Applications of Tensor Models in Wireless Communications and Mobile Computing

Lead Guest Editor: C. Alexandre R. Fernandes

Guest Editors: Jianhe Du, Alex da Silva, and André de Almeida

Applications of Tensor Models in Wireless Communications and Mobile Computing

Wireless Communications and Mobile Computing

Applications of Tensor Models in Wireless Communications and Mobile Computing

Lead Guest Editor: C. Alexandre R. Fernandes
Guest Editors: Jianhe Du, Alex da Silva, and André de Almeida

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 7393080, 2 pages
https://doi.org/10.1155/2020/7393080

Editorial

Applications of Tensor Models in Wireless Communications and Mobile Computing

C. Alexandre R. Fernandes,[1] Jianhe Du,[2] Alex P. da Silva,[3] and André L. F. de Almeida[1]

[1] *Federal University of Ceará (UFC), Fortaleza, Brazil*
[2] *Communication University of China, Beijing, China*
[3] *Alternative Energies and Atomic Energy Commission (CEA), Grenoble, France*

Correspondence should be addressed to C. Alexandre R. Fernandes; alexandrefernandes@ufc.br

Received 28 January 2020; Accepted 28 January 2020; Published 20 February 2020

Tensor decompositions, also named tensor factorizations, are very useful tools for representing and analyzing multidimensional data in a compact way. In some cases, tensor decompositions can be viewed as generalizations of matrix decompositions, such as the singular value decomposition (SVD), to higher-order arrays. In addition to enabling multidimensional data processing, tensor models have properties that become possible to solve undetermined solution problems under conditions more relaxed than conventional matrix approaches [1–3].

Recent developments in multilinear algebra have made possible the application of tensor analysis in several areas. In particular, tensor approaches have gained considerable space in wireless communications and mobile networks [4]. The proposition of new factorizations of tensors, as well the study of uniqueness and identifiability conditions, has a great potential to provide remarkable improvements in these fields.

Indeed, in the literature, one can find a wide range of applications of tensor and multilinear algebra in problems such as Internet traffic [5], missing data problems [6], big-data analysis [7], neuron networks [8], image processing [9], design of mobile communication systems [10, 11], and development of receivers for wireless systems [12].

Although much research has been done within this subject, many challenges are still to be explored. The proposition of new tensor decompositions, factorization algorithms, and mathematical properties has a great potential to bring significant impacts in several areas. The purpose of this special issue is to publish original efforts presenting recent advances in tensor decompositions with applications in communication systems and mobile computing.

B. Kurt et al. propose the use of tensor techniques for modeling network flows, which is exploited to solve recovery information problems. The novelty of this work consists in the first application of multiway methods for network monitoring, being an interesting alternative for recovering the true flow length distribution from the sampled traffic data. The data is modeled as a three-way array, with each characteristic being estimated using a nonnegative tensor factorization. The proposed tensor factorization scheme is validated with synthetic data and applied to real-world data with two different sampling methods.

P. R. B. Gomes et al. address the problem of joint downlink (DL) and uplink (UL) channel estimation for millimeter wave (mmWave) multiple-input multiple-output (MIMO) systems and develop two tensor-based semiblind receivers by capitalizing on the multilinear structure and sparse nature of the received signal at the BS equipped with a hybrid analog-digital beamforming (HB) architecture. This paper shows the benefits of jointly estimating both the DL and UL channels as a single problem by concentrating most of the processing burden for channel estimation at the BS side, which are attractive features for massive MIMO systems.

L. Liu et al. present a new algorithm to estimate the direction of arrival (DOA) and polarization parameters of signals impinging on an array with electromagnetic (EM) vector-sensors by exploiting the canonical polyadic decomposition (CPD) of tensors. In addition to spatial and temporal diversities, further information from the polarization domain is exploited. The authors show that the parameters can be estimated by virtue of the diversities of the spatial and polarization belonging to the factor matrices, rather than using conventional subspace methods.

M. T. de Oliveira et al. address a practical DOA problem by proposing a low-cost antenna array-based drone tracking device for outdoor environments. The proposed solution is divided into hardware and software parts. The hardware part of the proposed device is based on off-the-shelf components such as an omnidirectional antenna array, a 4-channel software defined radio (SDR) platform with carrier frequency ranging from 70 MHz to 6 GHz, a FPGA motherboard, and a laptop. The software part includes algorithms for calibration, model order selection (MOS), and DOA estimation, including specific preprocessing steps and a tensor-based estimator to increase the DOA accuracy.

With this special issue, we hope that readers will be interested in applications of tensor decompositions and they will find this issue helpful for their research.

Conflicts of Interest

The editors have no conflicts of interest to the assigned manuscripts when handling them and making decisions.

C. Alexandre R. Fernandes
Jianhe Du
Alex P. da Silva
André L. F. de Almeida

References

[1] A. Cichoki, D. Mandic, L. de Lathawer et al., "Tensor decompositions for signal processing applications: from two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[2] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[3] P. Comom, "Tensor decompositions—state of the art and applications," in *Proceedings of the MA Conference Mathematics in Signal Processing*, Warwick, UK, 2000.

[4] A. L. F. de Almeida, G. Favier, J. P. C. L. da Costa, and J. C. M. Mota, "Overview of tensor decompositions with applications to communications," in *Signals and Images: Advances and Results in Speech, Estimation, Compression, Recognition, filtering, and Processing*, pp. 325–356, CRC Book, Boca Raton, FL, USA, 2016.

[5] K. Xie, C. Peng, X. Wang et al., "Accurate recovery of internet traffic data under variable rate measurements," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1137–1150, 2018.

[6] K. Xie, L. Wang, X. Wang et al., "Accurate recovery of internet traffic data: a sequential tensor completion approach," *IEEE/ACM Transactions on Networking*, vol. 26, no. 2, pp. 793–806, 2018.

[7] A. Cichocki, "Tensor networks for big data analytics and large-scale optimization problems," 2014, https://arxiv.org/abs/1407.3124.

[8] W. Hong, W. Xu, J. Qi, and Y. Weng, "Neural tensor network for multi- label classification," *IEEE Access*, vol. 7, pp. 96936–96941, 2019.

[9] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5910–5922, 2019.

[10] G. Favier, C. A. R. Fernandes, and A. L. F. de Almeida, "Nested Tucker tensor decomposition with application to MIMO relay systems using tensor space-time coding (TSTC)," *Signal Processing*, vol. 128, pp. 318–331, 2016.

[11] D. S. Rocha, G. Favier, and C. A. R. Fernandes, "Tensor coding for three-hop MIMO relay systems," in *roceedings of the IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil, 2018.

[12] W. d. C. Freitas Jr., G. Favier, and A. L. F. de Almeida, "Sequential closed-form semiblind receiver for space-time coded multihop relaying systems," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1773–1777, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 4858137, 13 pages
https://doi.org/10.1155/2019/4858137

Research Article

Joint DL and UL Channel Estimation for Millimeter Wave MIMO Systems Using Tensor Modeling

Paulo R. B. Gomes,[1] **André L. F. de Almeida**,[1] **João Paulo C. L. da Costa**,[2] **and Rafael T. de Sousa Jr.**[2]

[1]*Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza-CE, Brazil*
[2]*Department of Electrical Engineering, University of Brasília, Brasília-DF, Brazil*

Correspondence should be addressed to André L. F. de Almeida; andre@gtel.ufc.br

Received 27 February 2019; Revised 7 June 2019; Accepted 25 July 2019; Published 15 September 2019

Academic Editor: Miguel Garcia-Pineda

In this paper, we address the problem of joint downlink (DL) and uplink (UL) channel estimation for millimeter wave (mmWave) multiple-input multiple-output (MIMO) systems. Assuming a closed-loop and multifrequency-based channel training framework in which pilot signals received by multiple antenna mobile stations (MSs) are coded and spread in the frequency domain via multiple adjacent subcarriers, we propose two tensor-based semiblind receivers by capitalizing on the multilinear structure and sparse feature of the received signal at the BS equipped with a hybrid analog-digital beamforming (HB) architecture. As a first processing stage, the joint estimation of the compressed DL and UL channel matrices can be obtained in an iterative way by means of an alternating least squares (ALS) algorithm that capitalizes on a parallel factors model for the received signals. Alternatively, for more restricted scenarios, a closed-form solution is also proposed. From the estimated effective channel matrices, the users' channel parameters such as angles of departure (AoD), angles of arrival (AoA), and path gains are then estimated in a second processing stage by solving independent compressed sensing (CS) problems (one for each MS). In contrast to the classical approach in the literature, in which the DL and UL channel estimation problems are usually considered as two separate problems, our idea is to jointly estimate both the DL and UL channels as a single problem by concentrating most of the processing burden for channel estimation at the BS side. Simulation results demonstrate that the proposed receivers achieve a performance close to the classical approach that is applied on DL and UL communication links separately, with the advantage of avoiding complex computations for channel estimation at the MS side as well as dedicated feedback channels for each MS, which are attractive features for massive MIMO systems.

