Separable Linearly Constrained Minimum Variance Beamformers[⋆]

Lucas N. Ribeiro[a,*], André L. F. de Almeida[a], João César M. Mota[a]

[a] *Wireless Research Telecommunications Group (GTEL), Universidade Federal do Ceará,
Campus do Pici, s/n, Bloco 722, CP 6005, 60455-760, Fortaleza, Brazil*

Abstract

Large-scale antenna systems offer many attractive features, including large array gain and improved spatial resolution, for example. However, classical beamforming methods, such as the linearly constrained minimum variance (LCMV) filter, do not perform well on this scenario due to large computational costs involved. To deal with this issue, we propose Kronecker-separable extensions of the LCMV filter and its stochastic gradient implementation, known as Frost's algorithm, for uniform rectangular arrays. We study the convergence of the proposed methods, investigate their computational complexity, and assess their source recovery performance with computer simulations. Our results show that our methods exhibit important computational savings while the source recovery performance losses are small.

Keywords: Beamforming, LCMV, stochastic gradient, Kronecker product, Khatri-Rao product.

1. Introduction

Large-scale (massive) antenna arrays will be employed in new generation wireless communication systems to increase performance by large array gain and accurate spatial filtering [1, 2]. To this end, the array antennas are weighted by complex coefficients which coherently combines (and possibly attenuates) the received (transmitted) signals from (towards) some directions. This beamforming technique is well-understood for cases where the array has a relatively small number of antennas. However, when the array size grows, classical beamforming

[⋆] This work has been supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), CAPES/PROBRAL Process no. 88887.144009/2017-00, CAPES/COFECUB Process no. 830/15, and the Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP).

[*] Corresponding author

Email addresses: nogueira@gtel.ufc.br (Lucas N. Ribeiro), andre@gtel.ufc.br (André L. F. de Almeida), mota@gtel.ufc.br (João César M. Mota) techniques, such as those discussed in [3, 4], become computationally expensive, calling for the development of more efficient solutions.

The first attempts to reduce the computational complexity of filtering methods can be traced back to some decades ago. One can cite the work of Treitel [5] which presented a low-dimensional multi-stage representation for bi-dimensional filters based on the eigendecomposition of the coefficient matrix, providing computational savings. However, the computation of large coefficient matrix is still expensive in general. In the context of massive multiple-input multiple-output systems [1], zero-forcing (ZF)/minimum mean square error (MMSE) detection schemes typically involve matrix inversions, which are particularly expensive at massive systems. To avoid that, some works proposed approximating the matrix inverse by the Neumann series expansion [6, 7]. Unfortunately, this approximation may not always converge and its overall computational complexity might still be comparable to matrix inversion, owing to a large number of matrix multiplications. As an alternative, other researchers suggested calculating these linear detectors by iterative algorithms, such as the Jacobi method [8, 9]. However, such methods may still suffer from slow convergence rate and high overall computational complexity. Another strategy to reduce the number of calculations of filtering methods consists of exploiting any special structure the filter may present. For instance, the array factor of multi-dimensional arrays can be factorized (separated) into components corresponding to each array dimension [10, 11]. From this property, the steering vector of multi-dimensional arrays assumes a Kronecker factorization which can be exploited to enhance the performance of array processing techniques.

Kronecker separable systems have been investigated in the context of system identification [12, 13, 14], beamforming [15, 16], and echo cancellation [17, 18]. In [12], the authors introduced a tensor least mean squares (TLMS) algorithm to identify a bilinear separable system. They observed that this representation reduces the number of parameters to be identified, increasing the convergence rate of the stochastic gradient. In [14], we extended the model of [12] to a trilinear separable system and proposed the trilinear Wiener-Hopf (TWH) algorithm to the system impulse response. The results have shown that the presented method exhibits better accuracy compared to the TLMS algorithm and the classical Wiener-Hopf filter. In [15, 16], we introduced the tensor minimum mean square error (TMMSE) beamformer, which applies the methodology of [14] in the MMSE beamforming problem. Furthermore, we proposed the Kronecker MMSE (KMMSE) beamformer, which exploits the Khatri-Rao structure of the uniform rectangular array (URA) manifold matrix. We have shown through computer simulations that the TMMSE and KMMSE filters offer significant computational savings with a controlled source recovery performance loss. We can also mention the work of [17], which adapts the TWH to identify a low-rank system, i.e., a linear and time-invariant system whose impulse response can be approximated as a sum of Kronecker-separable terms. Presented numerical results confirm that imposing the Kronecker structure on low-rank systems allows for computational complexity reduction and parameter estimation accuracy enhancement. In [18], fast recursive least squares methods for identifying second-order Kronecker separable (bilinear) systems are presented. Analytical and simulation results confirm the low computational costs and the identification performance of the proposed bilinear methods.

Our contributions [15, 16] have considered only the MMSE criterion, which requires pilot sequences to design the parameters of the beamforming filter. However, in some applications, this supervised design approach is unavailable. As alternatives to the MMSE filter, one can use other strategies such as the minimum variance distortionless response filter (MVDR) and the global sidelobe canceller (GSC) [10]. Separable extensions of these methods have already been presented. In [19], the authors investigated the performance of a Tensor GSC and observed that it requires fewer snapshots to achieve a certain performance level, compared to the classical GSC filter. A tensor MVDR algorithm has been introduced in [20] for polarization sensitive arrays.

In the present work, we introduce separable extensions of the linear constrained minimum variance (LCMV) beamformer and its stochastic gradient implementation known as the Frost's algorithm [21] for a URA. The proposed methods can be applied to any separable array structure, including uniform linear arrays [15, 22] and other multi-scale planar array structures [23]. The main contributions of this paper are: (i) we present two strategies for designing separable LCMV- and Frost-type beamformers, (ii) we conduct a computational analysis of the proposed algorithms, (iii) we prove their convergence, and (iv) we assess their performance through computer simulations. Our simulation results show that the proposed separable methods are more computationally efficient than their classical counterparts without presenting significant signal recovery loss. More specifically, the proposed separable LCMV-type beamformers exhibit an important reduction in the number of multiplications relative to the classical LCMV filter. The signal recovery performance of our methods may be approximately equivalent or much better, depending on the scenario and implementation type, as it will be discussed. Furthermore, the separable Frost-type beamformers require less computational resources than their classical counterpart and they yield better signal recovery performance, especially in the low signal-to-noise ratio (SNR) regime.

This work is organized as follows: the system model is introduced in Section 2 and the proposed beamforming methods are presented in Section 3. Therein, we also discuss their computational complexity. Simulation results are shown and discussed in Section 4, and the work is concluded in Section 5.

1.1. Notation

$\boldsymbol{x}$ denotes vectors, and $\boldsymbol{X}$ matrices. The (i,j)-th entry of $\boldsymbol{X}$ is written as $[\boldsymbol{X}]_{i,j}$. The transpose, and the conjugate transpose (Hermitian) of $\boldsymbol{X}$ are denoted by $\boldsymbol{X}^{\mathsf{T}}$, $\boldsymbol{X}^{\mathsf{H}}$, respectively. The $(M \times M)$-dimensional identity matrix is represented by $\boldsymbol{I}_M$ and the $(M \times N)$-dimensional null matrix by $\boldsymbol{0}_{M\times N}$. The ℓ_2 norm, the statistical expected value operator, the vectorization operator, and the matrix trace are respectively denoted as $\|\cdot\|_2$, $\mathbb{E}[\cdot]$, $\mathrm{vec}(\cdot)$, $\mathrm{Tr}(\cdot)$. The Kronecker product, the Khatri-Rao product, the elementwise product, and the Big-O notation are referred to as $\otimes$, $\diamond$, $\odot$, and $O(\cdot)$, respectively.

2. System Model

Let us consider a multi-antenna system equipped with a URA of $N = N_h N_v$ omni-directional antennas, with N_h columns and N_v rows along the yz plane, as depicted in Figure 1. The antenna array is designed to operate at wavelength λ and the inter-element spacing in both horizontal and vertical directions is $\lambda/2$.

We assume that R narrow-band wavefronts with the same wavelength λ in the far-field propagation impinges onto the URA from directions (ϕ_r, θ_r), $r = 1, \ldots, R$. These wavefronts carry mutually independent digitally-modulated signals with zero mean and variance $\sigma^2_{s,r}$, $r = 1, \ldots, R$. The discrete-time model for the received signal at the n-th antenna at instant k is then given by:

$$x_n[k] = \sum_{r=1}^{R} a_n(\phi_r, \theta_r) s_r[k] + b_n[k],$$

where $a_n(\phi_r, \theta_r)$ denotes the array response to the r-th wavefront at the n-th antenna, $s_r[k]$ the complex-envelope of the digitally-modulated symbol, and $b_n[k]$ the complex additive white Gaussian sensor noise (AWGN) with zero mean and variance σ_b^2. Since the rectangular array is uniformly spaced with half-wavelength spacing, the array response can be written as [10]

$$a_n(\phi_r, \theta_r) = e^{j\pi[(n_h-1)\sin\phi_r \sin\theta_r + (n_v-1)\cos\theta_r]}. \tag{1}$$

with $n = n_h + (n_v - 1)N_h$, $n_h \in \{1, \ldots, N_h\}$, $n_v \in \{1, \ldots, N_v\}$. For notation simplicity, we parameterize the array response by the horizontal and vertical direction cosines $p_r = \sin\phi_r \sin\theta_r$ and $q_r = \cos\theta_r$, respectively. Let us define the array steering vector $\boldsymbol{a}(p_r, q_r) = [a_1(p_r, q_r), \ldots, a_N(p_r, q_r)]^{\mathsf{T}}$ and symbols vector $\boldsymbol{s}[k] = [s_1[k], \ldots, s_R[k]]^{\mathsf{T}}$. Using matrix notation, the received signals can be expressed as

$$\boldsymbol{x}[k] = [x_1[k], \ldots, x_N[k]]^{\mathsf{T}} = \boldsymbol{A}\boldsymbol{s}[k] + \boldsymbol{b}[k], \tag{2}$$

where $\boldsymbol{A} = [\boldsymbol{a}(p_1, q_1), \ldots, \boldsymbol{a}(p_R, q_R)] \in \mathbb{C}^{N \times R}$ denotes the array manifold matrix, and $\boldsymbol{b}[k] = [b_1[k], \ldots, b_N[k]]^{\mathsf{T}}$ the AWGN vector. Note that the model (2) is valid only for a specific angular range where the antenna response is considered to be omni-directional.

The array response (1) can be *separated* into horizontal and vertical contributions due to the URA bi-dimensionality [10]. More specifically, (1) can be factorized as

$$a_n(p_r, q_r) = a_{n_h}(p_r) a_{n_v}(q_r), \tag{3}$$

where $a_{n_h}(p_r) = e^{j\pi(n_h-1)p_r}$ and $a_{n_v}(q_r) = e^{j\pi(n_v-1)q_r}$. The sub-array steering vectors are then defined as

$$\boldsymbol{a}_h(p_r) = [a_1(p_r), \ldots, a_{N_h}(p_r)]^{\mathsf{T}} \in \mathbb{C}^{N_h},$$
$$\boldsymbol{a}_v(q_r) = [a_1(q_r), \ldots, a_{N_v}(q_r)]^{\mathsf{T}} \in \mathbb{C}^{N_v}.$$

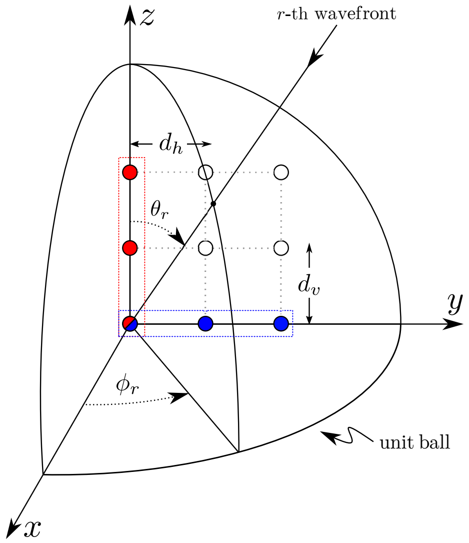

Figure 1: Uniform rectangular array in the yz plane.

