K iterations per snapshot and runs the reset periodically [24, 25] for convergence. These operations increase the computational load in the

doi: 10.1049/iet-spr.2009.0243

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 $k = 1, \ldots, K$:

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

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

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

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

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

$k = k + 1$

STEP 3: After K **iterations:**

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

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

$i = i + 1$

Figure 1 *CMV-CCG algorithm*

Initialization:

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

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

STEP 1: Start:

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

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

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

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

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

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

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

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

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

$k = k + 1$

STEP 3: After K **iterations:**

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

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

$i = i + 1$

Figure 2 *CCM-CCG algorithm*

sample-by-sample update. Here, we describe an 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.

3.3.1 Proposed CMV-MCG algorithm: The MCG algorithm was motivated from [27] for adaptive filtering. Here, we bring the idea into the paper 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 is expressed by

$$\tilde{\boldsymbol{v}}_{\mathrm{cmv}}(i) = \tilde{\boldsymbol{v}}_{\mathrm{cmv}}(i-1) + \tilde{\alpha}_{\mathrm{cmv}}(i)\tilde{\boldsymbol{p}}_{\mathrm{cmv}}(i) \tag{13}$$

where $\tilde{\boldsymbol{p}}_{\mathrm{cmv}}(i)$ is the direction vector, $\tilde{\alpha}_{\mathrm{cmv}}(i)$ is the corresponding coefficient and, in what follows, all the quantities related to the proposed MCG algorithm are denoted by an over tilde. Note that $\tilde{\boldsymbol{v}}_{\mathrm{cmv}}(i)$ formulates the relation with $\hat{\boldsymbol{R}}(i)$ and $\boldsymbol{a}(\theta_0)$ to enforce the constraint and solve the systems of equations.

From [27], one way to make the CG algorithm work with one iteration per snapshot is the application of the degenerated scheme. Under this condition, we need to ensure that the coefficient $\tilde{\alpha}_{\mathrm{cmv}}(i)$ satisfies the convergence bound [27, 28], which is given by

$$0 \le \tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i) \le 0.5\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i-1) \tag{14}$$

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

$$\begin{aligned}\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i) &= (1-\lambda)\boldsymbol{a}(\theta_0) + \lambda\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i-1)\\ &\quad - \tilde{\alpha}_{\mathrm{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\mathrm{cmv}}(i) - \boldsymbol{x}(i)\boldsymbol{x}^{\mathrm{H}}(i)\tilde{\boldsymbol{v}}_{\mathrm{cmv}}(i-1)\end{aligned} \tag{15}$$

Premultiplying (15) by $\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)$, taking the expectation of both sides and considering $\tilde{\boldsymbol{p}}_{\mathrm{cmv}}(i)$ uncorrelated with $\boldsymbol{x}(i)$, $\boldsymbol{a}(\theta_0)$ and $\tilde{\boldsymbol{v}}_{\mathrm{cmv}}(i-1)$ [27] yield

$$\begin{aligned}E[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i)] &\simeq \lambda E[\tilde{\boldsymbol{p}}_{\mathrm{cmvh}}^{\mathrm{H}}(i)\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i-1)]\\ &\quad - \lambda E[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\boldsymbol{a}(\theta_0)]\\ &\quad - E[\tilde{\alpha}_{\mathrm{cmv}}(i)[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\mathrm{cmv}}(i)]\end{aligned} \tag{16}$$

where the optimal solution $\boldsymbol{R}\tilde{\boldsymbol{v}}_{\mathrm{cmv,opt}} = \boldsymbol{a}(\theta_0)$ and $E[\tilde{\boldsymbol{v}}_{\mathrm{cmv}}(i-1) - \tilde{\boldsymbol{v}}_{\mathrm{cmv,opt}}] \approx 0$ have been used with the assumption that the algorithm converges. Making a rearrangement of (16) and following the convergence bound (14), we obtain

$$\begin{aligned}&\frac{(\lambda-0.5)E[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i-1)] - \lambda E[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\boldsymbol{a}(\theta_0)]}{E[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\mathrm{cmv}}(i)]}\\ &\quad \le E[\tilde{\alpha}_{\mathrm{cmv}}(i)] \le \frac{\lambda E[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\tilde{\boldsymbol{g}}_{\mathrm{cmv}}(i-1) - \tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\boldsymbol{a}(\theta_0)]}{E[\tilde{\boldsymbol{p}}_{\mathrm{cmv}}^{\mathrm{H}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\mathrm{cmv}}(i)]}\end{aligned} \tag{17}$$

The inequalities in (17) are satisfied if we define

$$\tilde{\alpha}_{\text{cmv}}(i) = [\tilde{\boldsymbol{p}}^{\text{H}}_{\text{cmv}}(i)\hat{\boldsymbol{R}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i)]^{-1}\{\lambda[\tilde{\boldsymbol{p}}^{\text{H}}_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1) - \tilde{\boldsymbol{p}}^{\text{H}}_{\text{cmv}}(i)\boldsymbol{a}(\theta_0)] - \tilde{\eta}\tilde{\boldsymbol{p}}^{\text{H}}_{\text{cmv}}(i)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)\} \quad (18)$$

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

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

$$\tilde{\boldsymbol{p}}_{\text{cmv}}(i+1) = \tilde{\boldsymbol{g}}_{\text{cmv}}(i) + \tilde{\beta}_{\text{cmv}}(i)\tilde{\boldsymbol{p}}_{\text{cmv}}(i) \quad (19)$$

where $\tilde{\beta}_{\text{cmv}}(i)$ is computed for avoiding the reset procedure by employing the Polak–Ribiere approach [24, 27, 29, 30], which is stated as

$$\tilde{\beta}_{\text{cmv}}(i) = [\tilde{\boldsymbol{g}}^{\text{H}}_{\text{cmv}}(i-1)\tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)]^{-1} \times [\tilde{\boldsymbol{g}}_{\text{cmv}}(i) - \tilde{\boldsymbol{g}}_{\text{cmv}}(i-1)]^{\text{H}}\tilde{\boldsymbol{g}}_{\text{cmv}}(i) \quad (20)$$

Until now, we derived the proposed MCG algorithm for the CMV criterion, whose weight solution is given by

$$\tilde{\boldsymbol{w}}_{\text{cmv}}(i) = [\boldsymbol{a}^{\text{H}}(\theta_0)\tilde{\boldsymbol{v}}_{\text{cmv}}(i)]^{-1}\gamma\tilde{\boldsymbol{v}}_{\text{cmv}}(i) \quad (21)$$

The proposed CMV-MCG algorithm is summarised in Fig. 3. We remove the subscript 'cmv' for compact expressions. Clearly, compared with (5), the weight solution in (21) ensures the constraint and solves the systems of equations without the matrix inversion and thus avoids numerical instability.