1. Introduction

In recent years, millimeter wave (mmWave) massive multiple-input multiple-output (MIMO) technology has been a subject of increasing interest in both academia and industry for future wireless standards due to its great potential to provide substantial gains in data rates and energy efficiency. However, due to the severe path loss over the mmWave frequency bands, large antenna arrays should be deployed at the base station (BS) and mobile stations (MSs) to provide sufficient beamforming gain in mmWave MIMO scenarios [1]. In this context, the implementation of fully digital beamforming architectures becomes prohibitive due its expensive cost, hardware constraints, and power consumption of high-resolution analog-to-digital converters (ADC) and digital-to-analog converters (DAC) per antenna port [2, 3]. To overcome these practical limitations, hybrid analog-digital beamforming (HB) architectures that split the signal processing between analog and digital domains using a reduced number of radio frequency (RF) chains (assumed to be smaller than the number of antennas) have been investigated [4, 5]. In the HB architectures, the digital part performs baseband processing using microprocessors, while the analog part can be implemented at the RF domain using different analog approaches such as phase-shifter networks [6], switches [3, 7], or lenses [8].

To fully benefit from the beamforming gains in mmWave MIMO systems, an accurate channel estimation is crucial to realize the hybrid precoding designs in which the analog part is used to improve the signal power, while the digital part is designed to suppress interuser interferences [9–11]. For this purpose, several channel estimation techniques such as [12–16] have been proposed. The authors of [12] proposed an iterative method based on the least squares estimation (LSE) concept and sparse message passing (SMP) algorithm. In this method, the location of nonzero entries of the channel vector is detected through the SMP, while the LSE is used for estimating the channel coefficients at each iteration. Zhu et al. [13] proposed an auxiliary beam pair design for mmWave channel estimation in which the best auxiliary beam pair is fed back to the transmitter via a feedback channel. The method proposed by Ghauch et al. [14] consists of a subspace-based approach that exploits the channel reciprocity in time-division duplexing (TDD) MIMO systems for hybrid precoding design. It iteratively estimates the dominant singular modes of the channel instead of the entire channel. In contrast, the works [15, 16] explore the angular sparsity of mmWave channels and use compressed sensing (CS) theory to estimate only the channel parameters from which the mmWave channel can be reconstructed. In [15], the angular spreads over the angle of arrival (AoA) and angle of departure (AoD) are considered in the channel modeling, while the low-rank structure of the channel is exploited to reduce the number of samples needed to recover the mmWave channel. Similarly, but disregarding the angular spreads in the spatial domains, the method in [16] uses the 2D unitary ESPRIT algorithm for spatial parameters estimation, while the path gains are estimated by means of the LS criterion.

The researchers [17–21] have proposed CS-based and tensor-based channel estimators for mmWave MIMO systems, respectively. They assume the conventional channel training framework, where the DL and UL channel estimation problems are treated separately (as two decoupled procedures at the MS and BS, respectively). In particular, for frequency division duplexing (FDD) systems, where channel estimation is usually carried at the power-limited MS side, computational complexity plays a significant role due to the large number of channel coefficients to be estimated. An interesting approach to deal with this problem exploits the poor scattering nature of the mmWave channels via CS techniques [22, 23]. For instance, in [17, 18], the intrinsic sparse feature of the mmWave channel is exploited and CS-based channel estimation algorithms are formulated. However, the adaptive algorithm proposed in [17] can be applied to estimate the DL or UL channel separately. In [19], a layered pilot transmission scheme is proposed to UL channel estimation, while [20, 21] exploit the DL communication of wideband mmWave channels. The main idea in [21] is to divide the overall channel estimation problem into three smaller CS subproblems via tensor-based modeling to estimate the channel parameters (AoDs, AoAs, and delays) with less computational complexity. The system model is formulated as a parallel factors (PARAFAC) decomposition [24], and tensor-based algorithms combined with CS tools are proposed to solve the channel estimation problem. However, Zhou et al. [19] only considered the UL channel estimation, while Zhou et al. and Araújo and de Almeida [20, 21] focused on the DL case. On the other hand, Shen et al. [25] proposed a closed-loop based training framework, where the MSs directly feed the received pilots back to the BS without channel estimation. Therein, a simplified approach is adopted, in which the UL channel is modeled as an additive white Gaussian noise (AWGN) term. Moreover, in [25], only the DL channel could be estimated. Different from [25], we are interested in a joint estimation of the UL and DL channels, and a more realistic multipath channel model that fits to mmWave MIMO scenario and HB architecture is considered.

In this paper, we study the problem of joint DL and UL channel estimation in the context of mmWave MIMO systems that employ HB architectures. Initially, we propose a novel closed-loop and multifrequency-based channel training framework in which the pilot signals received by multiple MSs are coded and spread in the frequency domain and then fed back to the BS over the same UL resources. Making use of the proposed framework for channel estimation, the received closed-loop signal at the BS can be modeled as a three-way array (i.e., a third-order tensor) that follows a PARAFAC model. By capitalizing both on the multidimensional and sparse structures of the received signal, we propose two tensor-based semiblind receivers for joint DL and UL channel estimation. The first receiver is an iterative solution based on the alternating least squares (ALS) algorithm [26]. The second is a closed-form solution based on the least squares Khatri-Rao factorization (LS-KRF) algorithm [27]. In the proposed receivers, we first obtain joint estimates of the compressed DL and UL channel matrices. Then, we exploit the sparse representation of the DL and UL channels to individually recover the channel parameters (AoDs, AoAs, and path gains) of each user via CS-based techniques. The proposed framework allows concentrating most of the processing burden for channel estimation at the BS side, i.e., avoiding unnecessary computational overhead for channel estimation at the power-limited MS side. Our simulation results reveal that the proposed receivers achieve a performance close to the classical framework that treats the estimation of DL and UL channels as separate problems.

In summary, the main contributions of this paper can be listed as follows:

(i) We propose a novel closed-loop and multifrequency-based channel training framework for channel estimation that focuses jointly on the DL and UL communication links. The proposed framework concentrates the processing burden for joint channel estimation at the BS, avoiding processing with high computational cost at the MS side.

(ii) We show that, by making use of the proposed framework for channel estimation, the received closed-loop signal can be modeled as a third-order tensor that follows a PARAFAC model. Then, we formulate two tensor-based semiblind receivers (iterative and closed-form ones) for joint DL and UL channel estimation by capitalizing on a tensor structure of the received closed-loop signal.

(iii) We study the identifiability issues under which the DL and UL channel matrices can be jointly and uniquely estimated using the proposed receivers. Useful lower bounds on the number of subcarriers required to accomplish the joint channel estimation are derived.

The rest of this paper is structured as follows: In Section 2, we provide as a presentation complement some important tensor definitions, tensor algebra operations, and a brief overview on the PARAFAC decomposition. In Section 3, we present the proposed channel training framework, and the system and channel models. Section 4 formulates the two proposed tensor-based semiblind receivers for joint DL and UL channel estimation. In Section 5, we analyze the identifiability conditions of the proposed receivers. Simulation results are provided in Section 6. Conclusions and perspectives for future work are drawn in Section 7.

1.1. Notation and Properties. Scalars, column vectors, matrices, and tensors are denoted by nonbold lowercase letters a, bold lowercase letters $\mathbf{a}$, bold uppercase letters $\mathbf{A}$, and calligraphic uppercase letters $\mathscr{A}$, respectively. The superscripts $\{\cdot\}^{\mathrm{T}}$, $\{\cdot\}^{*}$, $\{\cdot\}^{\mathrm{H}}$, and $\{\cdot\}^{\dagger}$ denote the transpose, complex conjugate, conjugate transpose, and pseudoinverse operations. $\|\cdot\|_{\mathrm{F}}$ represents the Frobenius norm of a matrix or tensor. The (i, r)-th entry of $\mathbf{A}$ is denoted by $[\mathbf{A}]_{i,r}$. The operator $\operatorname{diag}(\mathbf{a})$ converts $\mathbf{a}$ into a diagonal matrix, while $D_i(\mathbf{A})$ consists in a diagonal matrix formed by the i-th row of $\mathbf{A}$. $\operatorname{vec}(\mathbf{A})$ converts $\mathbf{A}$ to a vector $\mathbf{a}$ by stacking its columns on top of each other, while $\operatorname{unvec}_{I\times R}(\mathbf{a})$ converts $\mathbf{a} \in \mathbb{C}^{IR}$ to a matrix $\mathbf{A} \in \mathbb{C}^{I\times R}$. $\operatorname{vecd}(\mathbf{A})$ converts the diagonal elements of $\mathbf{A}$ into a vector. $\circ$ denotes the outer product operator. The Kronecker and Khatri-Rao products are denoted by $\otimes$ and $\diamond$, respectively. The Khatri-Rao product between the matrices $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_R] \in \mathbb{C}^{I\times R}$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_R] \in \mathbb{C}^{J\times R}$ corresponds to a column-wise Kronecker product, i.e.,

$$\mathbf{A} \diamond \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, \ldots, \mathbf{a}_R \otimes \mathbf{b}_R] \in \mathbb{C}^{IJ\times R}. \quad (1)$$

We shall make use of the following two properties of the Kronecker and Khatri-Rao products:

$$\mathbf{AC} \diamond \mathbf{B\,D} = (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \diamond \mathbf{D}), \quad (2)$$

$$\operatorname{vec}\left(\mathbf{ABC}^{\mathrm{T}}\right) = (\mathbf{C} \diamond \mathbf{A})\operatorname{vecd}(\mathbf{B}), \quad (3)$$

$$\mathbf{a} \otimes \mathbf{b} = \operatorname{vec}(\mathbf{b} \circ \mathbf{a}) \in \mathbb{C}^{IJ}, \quad (4)$$

where $\mathbf{B}$ is assumed to be a diagonal matrix in (3). In both cases, the matrices have compatible dimensions.