The horizontal and vertical sub-arrays of a URA are depicted in Figure 1. The separable representation in (3) leads to the Kronecker factorization of the array steering vectors:

$$\boldsymbol{a}(p_r, q_r) = \boldsymbol{a}_v(q_r) \otimes \boldsymbol{a}_h(p_r) \in \mathbb{C}^N, \tag{4}$$

and, consequently, the array manifold matrix can be rewritten as

$$\boldsymbol{A} = [\boldsymbol{a}_v(q_1) \otimes \boldsymbol{a}_h(p_1), \ldots, \boldsymbol{a}_v(q_R) \otimes \boldsymbol{a}_h(p_R)] = \boldsymbol{A}_v \diamond \boldsymbol{A}_h,$$

where

$$\begin{aligned} \boldsymbol{A}_h &= [\boldsymbol{a}_h(p_1), \ldots, \boldsymbol{a}_h(p_R)] \in \mathbb{C}^{N_h \times R}, \\ \boldsymbol{A}_v &= [\boldsymbol{a}_v(q_1), \ldots, \boldsymbol{a}_v(q_R)] \in \mathbb{C}^{N_v \times R} \end{aligned}$$

stand for the vertical and horizontal sub-array manifold matrices, respectively. Other useful representations of the array manifold are obtained by reshaping the $(N \times R)$-dimensional matrix $\boldsymbol{A}$ into

$$\boldsymbol{A}_{\text{URA,h}} = \left[\boldsymbol{a}_v^{\mathsf{T}}(q_1) \otimes \boldsymbol{a}_h(p_1), \ldots, \boldsymbol{a}_v^{\mathsf{T}}(q_R) \otimes \boldsymbol{a}_h(p_R)\right] \in \mathbb{C}^{N_h \times N_v R} \tag{5}$$

and

$$\boldsymbol{A}_{\text{URA,v}} = \left[\boldsymbol{a}_v(q_1) \otimes \boldsymbol{a}_h^{\mathsf{T}}(p_1), \ldots, \boldsymbol{a}_v(q_R) \otimes \boldsymbol{a}_h^{\mathsf{T}}(p_R)\right] \in \mathbb{C}^{N_v \times N_h R}. \tag{6}$$

We have shown in [16] that, using (5) and (6), the received signals vector $\boldsymbol{x}[k]$ can be reshaped into

$$\boldsymbol{X}[k] = \boldsymbol{A}_{\text{URA,h}}(\boldsymbol{s}[k] \otimes \boldsymbol{I}_{N_v}) + \boldsymbol{B}[k] \in \mathbb{C}^{N_h \times N_v} \tag{7}$$

and

$$\boldsymbol{X}^{\mathsf{T}}[k] = \boldsymbol{A}_{\mathrm{URA,v}}(\boldsymbol{s}[k] \otimes \boldsymbol{I}_{N_h}) + \boldsymbol{B}^{\mathsf{T}}[k] \in \mathbb{C}^{N_v \times N_h}, \tag{8}$$

where $\boldsymbol{B}[k] \in \mathbb{C}^{N_h \times N_v}$ is the matrix obtained by reshaping the AWGN vector $\boldsymbol{b}[k]$. Equations (4)–(8) emphasize the separable property of the URA in our system model. These models will be leveraged to design low-cost beamformers in Section 3.

Following our assumption, the spatial correlation matrices of the AWGN term, symbols, and received signals vectors are respectively given by

$$\begin{aligned}
\boldsymbol{R}_{bb} &= \mathbb{E}\left[\boldsymbol{b}[k]\boldsymbol{b}[k]^{\mathsf{H}}\right] = \sigma_b^2 \boldsymbol{I}_N, \\
\boldsymbol{R}_{ss} &= \mathbb{E}\left[\boldsymbol{s}[k]\boldsymbol{s}[k]^{\mathsf{H}}\right] = \mathrm{Diag}\left(\sigma_{s,1}^2, \sigma_{s,2}^2, \ldots, \sigma_{s,R}^2\right), \\
\boldsymbol{R}_{xx} &= \mathbb{E}\left[\boldsymbol{x}[k]\boldsymbol{x}[k]^{\mathsf{H}}\right] = \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^{\mathsf{H}} + \boldsymbol{R}_{bb}.
\end{aligned} \tag{9}$$

In practice, the correlation matrix of the observed signals is probably unknown and need thus to be estimated. In this case, we consider the following sample estimate over K snapshots:

$$\hat{\boldsymbol{R}}_{xx} = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{x}[k]\boldsymbol{x}^{\mathsf{H}}[k] \tag{10}$$

Hereafter, we assume without loss of generality that the first wavefront ($r = 1$) is the signal of interest while the other $R-1$ signals are regarded as interference. The desired symbols are now referred to as $s_d[k]$ with variance σ_d^2. We can further express the received signals correlation matrix as

$$\begin{aligned}
\boldsymbol{R}_{xx} &= \sigma_d^2 \boldsymbol{a}(p_d, q_d)\boldsymbol{a}(p_d, q_d)^{\mathsf{H}} + \sum_{r>1}^{R} \sigma_{s,r}^2 \boldsymbol{a}(p_r, q_r)\boldsymbol{a}(p_r, q_r)^{\mathsf{H}} + \boldsymbol{R}_{bb} \\
&= \boldsymbol{R}_{dd} + \boldsymbol{R}_{ii} + \boldsymbol{R}_{bb},
\end{aligned}$$

where $\boldsymbol{R}_{dd}$ and $\boldsymbol{R}_{ii}$ represent the correlation matrix of the desired and interference signals signature, respectively. Now, we define the signal-to-noise ratio $\mathrm{SNR} = \mathrm{Tr}(\boldsymbol{R}_{dd}) / \mathrm{Tr}(\boldsymbol{R}_{bb}) = \sigma_d^2/\sigma_b^2$ and the signal-to-interference ratio $\mathrm{SIR} = \mathrm{Tr}(\boldsymbol{R}_{dd}) / \mathrm{Tr}(\boldsymbol{R}_{ii}) = \sigma_d^2 / \sum_{r=2}^{R} \sigma_{s,r}^2$.

3. Beamforming Methods

We are interested in designing beamforming filters which recover the signal of interest from the observed signals by attenuating the interfering ones. If the direction cosines of each wavefront are *a priori* known from using parameter estimation methods such as MUSIC [3], for example, then one can employ the classical LCMV filter (recalled in Section 3.1.1). As the matrix inversions in the LCMV filter may bring computational issues, one can use its adaptive implementation, known as Frost's algorithm (outlined in Section 3.2.1).

In some scenarios, e.g., high-resolution beamforming with large-scale arrays, these classical solutions may become difficult to employ due to the large computational expenses of the filter optimization. To tackle this issue, we resort to parameter reduction by employing *separable filters* $\boldsymbol{w} = \boldsymbol{w}_h \otimes \boldsymbol{w}_v \in \mathbb{C}^N$, with $\boldsymbol{w}_h \in \mathbb{C}^{N_h}$ and $\boldsymbol{w}_v \in \mathbb{C}^{N_v}$. Instead of optimizing the full vector $\boldsymbol{w}$, as it is typically done, we optimize its lower-dimensional Kronecker factors. We show that this approach yields significant computational savings while exhibiting small source recovery degradation.

In the following, we discuss LCMV-based and Frost-based beamformers. The classical LCMV filter is recalled in Section 3.1.1 and we propose two novel separable extensions of this algorithm in Sections 3.1.2 and 3.1.3. The classical Frost's algorithm is then outlined in Section 3.2.1 and separable implementations are introduced in Section 3.2.2. Furthermore, we assess the computational complexity of each method by studying the number of complex multiplications necessary to obtain the beamforming filters.

3.1. LCMV-Type Beamformers

We consider a norm constraint regularization term (diagonal loading [24]) in all LCMV-based beamformers presented in this section to avoid numerical instability and to increase robustness against short sample support. Such numerical instability may arise whenever any two incoming signals arrive from similar directions, making $\boldsymbol{A}_v$ and $\boldsymbol{A}_h$ (and consequently $\boldsymbol{A}$) rank-deficient [16]. This norm constraint also ameliorates the filter robustness to slight errors in array characteristics [25].

3.1.1. Linearly Constrained Minimum Variance (LCMV)

The LCMV filter seeks to minimize the output signal $y[k] = \boldsymbol{w}^{\mathsf{H}}\boldsymbol{x}[k]$ considering linear constraints designed to recover the signal of interest and to filter out the undesired signals. We formulate a set of linear constraints based on the array factor: $\boldsymbol{w}^{\mathsf{H}}\boldsymbol{a}(p_r, q_r) = f_r$, $r = 1, \dots, R$. We assume perfect knowledge of all direction cosines, thus the linear constraints can be succinctly represented as $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{f}$, where $\boldsymbol{C} = \boldsymbol{A}$ denotes the array manifold constraint matrix and $\boldsymbol{f} = [f_1^*, f_2^*, \dots, f_R^*]^{\mathsf{T}}$ the array factor constraint vector. A typical choice for the array factor constraint vector which effectively attenuates the interfering signals is $\boldsymbol{f} = [1, 0, \dots, 0]^{\mathsf{T}}$, where the desired signal is indexed by $r = d = 1$.

The regularized LCMV beamformer is obtained by solving the following problem:

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimize}} \quad & \boldsymbol{w}^{\mathsf{H}}\boldsymbol{R}_{xx}\boldsymbol{w} + \delta \left\|\boldsymbol{w}\right\|_2^2 \\ \text{subject to} \quad & \boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{f}, \end{aligned} \tag{11}$$

where $\delta > 0$ is the regularization parameter. In Section 4, we discuss a practical procedure for choosing δ. By using Lagrange multipliers, we incorporate the linear constraint into the objective function:

$$J(\boldsymbol{w}, \boldsymbol{\lambda}) = \boldsymbol{w}^{\mathsf{H}}\boldsymbol{R}_{xx}\boldsymbol{w} + \boldsymbol{\lambda}^{\mathsf{H}}(\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} - \boldsymbol{f}) + \boldsymbol{\lambda}^{\mathsf{T}}(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{w}^* - \boldsymbol{f}^*) + \delta \left\|\boldsymbol{w}\right\|_2^2. \tag{12}$$

Table 1: Computational complexity of the LCMV beamformer.

$\boldsymbol{R}_{xx}$ **calculation**	**Number of multiplications**
Equation (9)	$3NR^2 + 3N^2R + R^2 + O(N^3) + O(R^3)$
Equation (10)	$2NR^2 + 2N^2R + R^2 + O(N^3) + O(R^3) + N^2K$

The gradient of (12) with respect to $\boldsymbol{w}^*$ is given by

$$\nabla_{\boldsymbol{w}^*} J(\boldsymbol{w}, \boldsymbol{\lambda}) = \boldsymbol{R}_{xx}\boldsymbol{w} + \boldsymbol{C}\boldsymbol{\lambda} + \delta\boldsymbol{w}. \tag{13}$$

Setting (13) to $\boldsymbol{0}_{N\times 1}$ and solving for $\boldsymbol{w}$ gives $\boldsymbol{w}_{\text{LCMV}} = -(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}\boldsymbol{\lambda}$. By plugging it into the linear constraint $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{f}$, we get $\boldsymbol{\lambda} = -[\boldsymbol{C}^{\mathsf{H}}(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}]^{-1}\boldsymbol{f}$, and finally:

$$\boldsymbol{w}_{\text{LCMV}} = (\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}\left[\boldsymbol{C}^{\mathsf{H}}(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}\right]^{-1}\boldsymbol{f}. \tag{14}$$

Note that the matrix $\boldsymbol{C}$ needs to be full-column rank R so $\left[\boldsymbol{C}^{\mathsf{H}}(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}\right]^{-1}$ exists. The correlation matrix $\boldsymbol{R}_{xx}$ can be calculated either by (9) if the statistics are known or by (10) otherwise.