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

$$\tilde{\boldsymbol{g}}_{\text{ccm}}(i) = (1-\lambda_y)\boldsymbol{a}(\theta_0) - \tilde{\alpha}_{\text{ccm}}(i)\hat{\boldsymbol{R}}_y(i)\tilde{\boldsymbol{p}}_{\text{ccm}}(i) + \lambda_y\tilde{\boldsymbol{g}}_{\text{ccm}}(i-1) - e_y(i)\boldsymbol{x}(i)\boldsymbol{x}^{\text{H}}(i)\tilde{\boldsymbol{v}}_{\text{ccm}}(i-1) \quad (22)$$

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 $i = 1, \ldots, N$

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

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

$(0 \leq \tilde{\eta} \leq 0.5)$

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

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

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

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

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

$i = i + 1$

Figure 3 *CMV-MCG algorithm*

Initialization:

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

$\tilde{\boldsymbol{w}}_{\text{ccm}}(0) = \boldsymbol{a}(\theta_0)/\|\boldsymbol{a}(\theta_0)\|^2$

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

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

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

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

$(0 \leq \tilde{\eta} \leq 0.5)$

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

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

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

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

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

$i = i + 1$

Figure 4 *CCM-MCG algorithm*

Following the same derivation as for the CMV criterion, we will obtain the CCM-MCG algorithm, which is summarised in Fig. 4. Comparing with the CCG algorithm, the MCG is a non-reset and low-complexity algorithm with one iteration per snapshot. It achieves superior performance by utilising the Polak–Ribiere approach, as will be shown in simulations.

4 Analysis of the proposed algorithms

In this section, we investigate the convexity, convergence and complexity of the proposed algorithms.

4.1 Global convergence and properties

The CMV criterion, as can be seen in (3), is a second-order function. Its analysis has been proved in [13]. Here, we focus on the CCM criterion in (4), which is a fourth-order function with an elaborate structure of the undesired local minima. We show in Appendix 1 that the convexity of the CCM cost function can be enforced by properly selecting the coefficient γ. Therefore the global convergence for the constrained adaptive algorithms can be guaranteed.

4.2 Computational complexity

In this part, we detail the computational complexity of the proposed and analysed algorithms. The comparison of the complexity with respect to different algorithms is listed in Table 1, where r is the rank. 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. Note that, compared with the RLS method, the

Table 1 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	$r(4m^2 + m - 2) + 5m^2 - m - 1$	$r(5m^2 + 3m) + 8m^2 + 2m$

complexity of the MCG algorithm reduces significantly if m (e.g. $m = 60$) is large for some applications in sonar or radar. The reduction is not visible if m is small, for example, $m = 4$ for wireless communications [31]. However, the proposed algorithm achieves an efficient implementation for the beamformer design and avoids numerical instability that occurs in the RLS method.

4.3 Convergence analysis

Theoretically, the CG-type algorithms are completed after at most $K = m$ iterations for each snapshot. Actually, accumulated floating point roundoff errors in the iterations cause the residual (negative gradient) to gradually lose accuracy and destroy the conjugacy of the direction vectors [18, 25]. The convergence analysis is regarded as an explanation about this error. In this paper, the algorithms are proposed according to the CMV and the CCM criteria, respectively. They are not applied directly for the weight solutions. The convergence property in this part should be analysed in consideration of the above cases. The subscripts 'cmv' and 'ccm' will be removed in the following terms for compactness.

According to the CG theorem, 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 $\boldsymbol{g}_k(i)$ is a linear combination of the previous residuals and $\hat{\boldsymbol{R}}(i)\boldsymbol{p}_k(i)$. If defining $\mathcal{S}_k(i)$ as the subspace spanned by the direction vectors and recalling $\boldsymbol{p}_{k+1}(i) \in \mathcal{S}_k(i)$ imply that each new $\mathcal{S}_{k+1}(i)$ is formed from the previous $\mathcal{S}_k(i)$ and $\hat{\boldsymbol{R}}(i)\mathcal{S}_k(i)$, we have

$$\begin{aligned}\mathcal{S}_k(i) &= \text{span}\{\boldsymbol{p}_1(i), \hat{\boldsymbol{R}}(i)\boldsymbol{p}_1(i), \ldots, \hat{\boldsymbol{R}}^{k-1}(i)\boldsymbol{p}_1(i)\} \\ &= \text{span}\{\boldsymbol{g}_0(i), \hat{\boldsymbol{R}}(i)\boldsymbol{g}_0(i), \ldots, \hat{\boldsymbol{R}}^{k-1}(i)\boldsymbol{g}_0(i)\}\end{aligned} \tag{23}$$

As we know, the residual vector can be written as

$$\boldsymbol{g}_k(i) = \boldsymbol{a}(\theta_0) - \hat{\boldsymbol{R}}(i)\boldsymbol{v}_k(i) = \hat{\boldsymbol{R}}(i)\boldsymbol{\varrho}_k(i) \tag{24}$$

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 (24), the second expression in (23) is given by

$$\mathcal{S}_k(i) = \text{span}\{\hat{\boldsymbol{R}}(i)\boldsymbol{\varrho}_0(i), \hat{\boldsymbol{R}}^2(i)\boldsymbol{\varrho}_0(i), \ldots, \hat{\boldsymbol{R}}^k(i)\boldsymbol{\varrho}_0(i)\} \tag{25}$$

which is the well known Krylov subspace [18]. For a fixed k, this subspace holds an important property, which is

$$\boldsymbol{\varrho}_{k+1}(i) = \left(\boldsymbol{I} + \sum_{j=1}^{k} \psi_j(i)\hat{\boldsymbol{R}}^j(i)\right)\boldsymbol{\varrho}_0(i) \tag{26}$$

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 $l' = j, \ldots, k$. To prove this property, substituting $\alpha_k(i)$ and