2. Tensor Preliminaries

In order to facilitate the presentation of the proposed receivers, we provide below a brief overview on some important tensor definitions and tensor algebra operations. We also introduce the PARAFAC decomposition and its different representation forms.

2.1. Tensor Definitions and Basic Operations. Throughout this paper, the definitions and operations involving tensors are in accordance with [28, 29]. A tensor is defined here as a multidimensional array. The order of a tensor corresponds to the number of dimensions. It can be seen as a generalization of a matrix to higher-order dimensions. For instance, a scalar is a tensor of order 0, a vector is a tensor of order 1, and a matrix is a tensor of order 2. An n-mode fiber of a tensor is a vector obtained by varying the n-th index and keeping all the other indexes fixed. Slices are two-dimensional sections of a tensor, obtained by fixing all but two indices. The operator $\mathbf{A} \sqcup_n \mathbf{B}$ denotes the concatenation of two matrices along the n-th mode of a tensor. The 1 mode, 2 mode, and 3 mode unfolding matrices of the third-order tensor $\mathscr{X} \in \mathbb{C}^{I_1\times I_2\times I_3}$, denoted by $[\mathscr{X}]_{(1)} \in \mathbb{C}^{I_1\times I_2 I_3}$, $[\mathscr{X}]_{(2)} \in \mathbb{C}^{I_2\times I_1 I_3}$, and $[\mathscr{X}]_{(3)} \in \mathbb{C}^{I_3\times I_1 I_2}$, are obtained by collecting all the 1 mode, 2 mode, and 3 mode fibers to be columns of the resulting matrices, respectively. The n-mode product between the tensor $\mathscr{X}$ and a matrix $\mathbf{A} \in \mathbb{C}^{R_n\times I_n}$ is denoted by $\mathscr{Y} = \mathscr{X} \times_n \mathbf{A}$, which is equivalent in a matrix fashion to $[\mathscr{Y}]_{(n)} = \mathbf{A}[\mathscr{X}]_{(n)}$ $(n = 1, 2, 3)$.

2.2. PARAFAC Decomposition. By definition, the PARAlell FACtor (PARAFAC) analysis decomposition of a third-order tensor $\mathscr{X} \in \mathbb{C}^{I_1\times I_2\times I_3}$, introduced by [24], is the factorization of $\mathscr{X}$ in a sum of R third-order rank-one tensors each one being formed by the outer product of three vectors. Mathematically, the PARAFAC decomposition of $\mathscr{X} \in \mathbb{C}^{I_1\times I_2\times I_3}$ is given by

$$\mathscr{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)}, \quad (5)$$

where R is the rank of the PARAFAC decomposition and is defined as the minimum number of rank-one tensors for which $\mathscr{X}$ holds exactly. The vector $\mathbf{a}_r^{(n)} \in \mathbb{C}^{I_n}$ denotes the r-th column of the factor matrix $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \ldots, \mathbf{a}_R^{(n)}] \in \mathbb{C}^{I_n\times R}$ along the n-th mode $(n = 1, 2, 3)$.

The PARAFAC decomposition can also be represented in terms of the frontal slices of $\mathscr{X}$ as follows:

$$\mathbf{X}_{i_3} = \mathbf{A}^{(1)} D_{i_3}\left(\mathbf{A}^{(3)}\right)\mathbf{A}^{(2)\mathrm{T}} \in \mathbb{C}^{I_1\times I_2}, \quad (6)$$

for $i_3 = 1, \ldots, I_{3.}$

By using n-mode product notation, equations (5) and (6) can be written as

$$\mathscr{X} = \mathscr{I}_{3,R} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}, \quad (7)$$

where $\mathscr{I}_{3,R}$ denotes a third-order identity tensor of size $R \times R \times R$. Its elements are equal to 1 when all indices are equal and 0 elsewhere.

The 1 mode, 2 mode, and 3 mode unfolding matrices of $\mathcal{X}$ admit the following factorizations with respect to the factor matrices $\mathbf{A}^{(n)}$ $(n = 1, 2, 3)$:

$$[\mathcal{X}]_{(1)} = \mathbf{A}^{(1)}\left(\mathbf{A}^{(3)} \diamond \mathbf{A}^{(2)}\right)^{\mathrm{T}}, \tag{8}$$

$$[\mathcal{X}]_{(2)} = \mathbf{A}^{(2)}\left(\mathbf{A}^{(3)} \diamond \mathbf{A}^{(1)}\right)^{\mathrm{T}}, \tag{9}$$

$$[\mathcal{X}]_{(3)} = \mathbf{A}^{(3)}\left(\mathbf{A}^{(2)} \diamond \mathbf{A}^{(1)}\right)^{\mathrm{T}}. \tag{10}$$

3. System and Channel Models

In this section, we introduce the proposed closed-loop and multifrequency channel training framework. Then, we formulate our DL and UL signal models. In addition, the considered mmWave massivo MIMO channel model is also presented.

3.1. Downlink Signal Model. Consider a wireless communication system operating in the FDD mode, where a BS equipped with N_{BS} antennas serves simultaneously U MSs equipped with N_{MS} antennas. We assume that the BS employs a hybrid beamforming architecture using M_{RF} RF chains. Due to the different instants of time dedicated to DL and UL communications, the beamforming matrix associated with DL transmission is denoted by $\mathbf{W} = \mathbf{W}_{\mathrm{RF}}\mathbf{W}_{\mathrm{BB}} \in \mathbb{C}^{N_{\mathrm{BS}} \times M_{\mathrm{RF}}}$, while the beamforming matrix associated with UL reception is denoted by $\mathbf{F} = \mathbf{F}_{\mathrm{RF}}\mathbf{F}_{\mathrm{BB}} \in \mathbb{C}^{N_{\mathrm{BS}} \times M_{\mathrm{RF}}}$. Note that equal beamforming matrices can also be considered in the transmission and reception phases without loss of generality. The BS transmits a length-T pilot sequence $\mathbf{s}_p \in \mathbb{C}^{T}$ over the p-th spatial direction using the beamforming vector $\mathbf{w}_p \in \mathbb{C}^{N_{\mathrm{BS}}}$ $(p = 1, \ldots, P$ and $P \leq M_{\mathrm{RF}})$. The received signal at the u-th MS over P different directions is given by

$$\mathbf{Y}_u = \mathbf{H}_u\mathbf{W}\mathbf{S} + \mathbf{V}_u^{(\mathrm{DL})} \in \mathbb{C}^{N_{\mathrm{MS}} \times T}, \tag{11}$$

where $\mathbf{H}_u \in \mathbb{C}^{N_{\mathrm{MS}} \times N_{\mathrm{BS}}}$ denotes the DL channel matrix associated with the u-th MS, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_P] \in \mathbb{C}^{N_{\mathrm{BS}} \times P}$ denotes the transmission beamforming matrix, $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_P]^{\mathrm{T}} \in \mathbb{C}^{P \times T}$ concatenates the pilot sequences to be sent by each transmission beam. The matrix $\mathbf{V}_u^{(\mathrm{DL})} \in \mathbb{C}^{N_{\mathrm{MS}} \times T}$ is the additive white Gaussian noise (AWGN) term at the u-th MS.

During the training phase, we assume identity matrices for the digital beamforming matrices, while the analog beamforming matrices have constant unit modulus entries with random phases. Thus, the entries of $\mathbf{W}$ and $\mathbf{F}$ are chosen uniformly from a unit circle scaled by a constant $1/\sqrt{N_{\mathrm{BS}}}$, i.e.,

$$\begin{aligned} [\mathbf{W}]_{i,j} &= \frac{1}{\sqrt{N_{\mathrm{BS}}}}e^{j\vartheta_{i,j}}, \\ [\mathbf{F}]_{i,j} &= \frac{1}{\sqrt{N_{\mathrm{BS}}}}e^{j\varphi_{i,j}}, \end{aligned} \tag{12}$$

where $\vartheta_{i,j}$ and $\varphi_{i,j} \in [-\pi, \pi]$ follow a uniform distribution. Since this work deals with the channel estimation problem, the optimum design of the beamforming matrices is not addressed here.