The number of multiplications carried out in the LCMV filter design is summarized in Table 1. To conduct a fair comparison among the LCMV-type beamformers, we include the calculation $\boldsymbol{R}_{xx}$ in the total number of multiplications. The computation of (9) yields $NR^2 + N^2R$ multiplications, whereas (10) yields N^2K. Then, the inversion of $(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}$ and $\left[\boldsymbol{C}^{\mathsf{H}}(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}\right]^{-1}$ requires $O(N^3)$ and $O(R^3)$ multiplications, respectively. Finally, the calculation of $\boldsymbol{C}^{\mathsf{H}}(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}$ and $\boldsymbol{R}_{xx}^{-1}\boldsymbol{C}\left(\boldsymbol{C}^{\mathsf{H}}(\boldsymbol{R}_{xx} + \delta\boldsymbol{I}_N)^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{f}$ demands $(NR^2 + N^2R)$ and $(R^2 + N^2R + NR^2)$ multiplications, respectively.

3.1.2. Tensor Linearly Constrained Minimum Variance (TLCMV)

We now exploit the bi-dimensional signal model (7) to design a separable beamformer $\boldsymbol{w} = \boldsymbol{w}_v \otimes \boldsymbol{w}_h$ with the LCMV criterion. Consider the filter output signal

$$\begin{aligned} y[k] &= \boldsymbol{w}^{\mathsf{H}}\boldsymbol{x}[k] = (\boldsymbol{w}_v^{\mathsf{H}} \otimes \boldsymbol{w}_h^{\mathsf{H}})\,\text{vec}\,(\boldsymbol{X}[k]) \\ &= \boldsymbol{w}_h^{\mathsf{H}}\boldsymbol{X}[k]\boldsymbol{w}_v^* \qquad (15) \\ &= \boldsymbol{w}_v^{\mathsf{H}}\boldsymbol{X}^{\mathsf{T}}[k]\boldsymbol{w}_h^*. \qquad (16) \end{aligned}$$

The following algebra property was applied to get (15) and (16). Let $\boldsymbol{X} = \boldsymbol{ABC}$ with matching dimensions, thus $\text{vec}(\boldsymbol{X}) = (\boldsymbol{C}^{\mathsf{T}} \otimes \boldsymbol{A})\,\text{vec}(\boldsymbol{B})$ [26]. It then follows that $\boldsymbol{A} = \boldsymbol{w}_h^{\mathsf{H}}$, $\boldsymbol{B} = \boldsymbol{X}[k]$, and $\boldsymbol{C} = \boldsymbol{w}_v^*$ in (15). Equation (16) is simply obtained by transposing (15). These expressions allow us to define the horizontal and vertical sub-array inputs $\boldsymbol{u}_h[k] = \boldsymbol{X}[k]\boldsymbol{w}_v^*$ and $\boldsymbol{u}_v[k] = \boldsymbol{X}^{\mathsf{T}}[k]\boldsymbol{w}_h^*$, respectively and represent the beamformer output in terms of the horizontal and vertical sub-beamformers:

$$y[k] = \boldsymbol{w}_h^{\mathsf{H}}\boldsymbol{u}_h[k] = \boldsymbol{w}_v^{\mathsf{H}}\boldsymbol{u}_v[k]. \tag{17}$$

If the signal statistics are perfectly known, the correlation matrices of $\boldsymbol{u}_h[k]$ and $\boldsymbol{u}_v[k]$ are respectively given by [16]

$$\begin{gathered}\boldsymbol{R}_{u_h u_h} = \mathbb{E}\left[\boldsymbol{u}_h[k]\boldsymbol{u}_h^{\mathsf{H}}[k]\right] = \\ \boldsymbol{A}_{\text{URA,h}}(\boldsymbol{R}_{ss} \otimes \boldsymbol{w}_v^* \boldsymbol{w}_v^{\mathsf{T}})\boldsymbol{A}_{\text{URA,h}}^{\mathsf{H}} + \sigma_b^2 \|\boldsymbol{w}_v\|_2^2 \boldsymbol{I}_{N_h}, \end{gathered} \tag{18}$$

$$\begin{gathered}\boldsymbol{R}_{u_v u_v} = \mathbb{E}\left[\boldsymbol{u}_v[k]\boldsymbol{u}_v^{\mathsf{H}}[k]\right] = \\ \boldsymbol{A}_{\text{URA,v}}(\boldsymbol{R}_{ss} \otimes \boldsymbol{w}_h^* \boldsymbol{w}_h^{\mathsf{T}})\boldsymbol{A}_{\text{URA,v}}^{\mathsf{H}} + \sigma_b^2 \|\boldsymbol{w}_h\|_2^2 \boldsymbol{I}_{N_v}. \end{gathered} \tag{19}$$

Otherwise, they can be estimated as

$$\hat{\boldsymbol{R}}_{u_h u_h} = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{u}_h[k]\boldsymbol{u}_h^{\mathsf{H}}[k] = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{X}[k]\boldsymbol{w}_v^*\boldsymbol{w}_v^{\mathsf{T}}\boldsymbol{X}[k]^{\mathsf{H}}, \tag{20}$$

$$\hat{\boldsymbol{R}}_{u_v u_v} = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{u}_v[k]\boldsymbol{u}_v^{\mathsf{H}}[k] = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{X}^{\mathsf{T}}[k]\boldsymbol{w}_h^*\boldsymbol{w}_h^{\mathsf{T}}\boldsymbol{X}^*[k]. \tag{21}$$

We now show that the linear constraints $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{f}$ can also be separated. To this end, consider the following property of the Khatri-Rao product [26]:

$$(\boldsymbol{P} \diamond \boldsymbol{Q})^{\mathsf{T}}(\boldsymbol{r} \otimes \boldsymbol{s}) = (\boldsymbol{P}^{\mathsf{T}}\boldsymbol{r}) \odot (\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{s}), \tag{22}$$

where $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{r}$, and $\boldsymbol{s}$ are arbitrary matrices and vectors with matching dimensions. Replacing $\boldsymbol{P}$ by $\boldsymbol{C} = \boldsymbol{A} = \boldsymbol{A}_v \diamond \boldsymbol{A}_h$ and $\boldsymbol{Q}$ by $\boldsymbol{w} = \boldsymbol{w}_v \otimes \boldsymbol{w}_h$ in (22) and applying this property to $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{f}$, we get:

$$(\boldsymbol{A}_v \diamond \boldsymbol{A}_h)^{\mathsf{H}}(\boldsymbol{w}_v \otimes \boldsymbol{w}_h) = (\boldsymbol{A}_v^{\mathsf{H}}\boldsymbol{w}_v) \odot (\boldsymbol{A}_h^{\mathsf{H}}\boldsymbol{w}_h). \tag{23}$$

Now, consider the separable model (23) and assume $\boldsymbol{f} = \boldsymbol{f}_v \odot \boldsymbol{f}_h$, with $\boldsymbol{f}_v = \boldsymbol{f}_h = [1, 0, \ldots, 0]^{\mathsf{T}} \in \mathbb{C}^R$. Then the linear constraint $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{f}$ can be separated as

$$\begin{bmatrix} \left(\boldsymbol{a}_v(q_1)^{\mathsf{H}}\boldsymbol{w}_v\right)\left(\boldsymbol{a}_h(p_1)^{\mathsf{H}}\boldsymbol{w}_h\right) \\ \vdots \\ \left(\boldsymbol{a}_v(q_R)^{\mathsf{H}}\boldsymbol{w}_v\right)\left(\boldsymbol{a}_h(p_R)^{\mathsf{H}}\boldsymbol{w}_h\right) \end{bmatrix} = \begin{bmatrix} [\boldsymbol{f}]_1^* \\ \vdots \\ [\boldsymbol{f}]_R^* \end{bmatrix} = \begin{bmatrix} [\boldsymbol{f}_v]_1^*[\boldsymbol{f}_h]_1^* \\ \vdots \\ [\boldsymbol{f}_v]_R^*[\boldsymbol{f}_h]_R^* \end{bmatrix}. \tag{24}$$

Equation (24) reveals that the scalar elements of $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w}$ can be factorized into horizontal and vertical components $\boldsymbol{a}_h(p_r)^{\mathsf{H}}\boldsymbol{w}_h$ and $\boldsymbol{a}_v(q_r)^{\mathsf{H}}\boldsymbol{w}_v$, respectively, which leads to $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = (\boldsymbol{A}_v^{\mathsf{H}}\boldsymbol{w}_v) \odot (\boldsymbol{A}_h^{\mathsf{H}}\boldsymbol{w}_h) = \boldsymbol{f}_v \odot \boldsymbol{f}_h$.

To design the sub-filters $\boldsymbol{w}_h$ and $\boldsymbol{w}_v$, we formulate the following regularized LCMV problem:

$$\begin{aligned} \underset{\boldsymbol{w}_h,\, \boldsymbol{w}_v}{\text{minimize}} \quad & \mathbb{E}\left[|y[k]|^2\right] + \delta_h \|\boldsymbol{w}_h\|_2^2 + \delta_v \|\boldsymbol{w}_v\|_2^2 \\ \text{subject to} \quad & \boldsymbol{C}_h^{\mathsf{H}}\boldsymbol{w}_h = \boldsymbol{f}_h, \\ & \boldsymbol{C}_v^{\mathsf{H}}\boldsymbol{w}_v = \boldsymbol{f}_v, \end{aligned} \tag{25}$$

where δ_h, $\delta_v > 0$ are the regularization parameters. We reformulate (25) as an unconstrained optimization problem with the Lagrange multipliers method. Let

$$\begin{aligned} J(\boldsymbol{w}_h, \boldsymbol{\lambda}_h, \boldsymbol{w}_v, \boldsymbol{\lambda}_v) &= \mathbb{E}\left[|y[k]|^2\right] \\ &+ \boldsymbol{\lambda}_h^{\mathsf{H}}(\boldsymbol{C}_h^{\mathsf{H}}\boldsymbol{w}_h - \boldsymbol{f}_h) + \boldsymbol{\lambda}_h^{\mathsf{T}}(\boldsymbol{C}_h^{\mathsf{T}}\boldsymbol{w}_h^* - \boldsymbol{f}_h^*) \\ &+ \boldsymbol{\lambda}_v^{\mathsf{H}}(\boldsymbol{C}_v^{\mathsf{H}}\boldsymbol{w}_v - \boldsymbol{f}_v) + \boldsymbol{\lambda}_v^{\mathsf{T}}(\boldsymbol{C}_v^{\mathsf{T}}\boldsymbol{w}_v^* - \boldsymbol{f}_v^*) \\ &+ \delta_h \left\|\boldsymbol{w}_h\right\|_2^2 + \delta_v \left\|\boldsymbol{w}_v\right\|_2^2 \end{aligned} \tag{26}$$

denote the unconstrained objective function, where $\boldsymbol{\lambda}_h \in \mathbb{C}^{N_h}$ and $\boldsymbol{\lambda}_v \in \mathbb{C}^{N_v}$ stand for the horizontal and vertical sub-filters' multipliers. Thus (25) is rewritten as

$$\underset{\boldsymbol{w}_h, \boldsymbol{\lambda}_h, \boldsymbol{w}_v, \boldsymbol{\lambda}_v}{\text{minimize}} \quad J(\boldsymbol{w}_h, \boldsymbol{\lambda}_h, \boldsymbol{w}_v, \boldsymbol{\lambda}_v). \tag{27}$$