$\mathbf{p}_{k+1}(i)$ into $\mathbf{v}_{k+1}(i)$ iteratively, we obtain

$$\begin{aligned}\mathbf{v}_{k+1}(i) &= \mathbf{v}_0(i) + \sum_{j=1}^{k+1} \alpha_j(i)\mathbf{p}_j(i) \\ &= \mathbf{v}_0(i) + \sum_{j=1}^{k+1} \alpha_j(i)[\mathbf{g}_{j-1}(i) + \beta_{j-1}(i)\mathbf{g}_{j-2}(i) + \cdots \\ &\quad + \beta_{j-1}(i)\cdots\beta_2(i)\mathbf{g}_1(i) + \beta_{j-1}(i)\cdots\beta_1(i)\mathbf{p}_1(i)] \\ &= \mathbf{v}_0(i) + \{L_{g_k}(i)\mathbf{g}_k(i) + L_{g_{k-1}}\mathbf{g}_{k-1}(i) + \cdots \\ &\quad + L_{g_1}(i)\mathbf{g}_1(i) + L_{p_1}(i)\mathbf{p}_1(i)\}\end{aligned} \tag{27}$$

where

$L_{g_k}(i) = \alpha_{k+1}(i), L_{g_{k-1}}(i) = \alpha_k(i) + \alpha_{k+1}(i)\beta_k(i),\ L_{g_1}(i) = \alpha_2(i) + \alpha_3(i)\beta_2(i) + \alpha_4(i)\beta_3(i)\,\beta_2(i) + \cdots + \alpha_{k+1}(i)\,\beta_k(i)\beta_{k-1}(i)\cdots\beta_2(i)$ and

$L_{p_1}(i) = \alpha_1(i) + \alpha_2(i)\beta_1(i) + \alpha_3(i)\beta_2(i)\beta_1(i) + \cdots + \alpha_{k+1}(i)\beta_k(i)\beta_{k-1}(i)\cdots\beta_1(i).$

In (27), the coefficients $L_{g_l}(i)$ for $l = 1, \ldots, k$ are constants and $\mathbf{p}_1(i) = \mathbf{g}_0(i)$. Thus, this implies that $\{\mathbf{g}_0(i), \ldots, \mathbf{g}_k(i)\} \in \mathcal{S}_k(i)$. Subtracting (27) from $\mathbf{v}_{\mathrm{opt}}(i)$ and combining the expressions in (24), (25) and (27), we obtain

$$\boldsymbol{\varrho}_{k+1}(i) = \boldsymbol{\varrho}_0(i) + \sum_{j=1}^{k} \psi_j(i)\hat{\mathbf{R}}^j(i)\boldsymbol{\varrho}_0(i) \tag{28}$$

where $\psi_j(i)$ has been defined in (26). Making a rearrangement leads to (26).

The importance of (26) is to measure the error energy norm $\|\boldsymbol{\varrho}_{k+1}(i)\|_{\hat{\mathbf{R}}(i)} = (\boldsymbol{\varrho}_{k+1}^{\mathrm{H}}(i)\hat{\mathbf{R}}(i)\boldsymbol{\varrho}_{k+1}(i))^{1/2}$ for the convergence analysis. The expression in parentheses of (26) can be written as a polynomial $P_k(\hat{\mathbf{R}}(i))$ of degree k [25]. Then, we have

$$\boldsymbol{\varrho}_{k+1}(i) = P_k(\hat{\mathbf{R}}(i))\boldsymbol{\varrho}_0(i) \tag{29}$$

where we require that $P_0(\hat{\mathbf{R}}(i)) = \mathbf{I}$ since the algorithm cannot converge at the initial step. $\boldsymbol{\varrho}_0(i)$ can be defined as a linear combination of distinct eigenvectors with respect to $\hat{\mathbf{R}}(i)$, which yields $\boldsymbol{\varrho}_0(i) = \sum_j \xi_j(i)\mathbf{z}_j(i)$, where $\xi_j(i)$ are scalars not all zero and j is the index that corresponds to the number of the eigenvectors $\mathbf{z}_j(i)$. If we notice that $P_k(\hat{\mathbf{R}}(i))\mathbf{z}_j(i) = P_k(\tau_j(i))\mathbf{z}_j(i)$ [25], where $\tau_j(i)$ is the eigenvalue corresponding to $\mathbf{z}_j(i)$, (29) can be expressed as

$$\boldsymbol{\varrho}_{k+1}(i) = \sum_j \xi_j(i)P_k(\tau_j(i))\mathbf{z}_j(i) \tag{30}$$

$$\|\boldsymbol{\varrho}_{k+1}(i)\|^2_{\hat{\mathbf{R}}(i)} = \sum_j \xi_j^2(i)P_k^2(\tau_j(i))\tau_j(i) \tag{31}$$

The proposed algorithm tries to find the polynomial $P_k(\tau_j(i))$ that minimises (31) for the convergence, which should be fast even with the worst eigenvector. Therefore, this implies

$$\begin{aligned}\|\boldsymbol{\varrho}_{k+1}(i)\|^2_{\hat{\mathbf{R}}(i)} &\le \min_{P_k} \max_{\tau(i)\in\Lambda(\hat{\mathbf{R}}(i))} P_k^2(\tau(i)) \sum_j \xi_j^2(i)\tau_j(i) \\ &= \min_{P_k} \max_{\tau(i)\in\Lambda(\hat{\mathbf{R}}(i))} P_k^2(\tau(i))\|\boldsymbol{\varrho}_0(i)\|^2_{\hat{\mathbf{R}}(i)}\end{aligned} \tag{32}$$

where $\Lambda(\hat{\mathbf{R}}(i))$ is the set of the eigenvalues with respect to $\hat{\mathbf{R}}(i)$. In order to analyse the connection between the error energy norm and the eigenvalues in (32), we employ the Chebyshev polynomials, which yields [25]

$$P_k(\tau(i)) = \frac{T_k[(\tau_{\max}(i) + \tau_{\min}(i) - 2\tau(i))/(\tau_{\max}(i) - \tau_{\min}(i))]}{T_k[(\tau_{\max}(i) + \tau_{\min}(i))/(\tau_{\max}(i) - \tau_{\min}(i))]} \tag{33}$$

where $T_k(\omega) = 1/2\big[(\omega + \sqrt{\omega^2 - 1})^k + (\omega - \sqrt{\omega^2 - 1})^k\big]$ denotes the Chebyshev polynomials of degree k. This polynomial obeys the oscillating property of Chebyshev polynomials on the domain $\tau_{\min}(i) \le \tau(i) \le \tau_{\max}(i)$ [24]. The derivation of (33) is given in Appendix 2. Since the maximum value of the numerator of (33) is 1, we substitute it into the error energy norm, which should have