3.2. Uplink Signal Model. The pilot signal (11) received at the u-th MS is fed back to the BS after a multifrequency coding operation (i.e., no channel estimation is done at the MS side). More specifically, we assume that $\mathbf{Y}_u$ $(u = 1, \ldots, U)$ is coded and spread in the frequency domain across K adjacent subcarriers over which the fading channel is assumed to be constant. The coded signal of the u-th MS transmitted at the k-th subcarrier can be expressed as

$$\mathbf{Y}_{k,u} = \mathrm{diag}\left(\mathbf{c}_{k,u}\right)\mathbf{Y}_u \in \mathbb{C}^{N_{\mathrm{MS}} \times T}, \tag{13}$$

where $\mathbf{c}_{k,u} \in \mathbb{C}^{N_{\mathrm{MS}}}$ denotes a known code vector associated with the k-th subcarrier and used by u-th MS. It is worth noting that the coding vectors used by the different MSs do not need to be orthogonal. As will be discussed later, the linear independence assumption is enough. In practice, this means that these codes can be locally generated at each MS as pseudorandom sequences, i.e., no prior signaling and coordination between MSs is necessary [30].

In the UL communication, the BS employs Q beamforming vectors $\mathbf{f}_q \in \mathbb{C}^{N_{\mathrm{BS}}}$ $(q = 1, \ldots, Q$ and $Q \leq M_{\mathrm{RF}})$ to receive the coded uplink pilot signals over a set of Q different spatial directions. The received closed-loop signal at the BS associated with the k-th subcarrier is then given by

$$\begin{aligned} \mathbf{X}_k &= \mathbf{F}^{\mathrm{H}}\left(\sum_{u=1}^{U}\mathbf{G}_u\mathbf{Y}_{k,u} + \mathbf{V}^{(\mathrm{UL})}\right) \\ &= \left(\mathbf{F}^{H}\mathbf{G}_{\mathrm{e}}\right)\mathrm{diag}\left(\mathbf{c}_k\right)\mathbf{Y}_{\mathrm{e}}^{\mathrm{T}} + \mathbf{F}^{\mathrm{H}}\mathbf{V}^{(\mathrm{UL})} \in \mathbb{C}^{Q \times T}, \end{aligned} \tag{14}$$

where $\mathbf{G}_{\mathrm{e}} = [\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_U] \in \mathbb{C}^{N_{\mathrm{BS}} \times UN_{\mathrm{MS}}}$ denotes an extended version of the UL channel matrix that concatenates the U UL channel matrices $\mathbf{G}_u \in \mathbb{C}^{N_{\mathrm{BS}} \times N_{\mathrm{MS}}}$ $(u = 1, \ldots, U)$ of all MSs, $\mathbf{Y}_{\mathrm{e}} = [\mathbf{Y}_1^{\mathrm{T}}, \mathbf{Y}_2^{\mathrm{T}}, \ldots, \mathbf{Y}_U^{\mathrm{T}}] \in \mathbb{C}^{T \times UN_{\mathrm{MS}}}$ denotes an extended matrix that concatenates the feedback signals sent by all MSs, $\mathbf{c}_k = [\mathbf{c}_{k,1}^{\mathrm{T}}, \mathbf{c}_{k,2}^{\mathrm{T}}, \ldots, \mathbf{c}_{k,U}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{UN_{\mathrm{MS}}}$ is an extended code vector that contains the coding vectors of all MSs with respect to the k-th subcarrier, and $\mathbf{F}^{\mathrm{H}}\mathbf{V}^{(\mathrm{UL})}$ represents the filtered noise term at the output of the RF chains.

3.3. Conventional × Proposed Channel Training Framework. The conventional framework for channel estimation, summarized in Figure 1, assumes channel reciprocity in TDD or treat the DL and UL channel estimation as two separated problems in FDD, i.e., solved independently at the MS and BS, respectively. For the DL channel estimation, the BS first sends pilot signals to all MSs. At the MS side, the DL channel estimation can be performed by means of the state-of-the-art least squares (LS), minimum mean square error (MMSE), or CS-based estimators. Then, the estimated DL channel is reported back to the BS via dedicated UL resources [30]. To solve the UL channel estimation problem, a pilot signal is sent to the BS by each MS. Finally, the UL channels of all MSs are estimated by the BS. In practice, the DL channel estimated from UL pilots under the reciprocity assumption may not be accurate due to radio frequency distortions or a

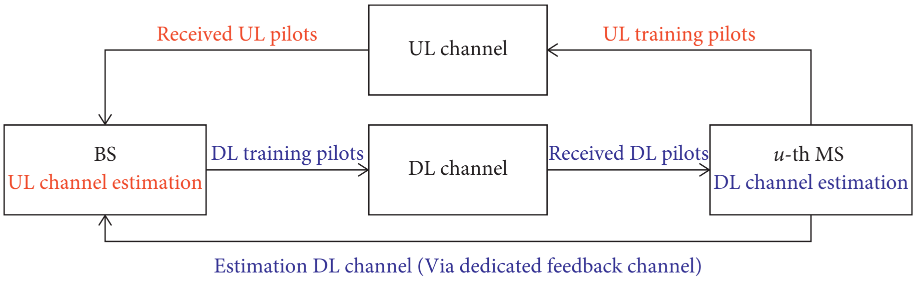


Figure 1: Conventional training framework. The DL and UL channel estimation problems are solved independently. The BS first transmits pilot sequences. Then, the DL channel is estimated at the MS side. The estimated DL channel is fed back to the BS via dedicated uplink resources. The UL channel is estimated at the BS side. The text in blue refer to DL communication, while the text in red refer to UL communication.

different carrier frequency such as in FDD. In addition, the conventional framework for channel estimation may imply a high computational complexity at the MS side, especially for power-limited devices.

In the proposed framework, summarized in Figure 2, no processing for channel estimation is performed at the MSs side. In contrast to the conventional approach, the received pilot signals at each MS are fed back to the BS after a multifrequency coding operation across a set of adjacent subcarriers. After this closed-loop procedure, the joint DL and UL channel estimation can be performed at the BS from the received signal given in (14). We can note that the proposed framework alleviates computational overhead due to channel estimation at the power-limited devices, by shifting this processing burden to the BS side. Furthermore, it also relaxes channel reciprocity assumptions since the DL and UL channels can be estimated jointly from (14).

After estimating the UL and DL channels, the BS may report the estimated parameters (AoDs, AoAs, and path gains) to the MS. Then, each MS can rebuild an estimation of the DL channel (according to procedure presented in Section 4.5) before decoding the information data. This approach reduces the overhead since a number of DL channel parameters reported to the MS is much smaller with respect to the size of the DL channel matrix in an mmWave MIMO system.

3.4. Channel Model.

In (11) and (14), we consider a general formulation in which the DL and UL channels are completely independent. In other words, the channels do not share any reciprocity in the angular or path gain domains. We also assume that the UL channels are constant across the K adjacent subcarriers used in the multifrequency coding operation. Due to the severe path loss, mmWave channels can be modeled by a narrow-band clustered channel model with few L_u dominant paths between the u-th MS and the BS. The DL channel matrix $\mathbf{H}_u \in \mathbb{C}^{N_{\mathrm{MS}} \times N_{\mathrm{BS}}}$ associated with the u-th MS can be written as [19]

$$\mathbf{H}_u = \sum_{l=1}^{L_u} \alpha_{u,l} \mathbf{a}_{\mathrm{MS}}\left(\theta_{u,l}\right) \mathbf{a}_{\mathrm{BS}}\left(\phi_{u,l}\right)^{\mathrm{T}}, \tag{15}$$

where $\alpha_{u,l}$ denotes the complex path gain of the u-th MS related to the l-th path in the DL communication. The path gains are modeled as circular symmetric Gaussian random variables with zero mean and unit variance. $\mathbf{a}_{\mathrm{MS}}(\theta_{u,l}) \in \mathbb{C}^{N_{\mathrm{MS}}}$ and $\mathbf{a}_{\mathrm{BS}}(\phi_{u,l}) \in \mathbb{C}^{N_{\mathrm{BS}}}$ are the antenna array response vectors evaluated at the angle of arrival $\theta_{u,l}$ and angle of departure $\phi_{u,l}$ uniformly distributed in the interval $[0, 2\pi]$. Throughout this paper, we assume uniform linear arrays (ULAs) at the BS and MSs. However, the proposed method can be applied to arbitrary array geometries without loss of generality. For ULA configurations with interantennas spacing equals to $d = \lambda/2$, where λ denotes the wavelength of the signal, the array response vectors at the MS and BS can be formulated as

$$\begin{aligned}
\mathbf{a}_{\mathrm{MS}}\left(\theta_{u,l}\right) &= \frac{1}{\sqrt{N_{\mathrm{MS}}}}\left[1, e^{j\pi\cos\theta_{u,l}}, \dots, e^{j\pi\left(N_{\mathrm{MS}}-1\right)\cos\theta_{u,l}}\right]^{\mathrm{T}}, \\
\mathbf{a}_{\mathrm{BS}}\left(\phi_{u,l}\right) &= \frac{1}{\sqrt{N_{\mathrm{BS}}}}\left[1, e^{j\pi\cos\phi_{u,l}}, \dots, e^{j\pi\left(N_{\mathrm{BS}}-1\right)\cos\phi_{u,l}}\right]^{\mathrm{T}}.
\end{aligned} \tag{16}$$