From (17), it follows that the output signal power in (26) can be expressed as

$$\mathbb{E}\left[|y[k]|^2\right] = \boldsymbol{w}_h^{\mathsf{H}}\boldsymbol{R}_{u_h u_h}\boldsymbol{w}_h = \boldsymbol{w}_v^{\mathsf{H}}\boldsymbol{R}_{u_v u_v}\boldsymbol{w}_v.$$

We employ the block coordinate descent method [27] (also known as nonlinear Gauss-Seidel method) to solve (27). It consists of sequentially solving the parameter blocks $(\boldsymbol{w}_h, \boldsymbol{\lambda}_h)$ and $(\boldsymbol{w}_v, \boldsymbol{\lambda}_v)$ until a convergence criterion triggers. This "divide to conquer" approach breaks (27) down into the two coupled subproblems:

$$\underset{\boldsymbol{w}_h, \boldsymbol{\lambda}_h}{\text{minimize}} \quad J_{\boldsymbol{w}_v, \boldsymbol{\lambda}_v}(\boldsymbol{w}_h, \boldsymbol{\lambda}_h) \tag{28}$$

and

$$\underset{\boldsymbol{w}_v, \boldsymbol{\lambda}_v}{\text{minimize}} \quad J_{\boldsymbol{w}_v, \boldsymbol{\lambda}_v}(\boldsymbol{w}_h, \boldsymbol{\lambda}_h) \tag{29}$$

where the sub-indexes of J now indicate which parameters are fixed when optimizing for the objective function's arguments. Hereafter, this method is referred to as Tensor LCMV (TLCMV) since it is based on unfolding matrices $\boldsymbol{A}_{\text{URA,h}}$ and $\boldsymbol{A}_{\text{URA,v}}$ of the array manifold tensor [16]. It is interesting to note that the "full" TLCMV problem (25) is non-convex (because of the Kronecker product between the vector parameters), while in sub-problems (28) and (29) the non-convexity is annulled since the correlation matrices $\boldsymbol{R}_{u_h u_h}$ and $\boldsymbol{R}_{u_v u_v}$ absorb the Kronecker product.

The solution of the horizontal sub-filter problem (28) is given by

$$\boldsymbol{w}_{\text{TLCMV,h}} = (\boldsymbol{R}_{u_h u_h} + \delta_h \boldsymbol{I}_{N_h})^{-1}\boldsymbol{C}_h \left[\boldsymbol{C}_h^{\mathsf{H}} \left(\boldsymbol{R}_{u_h u_h} + \delta_h \boldsymbol{I}_{N_h}\right)^{-1} \boldsymbol{C}_h\right]^{-1} \boldsymbol{f}_h. \tag{30}$$

Likewise, the vertical TLCMV sub-filter is obtained by

$$\boldsymbol{w}_{\text{TLCMV,v}} = (\boldsymbol{R}_{u_v u_v} + \delta_v \boldsymbol{I}_{N_h})^{-1}\boldsymbol{C}_v \left[\boldsymbol{C}_v^{\mathsf{H}} \left(\boldsymbol{R}_{u_v u_v} + \delta_v \boldsymbol{I}_{N_v}\right)^{-1} \boldsymbol{C}_v\right]^{-1} \boldsymbol{f}_v. \tag{31}$$

The TLCMV beamformer consists of, first, initializing $\boldsymbol{w}_v$ and $\boldsymbol{w}_h$ such that they satisfy their corresponding constraints, and then computing (30) and (31) sequentially until $\left\|\boldsymbol{w}^{(i)} - \boldsymbol{w}^{(i-1)}\right\|_2 < \epsilon$, where the super-index i represents the iteration number, and $\boldsymbol{w} = \boldsymbol{w}_{\mathrm{TLCMV,v}} \otimes \boldsymbol{w}_{\mathrm{TLCMV,h}}$. This procedure is summarized in Algorithm 1. The multiplications count for each line of this algorithm is detailed in Table 2.

To show that the TLCMV beamformer converges, we resort to the same reasoning as [17, 28]. The block coordinate descent method is known to converge when the solutions of (28) and (29) are uniquely attained [27, Proposition 2.7.1]. Since these sub-problems are themselves convex, this convergence criterion is satisfied. To illustrate this property, consider the initialization $\boldsymbol{w}_v^{(0)} = [1, \ldots, 0]^{\mathsf{T}}$. At the n-th coordinate block descent iteration, we have

$$
\begin{gathered}
(\boldsymbol{w}_h^{(n)}, \boldsymbol{\lambda}_h^{(n)}) = \arg\min J_{\boldsymbol{w}_v^{(n-1)}, \boldsymbol{\lambda}_v^{(n-1)}} \\
(\boldsymbol{w}_v^{(n)}, \boldsymbol{\lambda}_v^{(n)}) = \arg\min J_{\boldsymbol{w}_h^{(n)}, \boldsymbol{\lambda}_h^{(n)}} \ .
\end{gathered}
$$

as the local optimum solutions. Based on the convexity of (28) and (29), it follows that

$$
\begin{gathered}
J(\boldsymbol{w}_h^{(n-1)}, \boldsymbol{\lambda}_h^{(n-1)}, \boldsymbol{w}_v^{(n-1)}, \boldsymbol{\lambda}_v^{(n-1)}) \geq \\
J(\boldsymbol{w}_h^{(n)}, \boldsymbol{\lambda}_h^{(n)}, \boldsymbol{w}_v^{(n-1)}, \boldsymbol{\lambda}_v^{(n-1)}) \geq J(\boldsymbol{w}_h^{(n)}, \boldsymbol{\lambda}_h^{(n)}, \boldsymbol{w}_v^{(n)}, \boldsymbol{\lambda}_v^{(n)}),
\end{gathered}
$$

hence the local minimum solution is lower-bounded. Moreover, the TLCMV solution pair $(\boldsymbol{w}_h, \boldsymbol{w}_v)$, after convergence, is unique, unlike the methods of [16, 17, 28], which exhibit scale ambiguity. TLCMV does not face this issue owing to the array factor constraints $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{f}$, which removes any scale ambiguity in $\boldsymbol{w}_h$ and $\boldsymbol{w}_v$.

Algorithm 1 Tensor LCMV beamformer

```
Require: $\epsilon$, $\delta_h$, $\delta_v$, $\boldsymbol{f}_h$, $\boldsymbol{f}_v$
$\boldsymbol{w}_v \leftarrow [1, 0, \ldots, 0]^{\mathsf{T}}$ and $\boldsymbol{w}_h \leftarrow [1, 0, \ldots, 0]^{\mathsf{T}}$
$i \leftarrow 0$
$\boldsymbol{w}^{(i)} \leftarrow \boldsymbol{w}_v \otimes \boldsymbol{w}_h$
repeat
    Compute $\boldsymbol{R}_{u_h u_h}$ by (18) or (20)
    Update $\boldsymbol{w}_h$ by (30)
    Compute $\boldsymbol{R}_{u_v u_v}$ by (19) or (21)
    Update $\boldsymbol{w}_v$ by (31)
    $i \leftarrow i + 1$
    $\boldsymbol{w}^{(i)} \leftarrow \boldsymbol{w}_v \otimes \boldsymbol{w}_h$
until $\left\|\boldsymbol{w}^{(i)} - \boldsymbol{w}^{(i-1)}\right\|_2^2 < \epsilon$
return $\boldsymbol{w}^{(i)}$
```

3.1.3. Kronecker Linearly Constrained Minimum Variance (KLCMV)

We regard the rectangular array as a set of linear sub-arrays in order to better understand the sensitivity in both angular domains and to present our

Table 2: Computational complexity of the TLCMV beamformer (Algorithm 1).

Line	**Number of multiplications (per iteration)**
5	If (18): $(N_v^2 + R^2N_v^2) + (N_hR^2N_v^2 + N_h^2RN_V) + (N_v + N_h + 1)$ If (20): $K(2N_hN_v + N_h^2)$
6	$2N_h^2R + 2N_hR^2 + R^2 + O(N_h^3) + O(R^3)$
7	If (19): $(N_h^2 + R^2N_h^2) + (N_vR^2N_h^2 + N_v^2RN_h) + (N_v + N_h + 1)$ If (21): $K(2N_hN_v + N_v^2)$
8	$2N_v^2R + 2N_vR^2 + R^2 + O(N_v^3) + O(R^3)$
10	N
11	N

separable LCMV beamformer. Any rectangular array is oriented relative to two axes, say, y and z. The linear sub-arrays oriented relative to y discriminate sources in the rectangular array's azimuth direction. Simultaneously, the linear sub-arrays relative to z are responsible for the rectangular array's sensitivity in the polar domain. This sub-array interpretation is strongly connected to the array separability property presented in Section 2. In fact, the matrix factors $\boldsymbol{A}_v$ and $\boldsymbol{A}_h$ of the URA manifold matrix $\boldsymbol{A} = \boldsymbol{A}_v \diamond \boldsymbol{A}_h$ are *uniform linear array* manifold matrices of sub-arrays along the vertical and horizontal directions, respectively.

This sub-array interpretation motivates us to design independent beamformers for $\boldsymbol{A}_v$ and $\boldsymbol{A}_h$ and combine them afterward to get the full rectangular array beamformer. The horizontal sub-array is responsible to filter the azimuth domain whereas the vertical sub-array the polar domain. As we have shown in [16], combining the sub-filters by the Kronecker product preserves the sub-filter properties at each domain, effectively recovering the signal of interest and attenuating the undesired signals. In this direction, instead of spatially sampling the signals using all N antennas, we consider only a horizontal sub-array with N_h antennas and a vertical sub-array with N_v antennas whose observed signals are respectively represented by

$$\boldsymbol{x}_h[k] = \boldsymbol{A}_h\boldsymbol{s}[k] + \boldsymbol{b}_h[k], \tag{32}$$

$$\boldsymbol{x}_v[k] = \boldsymbol{A}_v\boldsymbol{s}[k] + \boldsymbol{b}_v[k], \tag{33}$$

where $\boldsymbol{b}_h[k] \in \mathbb{C}^{N_h}$ and $\boldsymbol{b}_v[k] \in \mathbb{C}^{N_v}$ represent the additive Gaussian noise vector observed at the horizontal and vertical sub-arrays, respectively. These vectors are defined as

$$[\boldsymbol{b}_h[k]]_{n_h} = b_{n_h+(n_v-1)N_h}[k]\big|_{n_v=1},$$

$$[\boldsymbol{b}_v[k]]_{n_v} = b_{n_h+(n_v-1)N_h}[k]\big|_{n_h=1}.$$

The correlation matrices of (32) and (33) are given by

$$\boldsymbol{R}_{x_hx_h} = \boldsymbol{A}_h\boldsymbol{R}_{ss}\boldsymbol{A}_h^{\mathsf{H}} + \sigma_b^2\boldsymbol{I}_{N_h}, \tag{34}$$

$$\boldsymbol{R}_{x_vx_v} = \boldsymbol{A}_v\boldsymbol{R}_{ss}\boldsymbol{A}_v^{\mathsf{H}} + \sigma_b^2\boldsymbol{I}_{N_v}, \tag{35}$$

respectively, and their sample estimators are

$$\hat{\boldsymbol{R}}_{x_h x_h} = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{x}_h[k] \boldsymbol{x}_h^{\mathsf{H}}[k], \tag{36}$$

$$\hat{\boldsymbol{R}}_{x_v x_v} = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{x}_v[k] \boldsymbol{x}_v^{\mathsf{H}}[k]. \tag{37}$$