$$\begin{aligned}\|\boldsymbol{\varrho}_{k+1}(i)\|_{\hat{\mathbf{R}}(i)} &= (\boldsymbol{\varrho}_{k+1}^{\mathrm{H}}(i)\hat{\mathbf{R}}(i)\boldsymbol{\varrho}_{k+1}(i))^{1/2} \\ &\le T_k\left(\frac{\tau_{\max}(i) + \tau_{\min}(i)}{\tau_{\max}(i) - \tau_{\min}(i)}\right)^{-1}\|\boldsymbol{\varrho}_0(i)\|_{\hat{\mathbf{R}}(i)} \\ &= T_k\left(\frac{\kappa(i) + 1}{\kappa(i) - 1}\right)^{-1}\|\boldsymbol{\varrho}_0(i)\|_{\hat{\mathbf{R}}(i)} \\ &= 2\left[\left(\frac{\sqrt{\kappa(i)} + 1}{\sqrt{\kappa(i)} - 1}\right)^k + \left(\frac{\sqrt{\kappa(i)} - 1}{\sqrt{\kappa(i)} + 1}\right)^k\right]^{-1} \\ &\quad \times \|\boldsymbol{\varrho}_0(i)\|_{\hat{\mathbf{R}}(i)}\end{aligned} \tag{34}$$

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

$$\|\boldsymbol{\varrho}_{k+1}(i)\|_{\hat{\mathbf{R}}(i)} \le 2\left(\frac{\sqrt{\kappa(i)} - 1}{\sqrt{\kappa(i)} + 1}\right)^k\|\boldsymbol{\varrho}_0(i)\|_{\hat{\mathbf{R}}(i)} \tag{35}$$

From (35), the CG-based weight vector converges following the iteration procedure. We know that $\mathbf{w}(i)$ is impacted by $\mathbf{g}_k(i)$ according to the proposed method. This implies the convergence of the proposed algorithm.

In conclusion, the convergence behaviour of the proposed algorithms 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, that is, the convergence is finished in one iteration. The convergence analysis is suitable to the CMV and the CCM criteria if we use the different matrix definitions $\hat{\mathbf{R}}(i)$ and $\hat{\mathbf{R}}_y(i)$.

doi: 10.1049/iet-spr.2009.0243

5 Simulation results

The simulations are carried out under both stationary and non-stationary scenarios for a ULA containing m sensor elements with half-wavelength spacing. For each experiment, 1000 runs are executed to obtain 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 (see Appendix 1 and (41)).

5.1 Comparison with the Krylov subspace methods

We compare the proposed algorithms with the SG [13], RLS [3], MSWF [6] and AVF [7] methods according to the CMV criterion by showing the output signal-to-interference-plus-noise ratio (SINR) against the input signal-to-noise ratio (SNR). The ULA is equipped with $m = 10$ sensor elements. We consider that the system has three interferers with equal power level of the desired user. The step size for the CMV-SG algorithm is set to $\mu_{\rm cmv} = 0.0002$. The iteration number is $K = m/2$ for the CCG algorithm. In Fig. 5, the output SINRs of the studied algorithms are very close to that obtained from (5) 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 [5, 6, 9].

5.2 SINR performance

Fig. 6 compares the proposed algorithms with the SG and RLS methods. There are five interferers in the system with

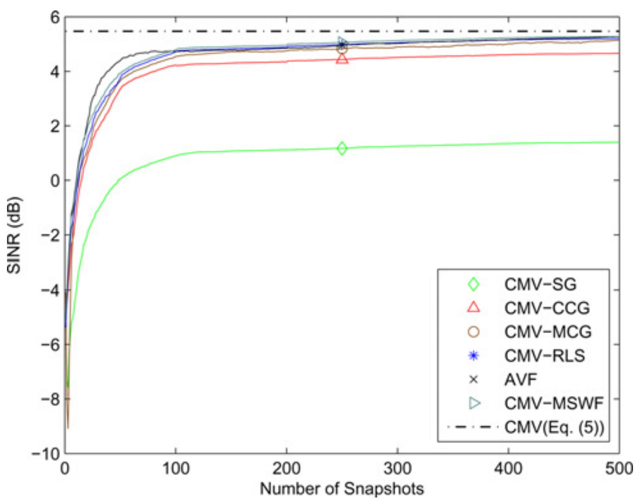

Figure 5 *Output SINR against the number of snapshots with $q = 4$, $m = 10$, SNR = 20 dB, $\lambda = 0.998$, $\tilde{\eta} = 0.49$, $\delta = 0.002$, $\tilde{\delta} = 0.001$ and $\mu_{cmv} = 0.0002$*

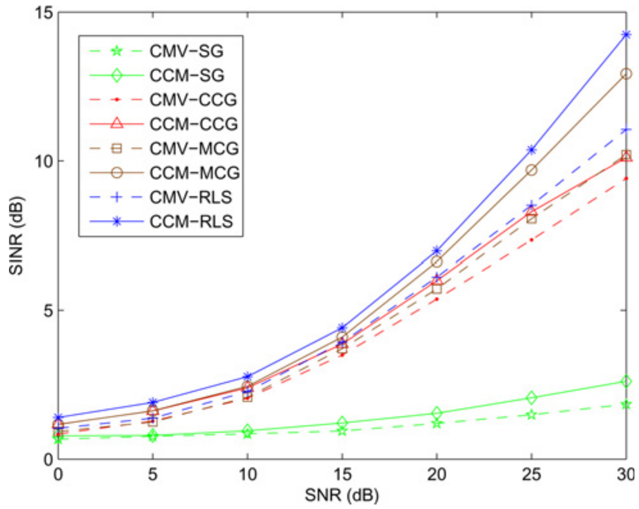

Figure 6 *Output SINR against input SNR with $q = 6$, $m = 16$, $\lambda = \lambda_y = 0.998$, $\tilde{\eta} = 0.49$, $\delta = \delta_y = 0.0018$, $\tilde{\delta} = 0.0015$, $\tilde{\delta}_y = 0.001$, $\mu_{cmv} = 0.0002$ and $\mu_{ccm} = 0.00015$*

one 5 dB, one 0 dB and three −0.5 dB above the desired power. It is observed that the SINRs of the RLS and proposed algorithms increase with higher input SNR, whereas the SG results show only a small improvement. The proposed MCG curves approach the RLS ones but with lower complexity. Also, the performance of the MCG algorithms is better than that of the CCG methods, which verifies the advantage of the Polak–Ribiere approach. Clearly, the adaptive algorithms for the CCM criterion achieve superior performance in contrast to those of the CMV criterion. For efficient presentation and convenience, we only illustrate the simulation results with respect to the CCM criterion in the following parts.