In matrix form, $\mathbf{H}_u$ can be rewritten as

$$\mathbf{H}_u = \mathbf{A}_{\mathrm{MS}} \operatorname{diag}(\boldsymbol{\alpha}) \mathbf{A}_{\mathrm{BS}}^{\mathrm{T}}, \tag{17}$$

where $\boldsymbol{\alpha} = \sqrt{N_{\mathrm{MS}} N_{\mathrm{BS}} / L_u}\,[\alpha_{u,1}, \alpha_{u,2}, \dots, \alpha_{u,L_u}]^{\mathrm{T}} \in \mathbb{C}^{L_u}$ denotes the vector that contains the L_u path gains in the DL. The array response matrices $\mathbf{A}_{\mathrm{MS}} \in \mathbb{C}^{N_{\mathrm{MS}} \times L_u}$ and $\mathbf{A}_{\mathrm{BS}} \in \mathbb{C}^{N_{\mathrm{BS}} \times L_u}$ at the MS and BS are expressed as

$$\begin{aligned}
\mathbf{A}_{\mathrm{MS}} &= \left[\mathbf{a}_{\mathrm{MS}}\left(\theta_{u,1}\right), \mathbf{a}_{\mathrm{MS}}\left(\theta_{u,2}\right), \dots, \mathbf{a}_{\mathrm{MS}}\left(\theta_{u,L_u}\right)\right], \\
\mathbf{A}_{\mathrm{BS}} &= \left[\mathbf{a}_{\mathrm{BS}}\left(\phi_{u,1}\right), \mathbf{a}_{\mathrm{BS}}\left(\phi_{u,2}\right), \dots, \mathbf{a}_{\mathrm{BS}}\left(\phi_{u,L_u}\right)\right].
\end{aligned} \tag{18}$$

The UL channel matrix $\mathbf{G}_u \in \mathbb{C}^{N_{\mathrm{BS}} \times N_{\mathrm{MS}}}$ from the u-th MS to the BS can be represented in a similar way. We define $\mathbf{G}_u$ as follows

$$\mathbf{G}_u = \mathbf{A}_{\mathrm{BS}} \operatorname{diag}(\boldsymbol{\beta}) \mathbf{A}_{\mathrm{MS}}^{\mathrm{T}}, \tag{19}$$

where $\mathbf{A}_{\mathrm{BS}} \in \mathbb{C}^{N_{\mathrm{BS}} \times M_u}$ and $\mathbf{A}_{\mathrm{MS}} \in \mathbb{C}^{N_{\mathrm{MS}} \times M_u}$ are now functions of the spatial parameters in the UL, while $\boldsymbol{\beta} = \sqrt{N_{\mathrm{MS}} N_{\mathrm{BS}} / M_u}\,[\beta_{u,1}, \beta_{u,2}, \dots, \beta_{u,M_u}]^{\mathrm{T}} \in \mathbb{C}^{M_u}$ denotes the vector that contains the M_u path gains of the UL channel.

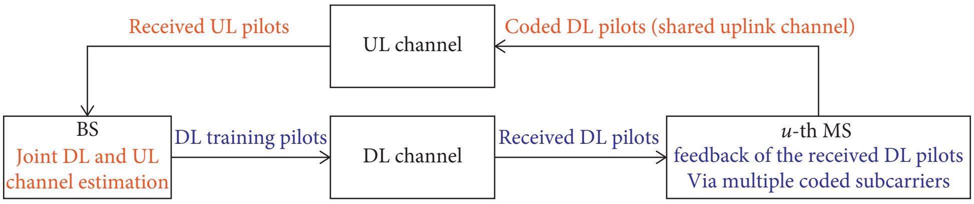

Received UL pilots
UL channel
Coded DL pilots (shared uplink channel)
BS
Joint DL and UL channel estimation
DL training pilots
DL channel
Received DL pilots
u-th MS
feedback of the received DL pilots
Via multiple coded subcarriers

Figure 2: Proposed closed-loop and multifrequency-based training framework. The DL and UL channels are jointly estimated. The BS first transmits pilot sequences. The MSs encode the received pilots and then feed them back to the BS. The BS jointly estimates the DL and UL channels. The text in blue refer to DL communication, while the text in red refer to UL communication.

4. Proposed Tensor-Based Semiblind Receivers for Joint DL and UL Channel Estimation

Our aim is to jointly estimate the DL and UL channel matrices $\mathbf{H}_u$ and $\mathbf{G}_u$ of each MS ($u = 1, \dots, U$) by solving a multiuser channel estimation problem at the BS. To this end, we first rewrite the received closed-loop signal (14) using a tensor formalism. Then, by capitalizing on its multidimensional structure, we formulate two tensor-based semiblind receivers to initially obtain the estimates for the compressed channel matrices. As a final step, we exploit the sparse representation of the DL and UL channels to estimate their parameters by decoupling the multiuser channel estimation problem into multiple single-user ones solved in a parallel way via separate CS problems.

4.1. PARAFAC Modeling. According to (6) and (7), the noiseless term in the received closed-loop signal (14) can be interpreted as the k-th frontal slice of the following third-order PARAFAC decomposition:

$$\mathcal{X} = \mathcal{I}_{3,U\cdot N_{\text{MS}}} \times_1 \left(\mathbf{F}^{\text{H}}\mathbf{G}_{\text{e}}\right) \times_2 \mathbf{Y}_{\text{e}} \times_3 \mathbf{C} \in \mathbb{C}^{Q\times T\times K}, \tag{20}$$

obtained by concatenating the K signal matrices $\{\mathbf{X}_k\}_{k=1}^{K}$ associated with the different adjacent subcarriers along the third mode of $\mathcal{X}$, i.e.,

$$\mathcal{X} = \mathbf{X}_1 \sqcup_3 \mathbf{X}_2 \sqcup_3 \dots \sqcup_3 \mathbf{X}_K. \tag{21}$$

By analogy with (7), the following correspondence holds:

$$\begin{gathered}\left(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}\right) \longleftrightarrow \left(\mathbf{F}^{\text{H}}\mathbf{G}_{\text{e}}, \mathbf{Y}_{\text{e}}, \mathbf{C}\right), \\ \left(I_1, I_2, I_3, R\right) \longleftrightarrow \left(Q, T, K, UN_{\text{MS}}\right).\end{gathered} \tag{22}$$

The three dimensions, or modes, of $\mathcal{X}$ stands for the number of receive beams, pilot sequence length, and number of subcarriers. The matrix $\mathbf{F}$ can be seen as a compression matrix associated with the first mode of $\mathcal{X}$ which reduces the size of the first mode from N_{BS} to Q RF chains (i.e., number of beams). The k-th row of the multifrequency coding matrix $\mathbf{C} \in \mathbb{C}^{K\times UN_{\text{MS}}}$ contains the code values used by the U MSs at the k-th subcarrier, i.e.,

$$\mathbf{C} = \left[\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\right]^{\text{T}} \in \mathbb{C}^{K\times UN_{\text{MS}}}. \tag{23}$$

Estimates of the channel parameters (AoDs, AoAs, and path gains) that build up the channel matrices $\{\mathbf{H}_u\}_{u=1}^{U}$ and $\{\mathbf{G}_u\}_{u=1}^{U}$ can be obtained by fitting the noisy version of $\mathcal{X}$ to a PARAFAC decomposition. In the following, we formulate the first stage of the proposed receivers that consists of estimating the factor matrices $\mathbf{F}^{\text{H}}\mathbf{G}_{\text{e}} \in \mathbb{C}^{Q\times UN_{\text{MS}}}$ and $\mathbf{Y}_{\text{e}} \in \mathbb{C}^{T\times UN_{\text{MS}}}$ in an iterative or closed-form way. Once the factor matrices are estimated, the second stage of the proposed receivers is to solve U independent CS problems that yield to channel parameter estimation of each MS, as will be shown later.

4.2. First Stage: Bilinear Alternating Least Squares (B-ALS Receiver). According to (8)–(10), we can obtain the following representations for the unfolding matrices $[\mathcal{X}]_{(1)} \in \mathbb{C}^{Q\times TK}$, $[\mathcal{X}]_{(2)} \in \mathbb{C}^{T\times QK}$, and $[\mathcal{X}]_{(3)} \in \mathbb{C}^{K\times QT}$ of $\mathcal{X}$ in terms of its factor matrices:

$$[\mathcal{X}]_{(1)} = \mathbf{\Phi}\left(\mathbf{C} \diamond \mathbf{Y}_{\text{e}}\right)^{\text{T}}, \tag{24}$$

$$[\mathcal{X}]_{(2)} = \mathbf{Y}_{\text{e}}\left(\mathbf{C} \diamond \mathbf{\Phi}\right)^{\text{T}}, \tag{25}$$

$$[\mathcal{X}]_{(3)} = \mathbf{C}\left(\mathbf{Y}_{\text{e}} \diamond \mathbf{\Phi}\right)^{\text{T}}, \tag{26}$$

where we denote $\mathbf{\Phi} = \mathbf{F}^{\text{H}}\mathbf{G}_{\text{e}}$ for the simplicity of representation.