From (32) and (33), we formulate the following regularized LCMV problems

$$\begin{aligned} \underset{\boldsymbol{w}_h}{\text{minimize}} \quad & \boldsymbol{w}_h^{\mathsf{H}} \boldsymbol{R}_{x_h x_h} \boldsymbol{w}_h + \delta_h \left\| \boldsymbol{w}_h \right\|_2^2 \\ \text{subject to} \quad & \boldsymbol{C}_h^{\mathsf{H}} \boldsymbol{w}_h = \boldsymbol{f}_h \end{aligned} \tag{38}$$

and

$$\begin{aligned} \underset{\boldsymbol{w}_v}{\text{minimize}} \quad & \boldsymbol{w}_v^{\mathsf{H}} \boldsymbol{R}_{x_v x_v} \boldsymbol{w}_v + \delta_v \left\| \boldsymbol{w}_v \right\|_2^2 \\ \text{subject to} \quad & \boldsymbol{C}_v^{\mathsf{H}} \boldsymbol{w}_v = \boldsymbol{f}_v, \end{aligned} \tag{39}$$

Similar to the TLCMV problems, (38) and (39) may become ill-posed for unfavorable angles of arrival and thus are regularized by the ℓ_2-norm penalization terms, weighted by the positive scalars δ_h and δ_v. The solutions of the Kronecker LCMV (KLCMV) problems (40) and (39) are given by

$$\boldsymbol{w}_{\text{KLCMV,h}} = (\boldsymbol{R}_{x_h x_h} + \delta_h \boldsymbol{I}_{N_h})^{-1} \boldsymbol{C}_h \left[\boldsymbol{C}_h^{\mathsf{H}} \left(\boldsymbol{R}_{x_h x_h} + \delta_h \boldsymbol{I}_{N_h} \right)^{-1} \boldsymbol{C}_h \right]^{-1} \boldsymbol{f}_h, \tag{40}$$

$$\boldsymbol{w}_{\text{KLCMV,v}} = (\boldsymbol{R}_{x_v x_v} + \delta_v \boldsymbol{I}_{N_h})^{-1} \boldsymbol{C}_h \left[\boldsymbol{C}_v^{\mathsf{H}} \left(\boldsymbol{R}_{x_v x_v} + \delta_v \boldsymbol{I}_{N_v} \right)^{-1} \boldsymbol{C}_v \right]^{-1} \boldsymbol{f}_v. \tag{41}$$

Note that the KLCMV solutions are independent, i.e., (40) and (41) do not depend on each other, as it is observed in the TLCMV solutions. Therefore, an iterative optimization procedure is unnecessary. Once these sub-filters are calculated, we build the full solution as $\boldsymbol{w}_{\text{KLCMV}} = \boldsymbol{w}_{\text{KLCMV,v}} \otimes \boldsymbol{w}_{\text{KLCMV,h}}$. The KLCMV beamformer is outlined in Algorithm 2 and the associated number of multiplications is specified in Table 3.

Algorithm 2 Kronecker LCMV beamformer

```
Require: $\delta_h$, $\delta_v$, $\boldsymbol{f}_h$, $\boldsymbol{f}_v$
Compute $\boldsymbol{R}_{x_h x_h}$ by (34) or (36)
Update $\boldsymbol{w}_h$ by (40)
Compute $\boldsymbol{R}_{x_v x_v}$ by (35) or (37)
Update $\boldsymbol{w}_v$ by (41)
return $\boldsymbol{w} \leftarrow \boldsymbol{w}_v \otimes \boldsymbol{w}_h$
```

3.2. Frost-Type Beamformers

In some situations, adaptive solutions may be preferred over "block" solutions, like the LCMV-type beamformers, due to their ability to track parameter

Table 3: Computational complexity of the KLCMV beamformer (Algorithm 2).

Line	Number of multiplications
1	If (34): $N_h^2 R + N_h R^2$
	If (36): $N_h^2 K$
2	$2N_h^2 R + 2N_h R^2 + R^2 + O(N_h^3) + O(R^3)$
3	If (35): $N_v^2 R + N_v R^2$
	If (37): $N_v^2 K$
4	$2N_v^2 R + 2N_v R^2 + R^2 + O(N_v^3) + O(R^3)$
5	N

variations. Moreover, inverting correlation matrices may still bring overwhelming computational complexity. As a solution, one can implement the LCMV-type beamformers using the stochastic gradient method to tackle these shortcomings. For completeness, we first recall the conventional Frost's beamformer, which is the stochastic gradient implementation of the classical LCMV filter. Then, the proposed separable extensions of Frost's algorithm are formulated.

3.2.1. Frost's Algorithm

One can solve the LCMV problem (11) by means of the stochastic gradient method to avoid inverting correlation matrices. This solution is known as Frost's algorithm [21]. From (13), it follows that the gradient descent formula is

$$\begin{aligned} \boldsymbol{w}[k+1] &= \boldsymbol{w}[k] - \mu \nabla_{\boldsymbol{w}^*} J(\boldsymbol{w}[k], \boldsymbol{\lambda}[k]) \\ &= \boldsymbol{w}[k] - \mu(\boldsymbol{R}_{xx}\boldsymbol{w}[k] + \boldsymbol{C}\boldsymbol{\lambda}[k]), \end{aligned} \tag{42}$$

where μ is a positive step size factor. The filter update (42) has to satisfy the linear constraint $\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w}[k+1] = \boldsymbol{f}$. Therefore, we insert (42) into this constraint and solve for $\boldsymbol{\lambda}[n]$:

$$\boldsymbol{\lambda}[k] = \frac{1}{\mu}(\boldsymbol{C}^{\mathsf{H}}\boldsymbol{C})^{-1}\boldsymbol{C}^{\mathsf{H}}\boldsymbol{w}[k] - (\boldsymbol{C}^{\mathsf{H}}\boldsymbol{C})^{-1}\boldsymbol{C}\boldsymbol{R}_{xx}\boldsymbol{w}[k] - \frac{1}{\mu}(\boldsymbol{C}^{\mathsf{H}}\boldsymbol{C})\boldsymbol{f}. \tag{43}$$

We derive the Frost's algorithm update formula by plugging (43) into (42), approximating the correlation matrix as $\boldsymbol{R}_{xx} \approx \boldsymbol{x}[k]\boldsymbol{x}^{\mathsf{H}}[k]$, and defining $\boldsymbol{P} = \boldsymbol{I}_N - \boldsymbol{C}(\boldsymbol{C}^{\mathsf{H}}\boldsymbol{C})^{-1}\boldsymbol{C}^{\mathsf{H}}$ and $\tilde{\boldsymbol{f}} = \boldsymbol{C}(\boldsymbol{C}^{\mathsf{H}}\boldsymbol{C})^{-1}\boldsymbol{f}$ [21]:

$$\begin{aligned} \boldsymbol{w}[k+1] &= \tilde{\boldsymbol{f}} + \boldsymbol{P}(\boldsymbol{w}[k] - \mu\boldsymbol{R}_{xx}\boldsymbol{w}[k]) \\ &= \tilde{\boldsymbol{f}} + \boldsymbol{P}(\boldsymbol{w}[k] - \mu\boldsymbol{x}[k]y^*[k]). \end{aligned}$$

In order to iterate towards a global minimum, the algorithm has to be initialized such that the linear constraints are satisfied. A typical initialization is $\boldsymbol{w}[0] = \tilde{\boldsymbol{f}}$ [21]. This beamformer is summarized in Algorithm 3 and its computational complexity is evaluated in Table 4.

Frost's algorithm is shown to converge to the LCMV solution $\boldsymbol{w}_{\text{opt}}$, given by (14), if the step size μ satisfies $0 < \mu < 1/\lambda_{\max}$, where $\lambda_{\max}$ denotes the largest eigenvalue of $\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}^{\mathsf{H}}$. For future convenience, we review the convergence proof presented in [21]. First, consider the following identities:

$$\tilde{\boldsymbol{f}} = \boldsymbol{w}_{\text{opt}} - \boldsymbol{P}\boldsymbol{w}_{\text{opt}}, \tag{44}$$

$$\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{w}_{\text{opt}} = \boldsymbol{0}_{N\times 1}, \tag{45}$$

$$\boldsymbol{P}^2 = \boldsymbol{P}. \tag{46}$$

Equation (44) is obtained by manipulating the product $\boldsymbol{P}\boldsymbol{w}_{\text{opt}}$, (45) is simply derived by plugging (14) into the equation, and the idempotence of $\boldsymbol{P}$ in (46) follows directly from its definition. Now, define the mean value of the weight vector $\mathbb{E}\left[\boldsymbol{w}[k+1]\right] = \tilde{\boldsymbol{f}} + \boldsymbol{P}(\mathbb{E}\left[\boldsymbol{w}[k]\right] - \mu\boldsymbol{R}_{xx}\mathbb{E}\left[\boldsymbol{w}[k]\right])$. Substituting (44) and (45) into $\mathbb{E}\left[\boldsymbol{w}[k+1]\right]$ yields:

$$\mathbb{E}\left[\boldsymbol{w}[k+1]\right] = \boldsymbol{w}_{\text{opt}} - \boldsymbol{P}\boldsymbol{w}_{\text{opt}} + \overbrace{\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{w}_{\text{opt}}}^{\boldsymbol{0}_{N\times 1}} + \boldsymbol{P}(\mathbb{E}\left[\boldsymbol{w}[k]\right] - \mu\boldsymbol{R}_{xx}\boldsymbol{w}[k]) \tag{47}$$

Subtracting $\boldsymbol{w}_{\text{opt}}$ from (47):

$$\mathbb{E}\left[\boldsymbol{w}[k+1]\right] - \boldsymbol{w}_{\text{opt}} = \boldsymbol{P}(\mathbb{E}\left[\boldsymbol{w}[k]\right] - \boldsymbol{w}_{\text{opt}} - \mu\boldsymbol{R}_{xx}(\mathbb{E}\left[\boldsymbol{w}[k]\right] - \boldsymbol{w}_{\text{opt}})) \tag{48}$$

Define $\boldsymbol{v}[k] = \mathbb{E}\left[\boldsymbol{w}[k]\right] - \boldsymbol{w}_{\text{opt}} \in \mathbb{C}^N$ as the mean distance vector from the optimal solution. Then, (48) can be expressed as

$$\boldsymbol{v}[k+1] = \boldsymbol{P}\boldsymbol{v}[k] - \mu\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{v}[k]. \tag{49}$$

Pre-multiplying (49) by $\boldsymbol{P}$ and applying its idempotence property (46) shows that $\boldsymbol{P}\boldsymbol{v}[k] = \boldsymbol{v}[k]$. Thus, (49) can be expressed as

$$\begin{aligned}\boldsymbol{v}[k+1] &= (\boldsymbol{I}_N - \mu\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P})\boldsymbol{v}[k] \\ &= (\boldsymbol{I}_N - \mu\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P})^{k+1}\boldsymbol{v}[0]\end{aligned} \tag{50}$$

In [21], the author argues that $\boldsymbol{v}[0] = \tilde{\boldsymbol{f}} - \boldsymbol{w}_{\text{opt}}$ can be expressed as a linear combination of the non-zero $\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}$ corresponding to non-zero eigenvalues. If $\boldsymbol{v}[0]$ is set to any eigenvector of $\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}$, say $\boldsymbol{q}_i$, with eigenvalue $\lambda_i \neq 0$, then (50) becomes

$$\boldsymbol{v}[k+1] = (1 - \mu\lambda_i)^{k+1}\boldsymbol{q}_i,$$

i.e., the mean distance vector along any eigenvector $\boldsymbol{q}_i$ of $\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}$ becomes a geometric series with ratio $(1 - \mu\lambda_i)$ [21]. Therefore, if the step size is chosen so that $0 < \mu < 1/\lambda_{\max}$, then the norm of $\boldsymbol{v}[k+1]$ is bounded as [21]

$$(1 - \mu\lambda_{\max})^{k+1}\left\|\boldsymbol{v}[0]\right\|_2 \leq \left\|\boldsymbol{v}[k+1]\right\|_2 \leq (1 - \mu\lambda_{\min})^{k+1}\left\|\boldsymbol{v}[0]\right\|_2,$$

where $\lambda_{\min}$ is the smallest eigenvalue of $\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}$. Consequently, $\lim_{k\to\infty}\left\|\boldsymbol{v}[k]\right\|_2 = 0$ and $\boldsymbol{w}[k]$ converges to $\boldsymbol{w}_{\text{opt}}$.