5.3 Performance against number of users

In this part, we illustrate the performance of the proposed algorithms with an increasing number of users. We consider the input SNR = INR = 20 dB. The fact that the number of the interferers increases deteriorates the output SINR of all the algorithms as shown in Fig. 7. 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 much closer to the RLS when compared with a small number of users, which shows that the proposed algorithms are robust in a severe environment.

5.4 Array beampatterns

Fig. 8 shows the beampatterns of the array with respect to the existing and proposed algorithms. The DOA of the desired user is $\theta_0 = 50^\circ$. There are five interferers with one 5 dB ($\theta_1 = 40^\circ$), one 0 dB ($\theta_2 = 70^\circ$), and three −5 dB ($\theta_3 = 20^\circ$, $\theta_4 = 30^\circ$ and $\theta_5 = 60^\circ$) above the desired power. The input SNR = 20 dB and the number of snapshot is $N = 1000$. From Fig. 8, the mainlobe beams

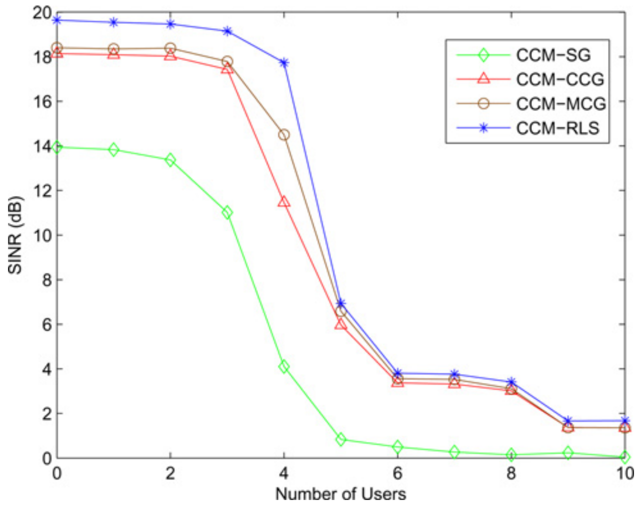

Figure 7 *Output SINR against the number of users (q) with $m = 16$, SNR = 20 dB, $\lambda_y = 0.998$, $\tilde{\eta} = 0.49$, $\delta_y = 0.002$, $\tilde{\delta}_y = 0.001$ and $\mu_{ccm} = 0.00015$*

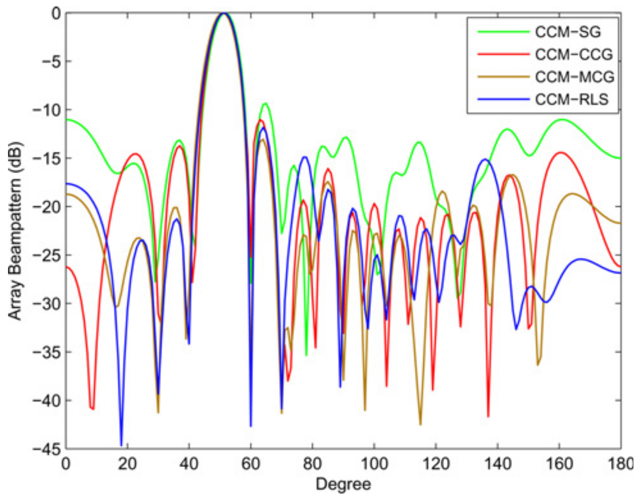

Figure 8 *Array beampattern against degree with $m = 16$, $q = 6$, SNR = 20 dB, $\lambda_y = 0.998$, $\tilde{\eta} = 0.49$, $\delta_y = 0.002$, $\tilde{\delta}_y = 0.0013$, $\mu_{ccm} = 0.00013$*

of the studied algorithms are directed at the direction of the desired user. The proposed algorithms have nulls at the DOAs of the interferers, especially for the MCG method, which forms the nulls as deep as those of the RLS method. Also, the noise level is suppressed sufficiently by the proposed algorithms, which is comparable to that of the RLS method and much lower than the SG method.

5.5 Performance with mismatch

The mismatch condition is analysed in Fig. 9, which includes two experiments. Fig. 9*a* shows the output SINR of each method against the number of snapshots with the known DOA of the desired user. The system works under the same condition as that in Fig. 6 with a defined SNR = 20 dB. The performance under the mismatch scenario is

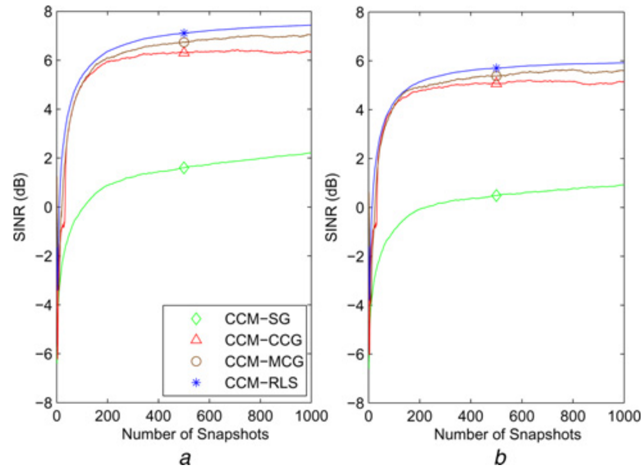

Figure 9 *Output SINR against the number of snapshots with $m = 16$, $q = 6$, SNR = 20 dB, $\mu_{ccm} = 0.00015$*

a Ideal steering vector condition
b Steering vector with mismatch 1°

given in Fig. 9*b*. The estimated DOA of the desired user is a constant value 1° 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 the RLS method, especially for the MCG algorithm, whose curve reaches the steady state rapidly and is very close to that of the RLS.