Since the multifrequency coding matrix $\mathbf{C}$ is assumed to be known at the BS while the DL and UL channel-state information are not available, the proposed B-ALS receiver consists of estimating $\mathbf{\Phi}$ and $\mathbf{Y}_{\text{e}}$ in an alternating way from $[\mathcal{X}]_{(1)}$ and $[\mathcal{X}]_{(2)}$ by optimizing, respectively, the following two nonlinear LS problems:

$$\begin{aligned}\widehat{\mathbf{\Phi}} &= \underset{\mathbf{\Phi}}{\operatorname{argmin}} \left\| [\mathcal{X}]_{(1)} - \mathbf{\Phi}\left(\mathbf{C} \diamond \mathbf{Y}_{\text{e}}\right)^{\text{T}} \right\|_{\text{F}}^{2}, \\ \widehat{\mathbf{Y}}_{\text{e}} &= \underset{\mathbf{Y}_{\text{e}}}{\operatorname{argmin}} \left\| [\mathcal{X}]_{(2)} - \mathbf{Y}_{\text{e}}\left(\mathbf{C} \diamond \mathbf{\Phi}\right)^{\text{T}} \right\|_{\text{F}}^{2}.\end{aligned} \tag{27}$$

The solutions of which are given by $\widehat{\mathbf{\Phi}} = [\mathcal{X}]_{(1)} [(\mathbf{C} \diamond \widehat{\mathbf{Y}}_{\text{e}})^{\text{T}}]^{\dagger}$ and $\widehat{\mathbf{Y}}_{\text{e}} = [\mathcal{X}]_{(2)} [(\mathbf{C} \diamond \widehat{\mathbf{\Phi}})^{\text{T}}]^{\dagger}$, respectively.

Each iteration of the bilinear ALS-PARAFAC algorithm contains only two LS updating steps. At each step, one factor matrix is updated, while the other is assumed fixed to its value obtained in the previous step [31]. This procedure is repeated until the convergence of the algorithm, denoted by

$$\varepsilon^{(i)} = \left\| [\mathcal{X}]_{(1)} - \widehat{\mathbf{\Phi}}\left(\mathbf{C} \diamond \widehat{\mathbf{Y}}_{\text{e}}\right)^{\text{T}} \right\|_{\text{F}}^{2}, \tag{28}$$

the residual error between the received signal tensor and the reconstructed signal tensor at the i-th iteration. We declare that the first stage has converged at the i-th iteration when

$$\left|\varepsilon^{(i)} - \varepsilon^{(i-1)}\right| \leq \sigma, \tag{29}$$

where σ is a threshold. In our computational simulations, we set $\sigma = 10^{-6}$. Convergence to the global minimum is always achieved within a few iterations due to the knowledge of the frequency spread matrix at the BS. The proposed B-ALS receiver is summarized in Algorithm 1.

4.3. Alternative Closed-Form Solution to the First Stage. In contrast to the B-ALS receiver, the second proposed receiver named LS-KRF is an alternative closed-form solution that can be employed in particular cases in which $K \geq UN_{\text{MS}}$. The idea is to filter the received signal tensor $\mathcal{X}$ by exploiting the knowledge of the multifrequency coding matrix $\mathbf{C}$ and then solve a set of rank 1 approximation problems.

Initially, by multiplying both sides of $[\mathcal{X}]_{(3)}^{\text{T}}$ in (26) by the pseudoinverse of $\mathbf{C}^{\text{T}}$ from the right-hand side, we obtain

$$\begin{aligned}\mathbf{Y}_{\text{e}} \diamond \boldsymbol{\Phi} &= [\mathcal{X}]_{(3)}^{\text{T}}\left(\mathbf{C}^{\text{T}}\right)^{\dagger} \\ &= \left[\mathbf{y}_1 \otimes \boldsymbol{\varphi}_1, \ldots, \mathbf{y}_{UN_{\text{MS}}} \otimes \boldsymbol{\varphi}_{UN_{\text{MS}}}\right] \in \mathbb{C}^{QT \times UN_{\text{MS}}}.\end{aligned} \tag{30}$$

According to property in (4), the i-th column of (30) $(i = 1, \ldots, UN_{\text{MS}})$ can be rewritten as

$$\mathbf{y}_i \otimes \boldsymbol{\varphi}_i = \text{vec}\left(\boldsymbol{\varphi}_i \circ \mathbf{y}_i\right), \tag{31}$$

which denotes the vectorization operation of the rank 1 matrix $\boldsymbol{\Psi}_i = \boldsymbol{\varphi}_i \circ \mathbf{y}_i \in \mathbb{C}^{Q \times T}$. By defining $\mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^{\text{H}}$ as the singular value decomposition (SVD) of $\boldsymbol{\Psi}_i$, estimates for $\boldsymbol{\varphi}_i \in \mathbb{C}^{Q}$ and $\mathbf{y}_i \in \mathbb{C}^{T}$ $(i = 1, \ldots, UN_{\text{MS}})$ can be obtained by truncating the SVD of $\boldsymbol{\Psi}_i$ to a rank 1 approximation, i.e., [27]

$$\begin{aligned}\widehat{\boldsymbol{\varphi}}_i &= \sqrt{\sigma_1}\mathbf{u}_1, \\ \widehat{\mathbf{y}}_i &= \sqrt{\sigma_1}\mathbf{v}_1^*,\end{aligned} \tag{32}$$

where $\mathbf{u}_1 \in \mathbb{C}^{Q}$ and $\mathbf{v}_1 \in \mathbb{C}^{T}$ are the corresponding first left and right singular vectors of $\mathbf{U}_i$ and $\mathbf{V}_i$, respectively. σ_1 denotes the largest singular value of the matrix $\boldsymbol{\Sigma}_i$. Final estimates for $\widehat{\boldsymbol{\Phi}}$ and $\widehat{\mathbf{Y}}_{\text{e}}$ are obtained by repeating this SVD computation $i = 1, \ldots, UN_{\text{MS}}$ times in parallel, one for each column of (30). The pseudocode of the LS-KRF receiver is summarized in Algorithm 2.

4.4. Second Stage: Sparse Formulation to DL and UL Channel Parameters Estimation. Once the matrices $\widehat{\boldsymbol{\Phi}}$ and $\widehat{\mathbf{Y}}_{\text{e}}$ are estimated, the second stage of the proposed receivers consists in estimating the channel parameters (AoDs, AoAs, and path gains) to reconstruct the channel matrices $\widehat{\mathbf{H}}_u$ and $\widehat{\mathbf{G}}_u$ related to each MS. Thanks to the knowledge of the multifrequency coding matrix $\mathbf{C}$, the estimated factor matrices are not affected by permutation of columns ambiguity. Therefore, the first stage provides automatic separation of the compressed users' channels. From $\widehat{\boldsymbol{\Phi}}$ and $\widehat{\mathbf{Y}}_{\text{e}}$, the multiuser channel estimation problem can be decoupled into U single-user ones as formulated below.

Let us rewrite the block representation of $\widehat{\boldsymbol{\Phi}}$, defined as

$$\widehat{\boldsymbol{\Phi}} = \left[\widehat{\boldsymbol{\Phi}}_1, \widehat{\boldsymbol{\Phi}}_2, \ldots, \widehat{\boldsymbol{\Phi}}_U\right] \in \mathbb{C}^{Q \times UN_{\text{MS}}}, \tag{33}$$

where

$$\widehat{\boldsymbol{\Phi}}_u = \mathbf{F}^{\text{H}}\widehat{\mathbf{G}}_u \in \mathbb{C}^{Q \times N_{\text{MS}}}, \quad u = 1, \ldots, U. \tag{34}$$

where $\widehat{\mathbf{G}}_u \in \mathbb{C}^{N_{\text{BS}} \times N_{\text{MS}}}$ denotes the estimated UL channel matrix related to the u-th MS. By replacing $\widehat{\mathbf{G}}_u$ for (19), and then vectorizing (34) according to the property in (3), we have

$$\widehat{\boldsymbol{\varphi}}_u = \text{vec}\left(\widehat{\boldsymbol{\Phi}}_u\right) = \left(\widehat{\mathbf{A}}_{\text{MS}} \diamond \mathbf{F}^{\text{H}}\widehat{\mathbf{A}}_{\text{BS}}\right)\widehat{\boldsymbol{\beta}}. \tag{35}$$

Using the property in (2), we straightforwardly obtain

$$\widehat{\boldsymbol{\varphi}}_u = \left(\mathbf{I}_{N_{\text{MS}}} \otimes \mathbf{F}^{\text{H}}\right)\left(\widehat{\mathbf{A}}_{\text{MS}} \diamond \widehat{\mathbf{A}}_{\text{BS}}\right)\widehat{\boldsymbol{\beta}} \in \mathbb{C}^{QN_{\text{MS}}}, \tag{36}$$

where $\mathbf{I}_{N_{\text{MS}}}$ denotes an identity matrix of size $N_{\text{MS}} \times N_{\text{MS}}$.