In practice, however, the eigenvalues of $\boldsymbol{R}_{xx}$ are probably unknown and the condition $0 < \mu < 1/\lambda_{\max}$ becomes hard to verify. In [21], this condition is relaxed to $0 < \mu < \frac{2}{3\,\mathrm{Tr}(\boldsymbol{R}_{xx})}$ by using some bounds on the steady-state misadjustment. The latter condition is simpler to calculate than the former, since $\mathrm{Tr}(\boldsymbol{R}_{xx}) = \mathbb{E}\left[\left\|\boldsymbol{x}[k]\right\|_2^2\right]$ represents the received signal vector power and can be easily estimated.

Algorithm 3 Frost's algorithm

```
1: $\boldsymbol{P} \leftarrow \boldsymbol{I}_N - \boldsymbol{C}(\boldsymbol{C}^{\mathsf{H}}\boldsymbol{C})^{-1}\boldsymbol{C}^{\mathsf{H}}$
2: $\tilde{\boldsymbol{f}} \leftarrow \boldsymbol{C}(\boldsymbol{C}^{\mathsf{H}}\boldsymbol{C})^{-1}\boldsymbol{f}$
3: $\boldsymbol{w}[0] \leftarrow \tilde{\boldsymbol{f}}$
4: for $k = 0, \ldots, K-1$ do
5:     $y[k] \leftarrow \boldsymbol{w}^{\mathsf{H}}[k]\boldsymbol{x}[k]$
6:     $\boldsymbol{w}[k+1] = \tilde{\boldsymbol{f}} + \boldsymbol{P}(\boldsymbol{w}[k] - \mu\boldsymbol{x}[k]y^*[k])$
7: end for
8: return $\boldsymbol{w}[k+1]$
```

Table 4: Computational complexity of Frost's beamformer (Algorithm 3).

Line	Number of multiplications
1	$N^2R + NR^2$
2	$NR^2 + NR + R^2 + O(R^3)$
5	KN
6	$K(N^2 + N + 1)$

3.2.2. Separable Frost Algorithms

A first strategy to design a separable beamformer $\boldsymbol{w} = \boldsymbol{w}_v \otimes \boldsymbol{w}_h$ consists of applying the stochastic gradient method to the sub-array input signals $\boldsymbol{u}_v[k]$ and $\boldsymbol{u}_h[k]$ (see Section 3.1.2). Since these input signals are coupled, we employ block coordinate descent, as in TLCMV. This beamformer can be regarded as an adaptive TLCMV implementation, thus we call it Tensor Frost (TFrost). It is summarized in Algorithm 4 and the complexity associated with each step is provided in Table 5. Another approach to separable beamformer design is to independently optimize sub-array beamformers, as done in KLCMV (see Section 3.1.3). At each Frost iteration, two independent sub-filters are computed as described in Algorithm 5, which we refer to as Kronecker Frost (KFrost). The computational cost of the KFrost algorithm is presented in Table 6.

Theoretical analysis of coupled adaptive filters is not straightforward [29]. The convergence proof of the classical Frost's algorithm discussed in Section 3.2.1 assumes that the filter input signal is stationary, implying $\boldsymbol{R}_{xx}$ does not change over k. Unfortunately, this is not the case for TFrost, as the related correlation matrices change whenever $\boldsymbol{w}_h[k]$ or $\boldsymbol{w}_v[k]$ are updated. However, we can still obtain meaningful convergence results from TFrost. Let us consider the optimization of $\boldsymbol{w}_h[k]$ keeping $\boldsymbol{w}_v[k]$ fixed. The horizontal sub-array input correlation matrix $\boldsymbol{R}_{u_h u_h}[k] = \mathbb{E}\left[\boldsymbol{X}[k]\boldsymbol{w}_v^*[k]\boldsymbol{w}_v^{\mathsf{T}}[k]\boldsymbol{X}^{\mathsf{T}}[k]\right]$ does not vary as long $\boldsymbol{w}_v[k]$ remains fixed, and, thus, the convergence analysis of Section 3.2.1 applies. For fixed $\boldsymbol{R}_{u_h u_h}[k]$, a quadratic surface, whose minimum point is achieved by the horizontal TLCMV filter (30), is available. Hence, the local optimization with respect to $\boldsymbol{w}_h[k]$ is guaranteed to converge as long as the horizontal step size μ_h satisfies $0 < \mu_h < 1/\lambda_{\mathrm{h,max}}[k]$, where $\lambda_{\mathrm{h,max}}[k]$ stands for the largest eigenvalue of $\boldsymbol{P}_h\boldsymbol{R}_{u_h u_h}[k]\boldsymbol{P}_h$. The same argument holds for the vertical sub-filter; $\boldsymbol{w}_v[k]$ converges to the vertical TLCMV filter (31) with fixed $\boldsymbol{w}_h[k+1]$

Table 5: Computational complexity of the Tensor Frost beamformer (Algorithm 4).

Line	Number of multiplications
1	$N_h^2 R + N_h R^2 + N_v^2 R + N_v R^2$
2	$N_h R^2 + N_h R + N_v R^2 + N_v R + 2R^2 + 2O(R^3)$
5	$K N_h N_v$
6	$K N_h$
7	$K(N_h^2 + N_h + 1)$
8	$K N_h N_v$
9	$K N_v$
10	$K(N_v^2 + N_v + 1)$
12	$N_h N_v$

as long as the vertical step size μ_v is within the interval $0 < \mu_v < 1/\lambda_{\mathrm{v,max}}[k]$. As the TLCMV block coordinate descent iterates, the optima points (given by the TLCMV sub-filters) also iterates until convergence (see Section 3.1.2) and so do the TFrost sub-filters. Note that the TFrost step size conditions to convergence can be relaxed to $0 < \mu_h < \frac{2}{3\,\mathrm{Tr}(\boldsymbol{R}_{u_h u_h}[k])}$ and $0 < \mu_v < \frac{2}{3\,\mathrm{Tr}(\boldsymbol{R}_{u_v u_v}[k])}$ by following the same reasoning presented in [21].

The convergence analysis of KFrost is relatively simpler to discuss, as it basically consists of two independent Frost sub-filters and the correlation matrices $\boldsymbol{R}_{x_h x_h}$ and $\boldsymbol{R}_{x_v x_v}$ are static. Let $\eta_{\mathrm{h,max}}$ and $\eta_{\mathrm{v,max}}$ denote the largest eigenvalues of $\boldsymbol{P}_h \boldsymbol{R}_{x_h x_h} \boldsymbol{P}_h$ and $\boldsymbol{P}_v \boldsymbol{R}_{x_v x_v} \boldsymbol{P}_v$, respectively. Therefore, the KFrost sub-filters $\boldsymbol{w}_h[k]$ and $\boldsymbol{w}_v[k]$ are guaranteed to converge to the KLCMV sub-filters (40) and (41) if the corresponding step size factors μ_h and μ_v satisfy $0 < \mu_h < 1/\eta_{\mathrm{h,max}}$ and $0 < \mu_v < 1/\eta_{\mathrm{v,max}}$, respectively. These conditions can be relaxed to [21]: $0 < \mu_h < \frac{2}{3\,\mathrm{Tr}(\boldsymbol{R}_{x_h x_h})}$ and $0 < \mu_v < \frac{2}{3\,\mathrm{Tr}(\boldsymbol{R}_{x_v x_v})}$.

Algorithm 4 Tensor Frost algorithm

```
1:  $\boldsymbol{P}_h \leftarrow \boldsymbol{I}_{N_h} - \boldsymbol{C}_h(\boldsymbol{C}_h^{\mathsf{H}}\boldsymbol{C}_h)^{-1}\boldsymbol{C}_h^{\mathsf{H}}$ and $\boldsymbol{P}_v \leftarrow \boldsymbol{I}_{N_v} - \boldsymbol{C}_v(\boldsymbol{C}_v^{\mathsf{H}}\boldsymbol{C}_v)^{-1}\boldsymbol{C}_v^{\mathsf{H}}$
2:  $\tilde{\boldsymbol{f}}_h \leftarrow \boldsymbol{C}_h(\boldsymbol{C}_h^{\mathsf{H}}\boldsymbol{C}_h)^{-1}\boldsymbol{f}_h$ and $\tilde{\boldsymbol{f}}_v \leftarrow \boldsymbol{C}_v(\boldsymbol{C}_v^{\mathsf{H}}\boldsymbol{C}_v)^{-1}\boldsymbol{f}_v$
3:  $\boldsymbol{w}_h[0] \leftarrow \tilde{\boldsymbol{f}}_h$ and $\boldsymbol{w}_v[0] \leftarrow \tilde{\boldsymbol{f}}_v$
4:  for $k = 0, \ldots, K-1$ do
5:      $\boldsymbol{u}_h[k] \leftarrow \boldsymbol{X}[k]\boldsymbol{w}_v^*[k]$
6:      $y_h[k] \leftarrow \boldsymbol{w}_h^{\mathsf{H}}[k]\boldsymbol{u}_h[k]$
7:      $\boldsymbol{w}_h[k+1] = \tilde{\boldsymbol{f}}_h + \boldsymbol{P}_h(\boldsymbol{w}_h[k] - \mu_h \boldsymbol{u}_h[k] y_h^*[k])$
8:      $\boldsymbol{u}_v[k] \leftarrow \boldsymbol{X}^{\mathsf{T}}[k]\boldsymbol{w}_h^*[k]$
9:      $y_v[k] \leftarrow \boldsymbol{w}_v^{\mathsf{H}}[k]\boldsymbol{u}_v[k]$
10:     $\boldsymbol{w}_v[k+1] = \tilde{\boldsymbol{f}}_v + \boldsymbol{P}_v(\boldsymbol{w}_v[k] - \mu_v \boldsymbol{u}_v[k] y_v^*[k])$
11: end for
12: return $\boldsymbol{w}_v[k+1] \otimes \boldsymbol{w}_h[k+1]$
```

Algorithm 5 Kronecker Frost algorithm

```
1: $\boldsymbol{P}_h \leftarrow \boldsymbol{I}_{N_h} - \boldsymbol{C}_h(\boldsymbol{C}_h^{\mathsf{H}}\boldsymbol{C}_h)^{-1}\boldsymbol{C}_h^{\mathsf{H}}$ and $\boldsymbol{P}_v \leftarrow \boldsymbol{I}_{N_v} - \boldsymbol{C}_v(\boldsymbol{C}_v^{\mathsf{H}}\boldsymbol{C}_v)^{-1}\boldsymbol{C}_v^{\mathsf{H}}$
2: $\tilde{\boldsymbol{f}}_h \leftarrow \boldsymbol{C}_h(\boldsymbol{C}_h^{\mathsf{H}}\boldsymbol{C}_h)^{-1}\boldsymbol{f}_h$ and $\tilde{\boldsymbol{f}}_v \leftarrow \boldsymbol{C}_v(\boldsymbol{C}_v^{\mathsf{H}}\boldsymbol{C}_v)^{-1}\boldsymbol{f}_v$
3: $\boldsymbol{w}_h[0] \leftarrow \tilde{\boldsymbol{f}}_h$ and $\boldsymbol{w}_v[0] \leftarrow \tilde{\boldsymbol{f}}_v$
4: for $k = 0, \ldots, K-1$ do
5:     $y_h[k] \leftarrow \boldsymbol{w}_h^{\mathsf{H}}[k]\boldsymbol{x}_h[k]$
6:     $y_v[k] \leftarrow \boldsymbol{w}_v^{\mathsf{H}}[k]\boldsymbol{x}_v[k]$
7:     $\boldsymbol{w}_h[k+1] = \tilde{\boldsymbol{f}}_h + \boldsymbol{P}_h(\boldsymbol{w}_h[k] - \mu_h \boldsymbol{x}_h[k] y_h^*[k])$
8:     $\boldsymbol{w}_v[k+1] = \tilde{\boldsymbol{f}}_v + \boldsymbol{P}_v(\boldsymbol{w}_v[k] - \mu_v \boldsymbol{x}_v[k] y_v^*[k])$
9: end for
10: return $\boldsymbol{w}_v[k+1] \otimes \boldsymbol{w}_h[k+1]$
```

Table 6: Computational complexity of Kronecker Frost beamformer (Algorithm 5).