5.6 Performance in non-stationary scenarios

We evaluate the performance of the proposed and analysed algorithms in a non-stationary scenario. In Fig. 10, the

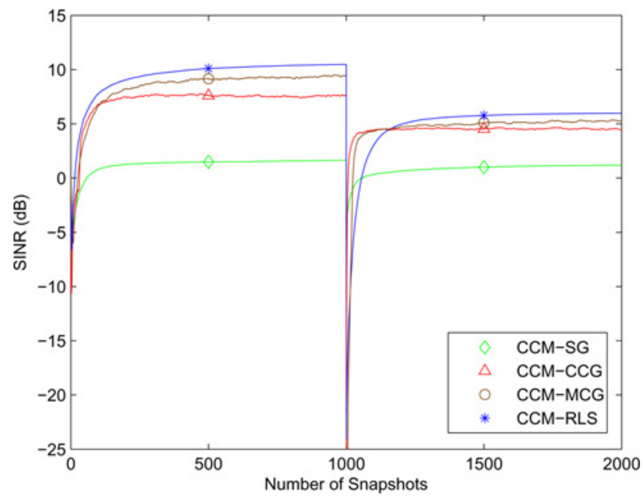

Figure 10 *Output SINR against the number of snapshots in a scenario where additional interferers suddenly enter and/or leave the system with $m = 16$, $q_1 = 6$, $q_2 = 8$, SNR = 20 dB, $\lambda_y = 0.998$, $\tilde{\eta} = 0.49$, $\delta_y = 0.003$, $\tilde{\delta}_y = 0.0025$, $\mu_{ccm} = 0.00016$*

IET Signal Process., 2010, Vol. 4, Iss. 6, pp. 686–697
doi: 10.1049/iet-spr.2009.0243

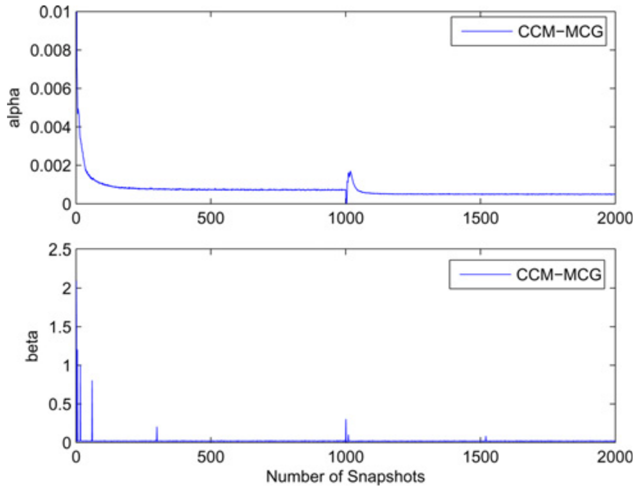

Figure 11 *Step size values α and β of the proposed CCM-MCG algorithm in a scenario where additional interferers suddenly enter and/or leave the system*

system starts with five interferers, one with 0 dB and the rest -0.5 dB above the desired power. Two more users with one 5 dB and one -0.5 dB above the desired power enter the system at the 1000th snapshot. We see that the SINRs of the algorithms reduce at the same time. The proposed algorithms rapidly track the change and recover to a steady state. The CCG algorithm exhibits a faster convergence rate at the beginning of the second stage since the reset procedure makes a good start for obtaining the fresh weight solution and so saves convergence time. 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. 11 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 coincides with the convergence, and for $\tilde{\beta}_{\text{ccm}}(i)$, according to $\tilde{\beta}_{\text{ccm}}(i) = [\tilde{\boldsymbol{g}}^{H}_{\text{ccm}}(i-1)\tilde{\boldsymbol{g}}_{\text{ccm}}(i-1)]^{-1}[\tilde{\boldsymbol{g}}_{\text{ccm}}(i) - \tilde{\boldsymbol{g}}_{\text{ccm}}(i-1)]^{H} \; \tilde{\boldsymbol{g}}_{\text{ccm}}(i)$, 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 as the extra interferers come into the system, which verifies the adaptability of the coefficients.

6 Conclusions

This paper proposed CG-based adaptive algorithms with respect to the CMV and CCM criteria for adaptive filtering with application to beamforming. We created a connection between the CG-based weight vector and the weight expressions of the design criteria to enforce the constraint and avoid the matrix inversion and numerical instability. 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 proposed algorithms was derived. Simulation results showed that the proposed algorithms achieve comparable fast convergence and tracking abilities with low complexity in the studied scenarios.

7 Acknowledgment

This work is supported in part by the Department of Electronics, the University of York.

8 References

[1] LIBERTI J.C., RAPPAPORT T.S.: 'Smart antennas for wireless communications: IS-95 and third generation CDMA applications' (Prentice-Hall, 1999)

[2] VERDU S.: 'Multiuser detection' (Cambridge University Press, 1998)

[3] HAYKIN S.: 'Adaptive filter theory' (Prentice-Hall, Englewood Cliffs, NJ, 1996, 4th edn.)

[4] BORAY G.K., SRINATH M.D.: 'Conjugate gradient techniques for adaptive filtering', *IEEE Trans. Circuits Syst. I*, 1992, **39**, pp. 1–10

[5] HONIG M.L., GOLDSTEIN J.S.: 'Adaptive reduced-rank interference suppression based on the multistage wiener filter', *IEEE Trans. Commun.*, 2002, **50**, pp. 986–994

[6] DE LAMARE R.C., HAARDT M., SAMPAIO-NETO R.: 'Blind adaptive constrained reduced-rank parameter estimation based on constant modulus design for CDMA interference suppression', *IEEE Trans. Signal Proc.*, 2008, **56**, pp. 2470–2482

[7] PADOS D.A., KARYSTINOS G.N.: 'An iterative algorithm for the computation of the MVDR filter', *IEEE Trans. Signal Process.*, 2001, **49**, pp. 290–300

[8] PADOS D.A., LOMBARDO F.J., BATALAMA S.N.: 'Auxiliary-vector filters and adaptive steering for DS/CDMA single-user detection', *IEEE Trans. Veh. Technol.*, 1999, **48**, pp. 1831–1839

[9] BURYKH S., ABED-MERAIM K.: 'Reduced-rank adaptive filtering using Krylov subspace', *EURASIP J. Appl. Signal Process.*, 2002, **2002**, pp. 1387–1400

[10] BOSE T., CHEN M.Q.: 'Conjugate gradient method in adaptive bilinear filtering', *IEEE Trans. Signal Process.*, 1995, **43**, pp. 1503–1508

[11] CHANG P.S., WILLSON A.N.: 'Adaptive filtering using modified conjugate gradient'. Proc. 38th Midwest Symp. on Circuits Systems, Rio de Janeiro, Brazil, August 1995, pp. 243–246

[12] APOLINÁRIO J.A., DE CAMPOS M.L.R., BERNAL C.P.: 'The constrained conjugate gradient algorithm', *IEEE Lett. Signal Process.*, 2000, **7**, pp. 351–354