The same procedure can directly be applied in the u-th block $\widehat{\mathbf{Y}}_u^{\text{T}}$ of the estimated factor matrix:

$$\widehat{\mathbf{Y}}_{\text{e}} = \left[\widehat{\mathbf{Y}}_1^{\text{T}}, \widehat{\mathbf{Y}}_2^{\text{T}}, \ldots, \widehat{\mathbf{Y}}_U^{\text{T}}\right] \in \mathbb{C}^{T \times UN_{\text{MS}}}, \tag{37}$$

where

$$\widehat{\mathbf{Y}}_u^{\text{T}} = \mathbf{S}^{\text{T}}\mathbf{W}^{\text{T}}\widehat{\mathbf{H}}_u^{\text{T}} + \mathbf{V}_u^{(\text{DL})\text{T}} \in \mathbb{C}^{T \times N_{\text{MS}}}. \tag{38}$$

$\widehat{\mathbf{H}}_u \in \mathbb{C}^{N_{\text{MS}} \times N_{\text{BS}}}$ denotes the estimated DL channel matrix defined above in (17). Similar to (36), we obtain the following vector formulation:

$$\widehat{\mathbf{y}}_u = \left(\mathbf{I}_{N_{\text{MS}}} \otimes \mathbf{S}^{\text{T}}\mathbf{W}^{\text{T}}\right)\left(\widehat{\mathbf{A}}_{\text{MS}} \diamond \widehat{\mathbf{A}}_{\text{BS}}\right)\widehat{\alpha} + \mathbf{v}_u^{(\text{DL})} \in \mathbb{C}^{TN_{\text{MS}}}, \tag{39}$$

where $\widehat{\mathbf{y}}_u = \text{vec}(\widehat{\mathbf{Y}}_u^{\text{T}})$ and $\mathbf{v}_u^{(\text{DL})} = \text{vec}(\mathbf{V}_u^{(\text{DL})\text{T}})$.

From (36) and (39), two independent CS problems can be formulated to jointly estimate the parameters of the DL and UL channels of the u-th MS. We assume that grid quantization errors are neglected, i.e., the AoDs and AoAs are drawn from a uniform angle grid of N points contained in the set $\{0, 2\pi/N, \ldots, 2\pi(N-1)/N\}$, with $N \gg L_u$ and $N \gg M_u$. Based on this assumption, we can obtain the following sparse formulations for the vectors $\widehat{\mathbf{y}}_u$ and $\widehat{\boldsymbol{\varphi}}_u$, respectively:

$$\widehat{\mathbf{y}}_u = \left(\mathbf{I}_{N_{\text{MS}}} \otimes \mathbf{S}^{\text{T}}\mathbf{W}^{\text{T}}\right)\boldsymbol{\Sigma}_{\text{D}}\overline{\alpha}, \tag{40}$$

$$\widehat{\boldsymbol{\varphi}}_u = \left(\mathbf{I}_{N_{\text{MS}}} \otimes \mathbf{F}^{\text{H}}\right)\boldsymbol{\Sigma}_{\text{D}}\overline{\boldsymbol{\beta}}, \tag{41}$$

where $\boldsymbol{\Sigma}_{\text{D}} \in \mathbb{C}^{N_{\text{MS}}N_{\text{BS}} \times N^2}$ denotes the known dictionary matrix used to solve the sparse signal recovery problem, defined as

$$\boldsymbol{\Sigma}_{\text{D}} = \overline{\mathbf{A}}_{\text{MS}} \otimes \overline{\mathbf{A}}_{\text{BS}} \in \mathbb{C}^{N_{\text{MS}}N_{\text{BS}} \times N^2}. \tag{42}$$

The matrices $\overline{\mathbf{A}}_{\text{MS}} \in \mathbb{C}^{N_{\text{MS}} \times N}$ and $\overline{\mathbf{A}}_{\text{BS}} \in \mathbb{C}^{N_{\text{BS}} \times N}$ that make the dictionary are given by

$$\begin{aligned}\overline{\mathbf{A}}_{\text{MS}} &= \left[\mathbf{a}_{\text{MS}}(0), \mathbf{a}_{\text{MS}}\left(\frac{2\pi}{N}\right), \ldots, \mathbf{a}_{\text{MS}}\left(\frac{2\pi(N-1)}{N}\right)\right], \\ \overline{\mathbf{A}}_{\text{BS}} &= \left[\mathbf{a}_{\text{BS}}(0), \mathbf{a}_{\text{BS}}\left(\frac{2\pi}{N}\right), \ldots, \mathbf{a}_{\text{BS}}\left(\frac{2\pi(N-1)}{N}\right)\right],\end{aligned} \tag{43}$$

```
(1) Set $i = 0$;
    Initialize randomly the factor matrix $\widehat{\mathbf{Y}}_{\mathrm{e}(i=0)}$;
(2) $i \longleftarrow i + 1$;
(3) From $[\mathcal{X}]_{(1)}$, obtain a LS estimate of $\widehat{\mathbf{\Phi}}_{(i)}$:
        $\widehat{\mathbf{\Phi}}_{(i)} = [\mathcal{X}]_{(1)} [(\mathbf{C} \diamond \widehat{\mathbf{Y}}_{\mathrm{e}(i-1)})^{\mathrm{T}}]^{\dagger}$;
(4) From $[\mathcal{X}]_{(2)}$, obtain a LS estimate of $\widehat{\mathbf{Y}}_{\mathrm{e}(i)}$:
        $\widehat{\mathbf{Y}}_{\mathrm{e}(i)} = [\mathcal{X}]_{(2)} [(\mathbf{C} \diamond \widehat{\mathbf{\Phi}}_{(i)})^{\mathrm{T}}]^{\dagger}$;
(5) Repeat steps 2–4 until convergence. The convergence is achieved when $|\varepsilon^{(i)} - \varepsilon^{(i-1)}| \leq 10^{-6}$, where $\varepsilon^{(i)}$ is the residual error
    computed in the $i$-th iteration.
```

Algorithm 1: Pseudocode of the B-ALS receiver.

```
(1) Apply the unvec$_{Q\times T}$ operator in the $i$-th column of (30) to obtain the rank 1 matrix $\mathbf{\Psi}_i \in \mathbb{C}^{Q\times T}$;
(2) Compute the SVD $\mathbf{\Psi}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^{\mathrm{H}}$. Then, obtain the estimates for the $i$-th columns of $\widehat{\mathbf{\Phi}}$ and $\widehat{\mathbf{Y}}_{\mathrm{e}}$ as:
        $\widehat{\boldsymbol{\varphi}}_i = \sqrt{\sigma_1}\mathbf{u}_1$ and $\widehat{\mathbf{y}}_i = \sqrt{\sigma_1}\mathbf{v}_1^{*}$,
        where $\mathbf{u}_1 \in \mathbb{C}^{Q}$ and $\mathbf{v}_1 \in \mathbb{C}^{T}$ are the first left and right singular vectors of $\mathbf{U}_i$ and $\mathbf{V}_i$, respectively, and $\sigma_1$ is the largest singular value;
(3) Repeat steps 1-2 for all columns of (30).
```

Algorithm 2: Pseudocode of the LS-KRF receiver.

and contain all points of the uniform angle grid. The left-hand side matrices $(\mathbf{I}_{N_{\mathrm{MS}}} \otimes \mathbf{S}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}})$ and $(\mathbf{I}_{N_{\mathrm{MS}}} \otimes \mathbf{F}^{\mathrm{H}})$ are called measurement matrices of the sparse problems. $\overline{\boldsymbol{\alpha}} \in \mathbb{C}^{N^2}$ and $\overline{\boldsymbol{\beta}} \in \mathbb{C}^{N^2}$ are sparse vectors obtained by augmenting the vector gains $\boldsymbol{\alpha}$ and β with zero elements, respectively.

Estimates for the parameters of the channel matrices $\mathbf{H}_u$ and $\mathbf{G}_u$ can be obtained by applying CS algorithms in the estimated sparse vectors (40) and (41). Many efficient algorithms such as orthogonal matching pursuit (OMP) [22], structured compressive sampling matching pursuit (S-CoSaMP) [25], and fast iterative shrinkage-thresholding (FISTA) [23], to name a few, can be used to solve these two sparse signal recovery problems. In a simplified view, the estimates for the path gains $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\beta}$ correspond to nonzero entries of the estimated sparse vectors $\overline{\boldsymbol{\alpha}}$ and $\overline{\beta}$, while estimation for the spatial parameters (AoDs and AoAs) are obtained by selecting the columns of the dictionary matrix $\mathbf{\Sigma}_{\mathrm{D}}$ related to the positions of the estimated path gains in the sparse vector. In our numerical results, we adopt the OMP algorithm to estimate the UL and DL channel parameters for simplicity reasons, although any state-of-the-art CS-based algorithm is equally applicable to solve problems (40) and (41).

Remark: Compared to (36), the sparse signal recovery problem formulated from (39) naturally incorporates the DL noise contribution in its structure. For this reason, the proposed closed-loop framework for channel estimation can lead to some performance degradation in the DL channel estimation compared to UL channel estimation. Therefore, we can observe a trade-off between DL channel estimation accuracy performed by the BS and reduction of the processing cost for channel estimation at the MS side. This discussion is reinforced by means of numerical simulations in Section 6.