Line	Number of multiplications
1	$N_h^2 R + N_h R^2 + N_v^2 R + N_v R^2$
2	$N_h R^2 + N_h R + N_v R^2 + N_v R + 2R^2 + 2O(R^3)$
5	KN_h
6	KN_v
7	$K(N_h^2 + N_h + 1)$
8	$K(N_v^2 + N_v + 1)$
10	$N_h N_v$

4. Simulation Results

In this section, we present the results of numerical experiments to compare the source recovery and computational performances of the proposed beamforming methods. As figure of merit, we consider the output SINR, which is defined as the desired signal power over the interference plus noise power after beamforming, i.e.,

$$\mathrm{SINR}_{\mathrm{out}} = \frac{\boldsymbol{w}^{\mathsf{H}}\boldsymbol{R}_{dd}\boldsymbol{w}}{\boldsymbol{w}^{\mathsf{H}}(\boldsymbol{R}_{ii} + \boldsymbol{R}_{bb})\boldsymbol{w}}.$$

First, let us present and discuss the results of the LCMV-type beamformers, which were obtained by Monte Carlo (MC) simulations from 1000 runs. At each MC experiment, a novel array manifold matrix $\boldsymbol{A}$ realization is generated. We set as system parameters $U = 4$ wavefronts and SIR $= -5$dB. Hereafter, we refer to the solutions obtained with perfect statistics knowledge as *analytical* and to those which employ statistical estimates as *sample* solutions. We consider in our simulations only square arrays, i.e., the system has $N_h N_v = N$ antennas, with $N_h = N_v$, although the proposed methods could be directly applied to uniform rectangular arrays in general. In wireless communications systems, the base-stations may employ rectangular arrays to increase the spatial resolution in a desired angular domain (usually azimuth). In [30], for example, 5G-compliant systems with 4 × 4 and 4 × 2 dual-polarized antennas are considered. We set

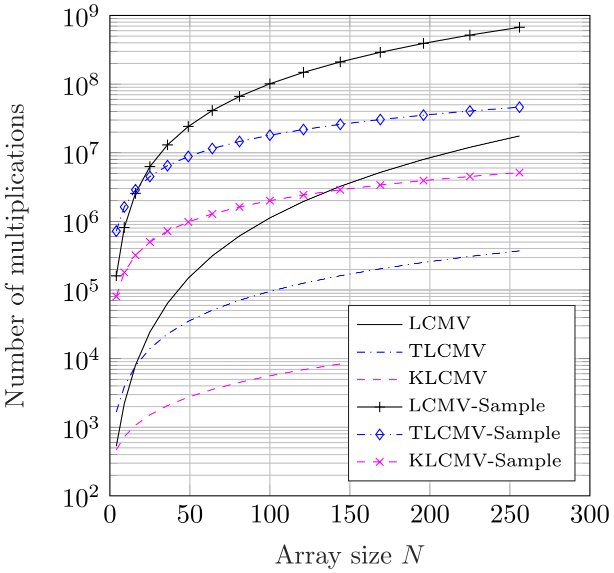

Figure 2: $I = 2$ (analytical), $I = 3$ (sample), $K = 10^4$ snapshots.

the TLCMV convergence threshold to $\epsilon = 10^{-3}$. We carried out preliminary experiments to study the choice of the regularization parameter δ and observed that any $\delta \geq 0.5$ is sufficient to fix the numerical instability issue and to improve robustness to short sample support. Therefore, we set $\delta = \delta_h = \delta_v = 1$ in our simulations. Moreover, preliminary simulations have shown that analytical and sample TLCMV converge within 2 and 3 iterations, respectively, over a wide SNR range. We therefore use these values when calculating the curves plotted in Figure 2.

Before discussing the output SINR performance, let us compare the computational performance of the LCMV-type beamformers listed in Tables 1, 2, and 3 as a function of the array size N. The number of multiplications required by each method to design the beamforming filter is plotted in Figure 2. The total number of multiplications carried out by TLCMV is calculated by multiplying the values listed in Table 2 by the number I of iterations until convergence. This result confirms that both proposed separable methods exhibit less computational complexity than the classical method. We note that TLCMV requires more calculations than KLCMV. This is because the former employs an iterative procedure and also the correlation matrices calculation is more involved. One can note that the sample solutions are more expensive than the analytical solutions since the correlation matrices estimation averages over $K = 10^4$ snapshots.

The proposed methods yield significant calculations saving in both analytical and sample scenarios (at least by one order of magnitude at $N = 256$), however, do they exhibit source recovery performance loss? To find out, let us consider

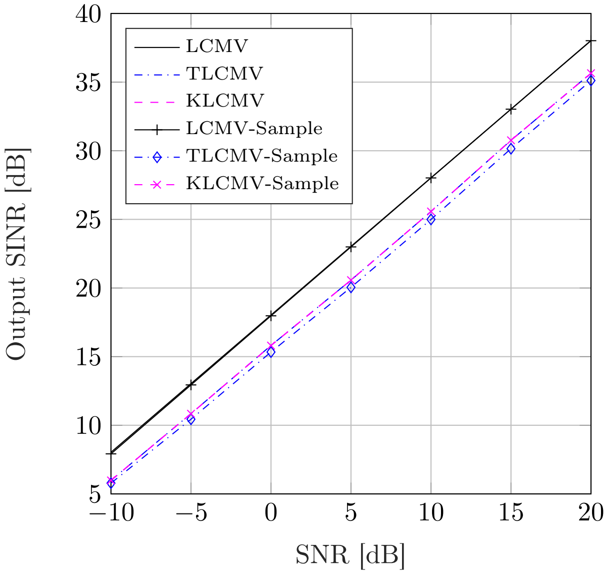

Figure 3: $N_h = N_v = 8$, $K = 10^4$ snapshots.

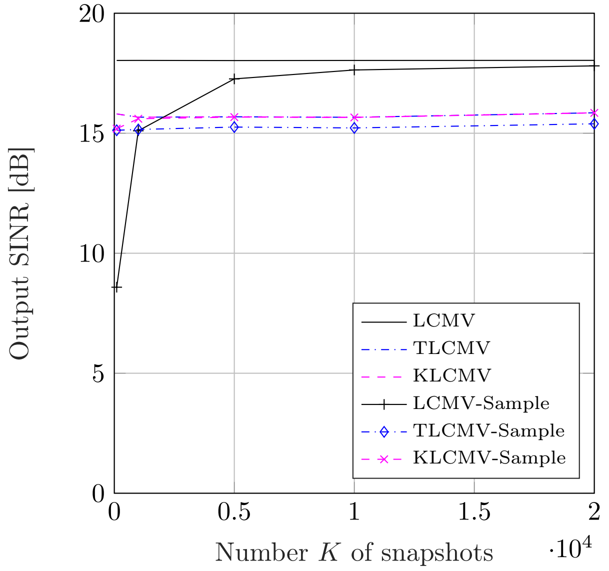

Figure 4: $N_h = N_v = 8$, SNR $=$ 0dB.

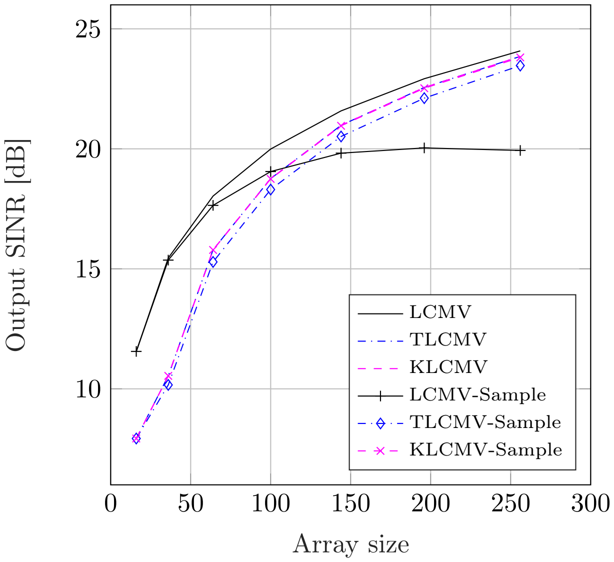

Figure 5: $K = 10^4$ snapshots, SNR = 0dB.

Figure 3, where the output SINR was obtained for different SNR values for an (8×8)-antennas array. We first notice that there is always a 3dB difference between the classical and separable approaches. Since the classical LCMV does not constrain the solution set to Kronecker filters, it is natural that it performs better than the other approach. Figure 3 also reveals that the source recovery performances of the TLCMV and KLCMV are practically the same (although KLCMV is much less expensive than TLCMV). Therefore, we conclude that the proposed separable solutions yield a 3dB due to the separability filter constraint.

To assess the influence of the number K of snapshots on the TLCMV-type beamformers performance, we computed the output SINR for different K values in Figure 4. The performance of the analytical filters is independent of K, so it serves as an upper bound for the corresponding sample solutions. This figure reveals that while source recovery quality of LCMV strongly depends on the number of snapshots, the separable filters are rather robust to it. This is because there are fewer statistics to estimate when designing the separable beamforming filter factors compared to the full-filter solution. For $K = 100$ at 0dB SNR, the separable filters yield 15dB SINR at the output, while LCMV-sample filter yields 8dB. This result accentuates the superior computational performance of the separable strategy for the sample beamformers.

We finally investigate the behavior of the LCMV-type beamformers for different array sizes, from $N = 16$ to $N = 256$, in Figure 5. Please note that $N = 4 \times 4$ is the smallest possible array for $U = 4$ wavefronts. Smaller arrays would not offer sufficient degrees of freedom to invert the correlation matrices required to compute the filter coefficients vector. This figure reveals that there is an important performance gap between the full and separable approaches when the array is relatively small. In this case, the inaccuracy of the small sub-filters limits the performance of the separable approaches. For $N = 16$ antennas, the LCMV filter produces relatively sharp beams towards the desired wavefront, while accurately placing nulls at the interfering directions, in contrast to the TLCMV and KLCMV sub-filters that roughly apply this spatial processing. Such an issue is discussed in [16] for MMSE-type beamformers. Therein, the array factors of separable methods are compared to those of the full MMSE beamformer to support this argument, and the same reasoning applies here. This performance gap, however, is decreased as the array size grows: the TLCMV and KLCMV sub-filters become more accurate, ameliorating the overall beamformer performance. Moreover, we observe that the behavior of sample LCMV saturates as N grows, whereas the sample separable filters approach the analytical LCMV. As the dimensions of the correlation matrix $\boldsymbol{R}_{xx}$ grow, more and more snapshots are required by LCMV to achieve a certain output SINR level, explaining its poor performance in the massive array scenario. On the other hand, for $N = 256$ antennas, for example, the separable filters need to estimate two (16×16)-dimensional correlation matrices.