[13] FROST O.L.: 'An algortihm for linearly constrained adaptive array processing', *IEEE Proc.*, 1972, **AP-30**, pp. 27–34

[14] CHEN Y., LE-NGOC T., CHAMPAGNE B., XU C.: 'Recursive least squares constant modulus algorithm for blind adaptive array', *IEEE Trans. Signal Process.*, 2004, **52**, pp. 1452–1456

[15] XU Z.Y., LIU P.: 'Code-constrained blind detection of CDMA signals in multipath channels', *IEEE Signal Process. Lett.*, 2002, **9**, pp. 389–392

[16] DE LAMARE R.C., SAMPAIO NETO R.: 'Blind adaptive code-constrained constant modulus algorithms for CDMA interference suppression in multipath channels', *IEEE Commun. Lett.*, 2005, **9**, pp. 334–336

[17] LI J., STOICA P.: 'Robust adaptive beamforming' (Wiley, Hoboken, NJ, 2006)

[18] GOLUB G.H., VAN LOAN C.F.: 'Matrix comuputations' (Johns Hopkins University Press, Baltimore, MD, 1996, 3rd edn.)

[19] TREES H.V.: 'Optimum array processing: part IV of detection, estimation, and modulation theory' (John Wiley & Sons, 2002)

[20] GODARD D.N.: 'Self-recovering equalization and carrier tracking in two-demensional data communication systems', *IEEE Trans. Commun.*, 1980, **28**, pp. 1867–1875

[21] COX H., ZESKIND R.M., OWEN M.M.: 'Robust adaptive beamforming', *IEEE Trans. Acoust. Speech Signal Process.*, 1987, **35**, pp. 1365–1376

[22] WEIPPERT M.E., HIEMSTRA J.D., GOLDSTEIN J.S., ZOLTOWSKI M.D.: 'Insights from the relationship between the multistage Wiener filter and the method of conjugate gradients'. Proc. IEEE Workshop on Sensor Array and Multichannel Signal Processing, August 2002, pp. 388–392

[23] SCHARF L.L., CHONG E.K.P., ZOLTOWSKI M.D., GOLDSTEIN J.S., REED I.S.: 'Subspace expansion and the equivalence of conjugate direction and multistage wiener filters', *IEEE Trans. Signal Process.*, 2008, **56**, pp. 5013–5019

[24] LUENBERGER D.G.: 'Linear and nonlinear programming' (Addison-Wesley, Reading, MA, 1984, 2nd edn.)

[25] WATKINS D.S.: 'Fundamentals of matrix computations', (John Wiley & Sons, New York, 2nd. edn.)

[26] WHITEHEAD J.B., TAKAWIRA F.: 'Performance analysis of the linearly constrained constant modulus algorithm-based multiuser detector', *IEEE Trans. Signal Process.*, 2005, **53**, pp. 643–653

[27] CHANG P.S., WILLSON A.N.: 'Analysis of conjugate gradient algorithms for adaptive filtering', *IEEE Trans. Signal Process.*, 2000, **48**, pp. 409–418

[28] AL-BAALI M.: 'Descent property and global convergence of the Fletcher–Reeves method with inexact line search', *IMA J. Numer. Anal.*, 1985, **5**, pp. 121–124

[29] FLETCHER R.: 'Practical methods of optimization' (Wiley, Chichester, UK, 1987, 2nd edn.)

[30] SHANNO D.F.: 'Conjugate gradient methods with inexact searches', *Math. Oper. Res.*, 1978, **3**, pp. 244–256

[31] LITVA J., LO T.K.: 'Digital beamforming in wireless communications' (Artech House, 1996)

[32] XU C.J., FENG G.Z., KWAK K.S.: 'A modified constrained constant modulus approach to blind adaptive multiuser detection', *IEEE Trans. Commun.*, 2001, **49**, pp. 1642–1648

[33] MASON J.C., HANDSCOMB D.C.: 'Chebyshev polynomials' (Chapman Hall/CRC, 2002, 1st edn.)

[34] HAMMING R.W.: 'Numerical methods for scientists and engineers' (Courier Dover Publications, 1986, 2nd edn.)

9 Appendix 1: convexity condition for the CCM criterion

We consider cost function (4), which can be written as

$$\begin{aligned} J_{\text{cm}} &= E[|y(i)|^4 - 2|y(i)|^2 + 1] \\ &= E[|\boldsymbol{w}^{\text{H}}(i)\boldsymbol{x}(i)\boldsymbol{x}^{\text{H}}(i)\boldsymbol{w}(i)|^2] - 2E[|\boldsymbol{w}^{\text{H}}(i)\boldsymbol{x}(i)|^2] + 1 \end{aligned} \quad (36)$$

where $\delta_p = 1$ and $\boldsymbol{x}(i) = \sum_{k=0}^{q-1} B_k d_k \boldsymbol{a}(\theta_k) + \boldsymbol{n}(i)$ from (1) with B_k being the signal amplitude and d_k is the transmitted bit of the kth user ($k = 0, \ldots, q-1$), respectively. Note that we have replaced s_k in (1) by $B_k d_k$. For the sake of analysis, we will follow the assumption in [32] 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 (4) convex. Under this assumption, we write the input vector as $\boldsymbol{x}(i) = \boldsymbol{ABd}(i)$, where $\boldsymbol{B} = \text{diag}[B_0, \ldots, B_{q-1}]$ and $\boldsymbol{d}(i) = [d_0(i), \ldots, d_{q-1}(i)]^{\text{T}}$. For simplicity, we remove the time instant i in the quantities. Letting $r_k = B_k \boldsymbol{w}^{\text{H}} \boldsymbol{a}(\theta_k)$ and $\boldsymbol{r} = [r_0, \ldots, r_{q-1}]^{\text{T}}$, (36) can be written as

$$J_{\text{cm}} = E[\boldsymbol{r}^{\text{H}}\boldsymbol{dd}^{\text{H}}\boldsymbol{rr}^{\text{H}}\boldsymbol{dd}^{\text{H}}\boldsymbol{r}] - 2E[\boldsymbol{r}^{\text{H}}\boldsymbol{dd}^{\text{H}}\boldsymbol{r}] + 1 \quad (37)$$

Since d_k are independent random variables, the evaluation of

doi: 10.1049/iet-spr.2009.0243

the first two terms in the brackets of (37) reads

$$\boldsymbol{r}^{\mathrm{H}}\boldsymbol{d}\boldsymbol{d}^{\mathrm{H}}\boldsymbol{r}\boldsymbol{r}^{\mathrm{H}}\boldsymbol{d}\boldsymbol{d}^{\mathrm{H}}\boldsymbol{r} = \sum_{k=0}^{q-1}\sum_{j=0}^{q-1}|d_k|^2|d_j|^2 r_k^* r_k r_j^* r_j$$
$$\boldsymbol{r}^{\mathrm{H}}\boldsymbol{d}\boldsymbol{d}^{\mathrm{H}}\boldsymbol{r} = \sum_{k=0}^{q-1}|d_k|^2 r_k^* r_k \tag{38}$$

Substituting (38) into (37) and using the constrained condition $\boldsymbol{w}^{\mathrm{H}}\boldsymbol{a}(\theta_0) = \gamma$, we have

$$J_{\mathrm{ccm}} = E[|d_0|^2 B_0^2\gamma^2 + \bar{\boldsymbol{r}}^{\mathrm{H}}\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^{\mathrm{H}}\bar{\boldsymbol{r}}]^2 - 2E[|d_0|^2 B_0^2\gamma^2 + \bar{\boldsymbol{r}}^{\mathrm{H}}\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^{\mathrm{H}}\bar{\boldsymbol{r}}] + 1 \tag{39}$$

where d_0 and B_0 denote the transmitted bit and amplitude relevant to the desired signal, $\bar{\boldsymbol{d}} = [d_1, \ldots, d_{q-1}]^{\mathrm{T}}$ and $\bar{\boldsymbol{r}} = [r_1, \ldots, r_{q-1}]^{\mathrm{T}}$. To examine the convexity property of (39), we compute the Hessian $\boldsymbol{H}$ with respect to $\bar{\boldsymbol{r}}^{\mathrm{H}}$ and $\bar{\boldsymbol{r}}$, that is $\boldsymbol{H} = (\partial/\partial\bar{\boldsymbol{r}}^{\mathrm{H}})(\partial J_{\mathrm{ccm}}/\partial\bar{\boldsymbol{r}})$ yields

$$\boldsymbol{H} = 2E[(|d_0|^2 B_0^2\gamma^2 - 1)\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^{\mathrm{H}} + \bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^{\mathrm{H}}\bar{\boldsymbol{r}}\bar{\boldsymbol{r}}^{\mathrm{H}}\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^{\mathrm{H}} + \bar{\boldsymbol{r}}^{\mathrm{H}}\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^{\mathrm{H}}\bar{\boldsymbol{r}}\bar{\boldsymbol{d}}\bar{\boldsymbol{d}}^{\mathrm{H}}] \tag{40}$$

where $\boldsymbol{H}$ should be positive semi-definite to ensure the convexity of the optimisation problem. The second and third terms in (40) 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 $\boldsymbol{w}$, that is, $\bar{\boldsymbol{r}} = \boldsymbol{C}\boldsymbol{w}$, where $\boldsymbol{B}' = \mathrm{diag}(B_1, \ldots, B_{q-1}) \in \mathcal{R}^{(q-1)\times(q-1)}$, $\boldsymbol{A}' = [\boldsymbol{a}\ (\theta_1), \ldots, a(\theta_{q-1})] \in \mathcal{C}^{m\times(q-1)}$ and $\boldsymbol{C} = \boldsymbol{B}'^{\mathrm{H}}\boldsymbol{A}'^{\mathrm{H}} \in \mathcal{C}^{(q-1)\times m}$. This expression shows that $J_{\mathrm{ccm}}(\boldsymbol{w})$ is a convex function of $\boldsymbol{w}$ as $J_{\mathrm{ccm}}(\bar{\boldsymbol{r}}) = J_{\mathrm{ccm}}(\boldsymbol{C}\boldsymbol{w})$ is of $\bar{\boldsymbol{r}}$ when

$$\gamma^2 \geq \frac{1}{|d_0|^2 B_0^2} \tag{41}$$

The optimisation problem is convex if the condition in (41) is satisfied. Note that this condition holds for all constant modulus constellations.

10 Appendix 2: derivation of (33)

We employ Chebyshev polynomials [33] for minimisation of (32) since they increase in magnitude more quickly outside the range $[-1, 1]$ than any other polynomial that is restricted to have magnitude less or equal to one inside this range. The Chebyshev polynomials of degree k, $T_k(\omega) = 1/2[(\omega + \sqrt{\omega^2 - 1})^k + (\omega - \sqrt{\omega^2 - 1})^k]$ can be written on the region $[-1, 1]$ as [34]

$$T_k(\omega) = \cos(k \cos^{-1}\omega), \quad -1 \leq \omega \leq 1 \tag{42}$$

in which we deduce that the Chebyshev polynomials hold $|T_k(\omega)| \leq 1$ for $-1 \leq \omega \leq 1$ and oscillate rapidly on the region $[-1, 1]$

$$T_k(\cos(l\pi/k)) = (-1)^l, \quad l = 0, 1, \ldots, k \tag{43}$$

where it is clear that the k zeros of T_k must fall between the $k + 1$ extrema of T_k in the range $[-1, 1]$.

Similarly, function (33) oscillates in the range $\pm T_k[(\tau_{\max}(i) + \tau_{\min}(i))/(\tau_{\max}(i) - \tau_{\min}(i))]^{-1}$ on the domain $[\tau_{\min}(i), \tau_{\max}(i)]$. Note that $P_k(\tau(i))$ still keeps the requirement that $P_k(0) = 1$.

Furthermore, it is important to show that there is no other polynomial better than P_k in the range $[\tau_{\min}(i), \tau_{\max}(i)]$. We use the contradiction way to prove this fact. Suppose that there is such a polynomial Q_k that satisfies $Q_k(\tau(i)) < T_k[(\tau_{\max}(i) + \tau_{\min}(i))/(\tau_{\max}(i) - \tau_{\min}(i))]^{-1}$ on the range $[\tau_{\min}(i), \tau_{\max}(i)]$. It follows that the polynomial $P_k - Q_k$ has a zero at $\tau(i) = 0$ and also k zeros on the range $[\tau_{\min}(i), \tau_{\max}(i)]$. In other words, $P_k - Q_k$ is a polynomial of degree k with at least $k + 1$ zeros, which is impossible. Hence, we conclude that the Chebyshev polynomial of degree k optimises (32).

doi: 10.1049/iet-spr.2009.0243