We now present the results obtained with Frost-type beamformers, considering Monte Carlo experiments of 20 runs and setting the step-size factor to $\mu = 10^{-2}$. In Figure 6, the number of multiplications demanded by the beamforming methods are plotted. This figure indicates that the separable Frost algorithms require fewer calculations than the classical Frost algorithm. It also shows that the computational complexities of TFrost and KFrost are basically the same. But what about their convergence rate? To analyze that, we first consider Figure 7, where the output SINR evolution with the algorithm iteration is shown for SNR of 0dB and 30dB and $N = 64$ antennas. One can see that there is a 7dB gap between the separable and full Frost algorithms. This gap disappears at high SNR, however. Another interesting feature of the separable beamformers is that they exhibit small fluctuation (misadjustment) after convergence is achieved. We notice that both Frost implementation strategies exhibit the same convergence properties: a few hundred iterations to converge at low SNR, and very few steps to converge at high SNR. We extend these results to a scenario with $N = 14 \times 14$ antennas in Figure 8. At 0dB SNR, we observe that the separable Frost algorithms converge faster than the classical Frost's algorithm while achieving essentially the same SINR at the filter output. At 30dB SNR, however, this advantage fades out, and all Frost-type algorithms exhibit the same performance. From these experiments, we conclude that the separable Frost algorithms exhibit superior performance than the classical Frost's algorithm in the computational sense, and they yield better source performance, especially in the low SNR regime.

5. Conclusion

We introduced separable linearly constrained beamformers and analyzed their computational and source recovery performance through computer simula-

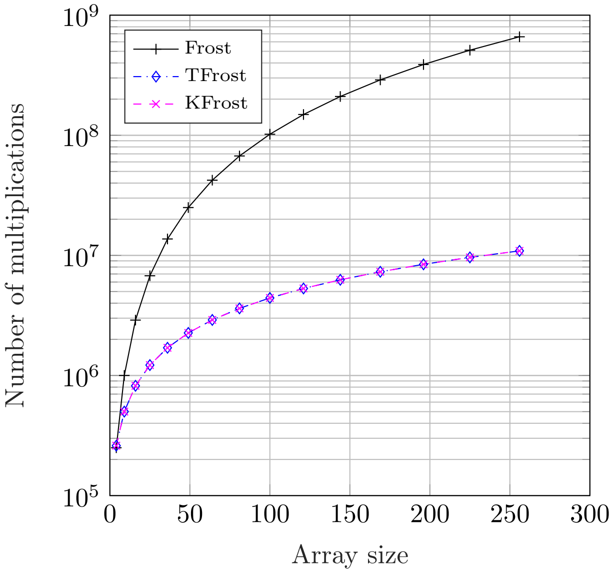

Figure 6: $K = 10^4$ snapshots.

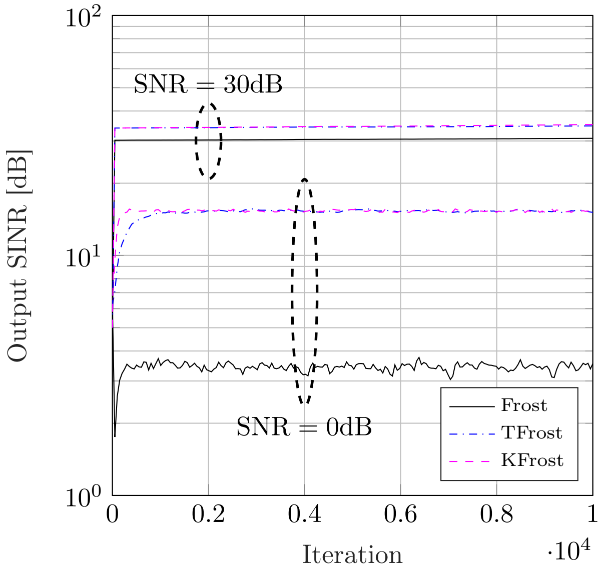

Figure 7: $N_h = N_v = 8$, $K = 10^4$ snapshots.

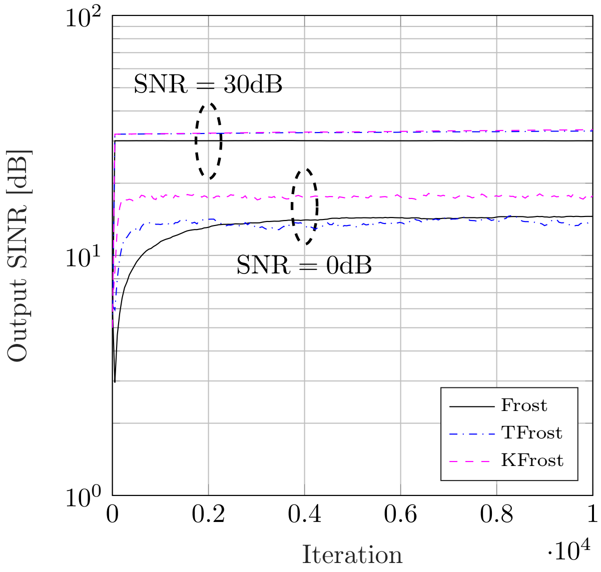

Figure 8: $N_h = N_v = 14$, $K = 10^4$ snapshots.

tions. We conducted a theoretical computational analysis of each beamforming method and discussed their convergence. Our simulation results emphasize the computational efficiency of our methods. The proposed KLCMV filter yields the best computational cost-source recovery trade-off among the LCMV-type beamformers. The proposed separable Frost-type algorithms exhibit better performance than the classical Frost's algorithm, especially in the low SNR regime.

References

[1] E. G. Larsson, O. Edfors, F. Tufvesson, T. L. Marzetta, Massive MIMO for next generation wireless systems, IEEE Communications Magazine 52 (2) (2014) 186–195. doi:10.1109/MCOM.2014.6736761.

[2] S. Schwarz, M. Rupp, Society in motion: challenges for LTE and beyond mobile communications, IEEE Communications Magazine 54 (5) (2016) 76–83.

[3] B. D. V. Veen, K. M. Buckley, Beamforming: a versatile approach to spatial filtering, IEEE ASSP Magazine 5 (2) (1988) 4–24. doi:10.1109/53.665.

[4] W. Liu, S. Weiss, Wideband beamforming: concepts and techniques, Vol. 17, John Wiley & Sons, 2010.

[5] S. Treitel, J. L. Shanks, The design of multistage separable planar filters, IEEE Transactions on Geoscience Electronics 9 (1) (1971) 10–27.

[6] C. Tang, C. Liu, L. Yuan, Z. Xing, High precision low complexity matrix inversion based on newton iteration for data detection in the massive MIMO, IEEE Communications Letters 20 (3) (2016) 490–493.

[7] F. Jiang, C. Li, Z. Gong, R. Su, Stair matrix and its applications to massive MIMO uplink data detection, IEEE Transactions on Communications 66 (6) (2018) 2437–2455. doi:10.1109/TCOMM.2017.2789211.

[8] X. Qin, Z. Yan, G. He, A near-optimal detection scheme based on joint steepest descent and jacobi method for uplink massive mimo systems, IEEE Communications Letters 20 (2) (2016) 276–279.

[9] B. Yin, M. Wu, J. R. Cavallaro, C. Studer, Conjugate gradient-based soft-output detection and precoding in massive MIMO systems, in: Proc. 2014 IEEE Global communications conference (GLOBECOM), pp. 3696–3701.

[10] H. L. Van Trees, Optimum array processing: Part IV of detection, estimation and modulation theory, Vol. 1, Wiley Online Library, 2002.

[11] L.-H. Lim, P. Comon, Blind multilinear identification, IEEE Transactions on Information Theory 60 (2) (2014) 1260–1280.

[12] M. Rupp, S. Schwarz, A tensor LMS algorithm, in: Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 3347–3351.

[13] M. Rupp, S. Schwarz, Gradient-based approaches to learn tensor products, in: Proc. 2015 23rd European Signal Processing Conference (EUSIPCO), 2015, pp. 2486–2490.

[14] L. N. Ribeiro, A. L. F. de Almeida, J. C. M. Mota, Identification of separable systems using trilinear filtering, in: Proc. 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015, pp. 189–192.

[15] L. N. Ribeiro, A. L. F. De Almeida, J. C. M. Mota, Tensor beamforming for multilinear translation invariant arrays, in: Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 2966–2970.

[16] L. N. Ribeiro, A. L. de Almeida, J. A. Nossek, J. C. M. Mota, Low-complexity separable beamformers for massive antenna array systems, arXiv preprint arXiv:1805.00176.

[17] C. Paleologu, J. Benesty, S. Ciochină, Linear system identification based on a Kronecker product decomposition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 26 (10) (2018) 1793–1808. doi:10.1109/TASLP.2018.2842146.

[18] C. Elisei-Iliescu, C. Stanciu, C. Paleologu, J. Benesty, C. Anghel, S. Ciochină, Efficient recursive least-squares algorithms for the identification of bilinear forms, Digital Signal Processing 83 (2018) 280–296.

[19] R. K. Miranda, J. P. C. da Costa, F. Roemer, A. L. de Almeida, G. Del Galdo, Generalized sidelobe cancellers for multidimensional separable arrays, in: Proc. 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015, pp. 193–196.

[20] L. Liu, J. Xie, L. Wang, Z. Zhang, Y. Zhu, Robust tensor beamforming for polarization sensitive arrays, Multidimensional Systems and Signal Processing (2018) 1–22.

[21] O. L. Frost, An algorithm for linearly constrained adaptive array processing, Proceedings of the IEEE 60 (8) (1972) 926–935.

[22] L.-H. Lim, P. Comon, Blind multilinear identification, IEEE Transactions on Information Theory 60 (2) (2014) 1260–1280.

[23] S. Miron, Y. Song, D. Brie, K. T. Wong, CANDECOMP/PARAFAC (CP) direction finding with multi-scale array, in: Proc. 2013 IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013, pp. 224–227.

[24] B. D. Carlson, Covariance matrix estimation errors and diagonal loading in adaptive arrays, IEEE Transactions on Aerospace and Electronic systems 24 (4) (1988) 397–401.

[25] H. Cox, R. Zeskind, M. Owen, Robust adaptive beamforming, IEEE Transactions on Acoustics, Speech, and Signal Processing 35 (10) (1987) 1365–1376.

[26] S. Liu, G. Trenkler, Hadamard, Khatri-Rao, Kronecker and other matrix products, International Journal of Information and Systems Sciences 4 (1) (2008) 160–177.

[27] D. P. Bertsekas, Nonlinear programming, Athena scientific Belmont, 1999.

[28] A. Yener, R. D. Yates, S. Ulukus, Interference management for CDMA systems through power control, multiuser detection, and beamforming, IEEE Transactions on Communications 49 (7) (2001) 1227–1239.

[29] R. Dallinger, M. Rupp, On robustness of coupled adaptive filters, Proc. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 3085–3088.

[30] E. Onggosanusi, M. S. Rahman, L. Guo, Y. Kwak, H. Noh, Y. Kim, S. Faxer, M. Harrison, M. Frenne, S. Grant, R. Chen, R. Tamrakar, a. Q. Gao, Modular and high-resolution channel state information and

beam management for 5G new radio, IEEE Communications Magazine 56 (3) (2018) 48–55. doi:10.1109/MCOM.2018.1700761.