4.5. Joint DL and UL Channel Estimation. Finally, from the estimated channel parameters (AoDs, AoAs, and path gains), the BS can construct the estimated DL and UL channel matrices $\widehat{\mathbf{H}}_u$ and $\widehat{\mathbf{G}}_u$ of the u-th MS according to relations (17) and (19) as follows:

$$\widehat{\mathbf{H}}_u = \widehat{\mathbf{A}}_{\mathrm{MS}} \operatorname{diag}(\widehat{\boldsymbol{\alpha}}) \widehat{\mathbf{A}}_{\mathrm{BS}}^{\mathrm{T}}, \tag{44}$$

$$\widehat{\mathbf{G}}_u = \widehat{\mathbf{A}}_{\mathrm{BS}} \operatorname{diag}(\widehat{\boldsymbol{\beta}}) \widehat{\mathbf{A}}_{\mathrm{MS}}^{\mathrm{T}}. \tag{45}$$

As previously presented in Section 4.4, the proposed receivers decouple the multiuser channel estimation problem into $2U$ single-user ones (U problems dedicated to each communication link) that can be solved independently for each MS. Since $2U$ digital processing units are available at the BS, the second stage of the proposed receivers can be computed in parallel. Therefore, its processing delay can be kept constant (i.e., it does not increase with the number of MSs), when the BS is equipped with multiple (at least $2U$) digital processing units. The parallelized processing for the second stage of the proposed receivers is illustrated in Figures 3 and 4. In addition, the overall pseudocode of the proposed two-stage tensor-based receivers for joint DL and UL channel estimation is summarized in Algorithm 3.

5. Identifiability Issues

In this section, we examine the identifiability issues under which the compressed DL and UL channel matrices $\mathbf{\Phi}$ and $\mathbf{Y}_{\mathrm{e}}$ can be jointly and uniquely recovered using the proposed receivers.

5.1. B-ALS Receiver. Unique LS solutions for the compressed DL and UL channel matrices $\mathbf{\Phi}$ and $\mathbf{Y}_{\mathrm{e}}$ obtained from (24) and (25) require that $(\mathbf{C} \diamond \mathbf{Y}_{\mathrm{e}})^{\mathrm{T}} \in \mathbb{C}^{UN_{\mathrm{MS}}\times KT}$ and

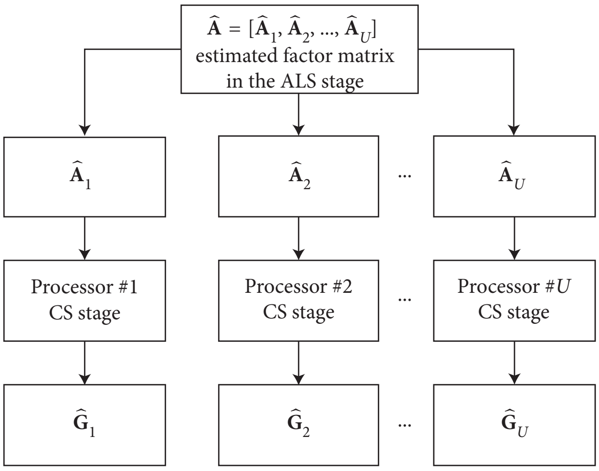


Figure 3: Illustration of the parallelized processing for the estimation of U uplink channels. The u-th block $\hat{\mathbf{A}}_u$ of the estimated factor matrix $\hat{\mathbf{A}}$ is forwarded and processed for a dedicated processor.

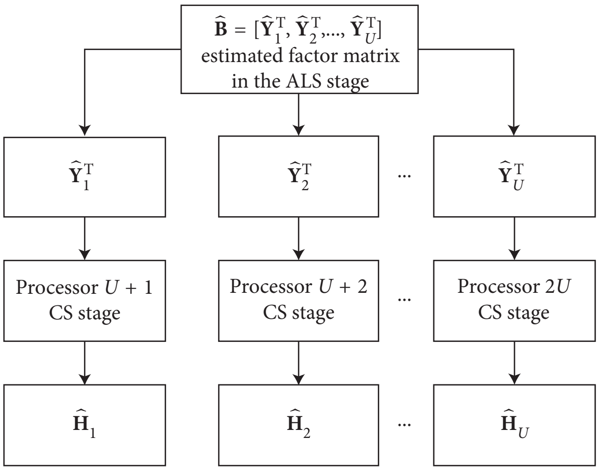

$\hat{\mathbf{B}} = [\hat{\mathbf{Y}}_1^{\mathrm{T}}, \hat{\mathbf{Y}}_2^{\mathrm{T}}, ..., \hat{\mathbf{Y}}_U^{\mathrm{T}}]$
estimated factor matrix
in the ALS stage
$\hat{\mathbf{Y}}_1^{\mathrm{T}}$
$\hat{\mathbf{Y}}_2^{\mathrm{T}}$
...
$\hat{\mathbf{Y}}_U^{\mathrm{T}}$
Processor $U + 1$
CS stage
Processor $U + 2$
CS stage
...
Processor $2U$
CS stage
$\hat{\mathbf{H}}_1$
$\hat{\mathbf{H}}_2$
...
$\hat{\mathbf{H}}_U$

Figure 4: Illustration of the parallelized processing for the estimation of the U downlink channels. The u-th block $\hat{\mathbf{Y}}_u^{\mathrm{T}}$ of the estimated factor matrix $\hat{\mathbf{B}}$ is forwarded and processed for a dedicated processor.

```
1 First Stage. estimation of the compressed channel matrices
(1.1) From the received signal tensor $\mathcal{X}$ in (20), obtain the estimated factor matrices $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{Y}}_e$ via B-ALS or LS-KRF described in
      Algorithms 1 and 2, respectively,
(2) Second Stage. parameters estimation and channels reconstruction
(2.1) From $\hat{\mathbf{\Phi}} = [\hat{\mathbf{\Phi}}_1, \hat{\mathbf{\Phi}}_2, \ldots, \hat{\mathbf{\Phi}}_U]$ and $\hat{\mathbf{Y}}_e = [\hat{\mathbf{Y}}_1^{\mathrm{T}}, \hat{\mathbf{Y}}_2^{\mathrm{T}}, \ldots, \hat{\mathbf{Y}}_U^{\mathrm{T}}]$, obtain the estimates of the channel parameters (AoDs, AoAs, and path
      gains) for each MS by applying a CS recovery algorithm (e.g., OMP) to problems (40) and (41) independently
(2.2) The BS constructs the estimated DL and UL channel matrices $\hat{\mathbf{H}}_u$ and $\hat{\mathbf{G}}_u$ according to (44) and (45), respectively.
```

Algorithm 3: Overall pseudocode of the proposed receivers for joint DL and UL channel estimation.

$(\mathbf{C} \diamond \mathbf{\Phi})^{\mathrm{T}} \in \mathbb{C}^{UN_{\mathrm{MS}} \times KQ}$ have full row rank to be right invertible. Hence, the following two conditions must be satisfied:

$$\begin{aligned} KT &\geq UN_{\mathrm{MS}}, \\ KQ &\geq UN_{\mathrm{MS}}. \end{aligned} \tag{46}$$

Combining these conditions yields the following lower bound on the number of subcarriers required for the multifrequency coding at the MS:

$$K \geq \max\left(\left\lceil \frac{UN_{\mathrm{MS}}}{T}, \frac{UN_{\mathrm{MS}}}{Q} \right\rceil\right), \tag{47}$$

where $\lceil x \rceil$ denotes the smallest integer number that is greater or equal to x.

5.2. LS-KRF Receiver. The LS-KRF receiver requires that the following necessary and sufficient uniqueness condition be satisfied:

$$K \geq U N_{\mathrm{MS}}. \tag{48}$$

Note that this condition indicates that the application of the LS-KRF receiver requires a more restricted scenario compared to the proposed B-ALS receiver since the number of frequency resources (subcarriers) increases with the number of antennas at the MSs and active MSs. On the contrary, the LS-KRF receiver is a closed-form solution in contrast to the iterative B-ALS receiver.

6. Simulation Results

In this section, we present a set of simulation results to evaluate the performance of the proposed joint DL and UL channel estimator. We compare the proposed channel training framework with the conventional training framework illustrated in Figure 1, where the CS-based OMP algorithm [22] is applied at both MSs and BS to estimate the channel parameters in a decoupled way. The OMP algorithm is also considered as the second stage of our algorithm, according Section 4.4. The MSs and the BS employ uniform linear arrays with half-wavelength-spaced antennas. We set $N_{\mathrm{BS}} = 32$, $N_{\mathrm{MS}} = 16$, $U = 2$, $N = 64$, and equal signal-to-noise ratio (SNR) for the DL and UL communications in all experiments. The obtained results are averaged over 1000 independent Monte Carlo runs. At each run, the DL and UL channel matrices with $L_u = 3$ and $M_u = 3$ paths per user and HB matrices are generated in accordance with equations (17), (19), and (12), respectively. The pilot signal $\mathbf{S}$ is a binary phase shift keying (BPSK) modulated matrix, and the multifrequency coding matrix $\mathbf{C}$ has random coefficients following a uniform distribution.

The receiver's performance is evaluated in terms of the normalized mean square error (NMSE) measures between the estimated and true DL and UL channel matrices:

$$\begin{aligned} \mathrm{NMSE}(\widehat{\mathbf{H}}) &= \frac{\sum_{u=1}^{U} \left\| \mathbf{H}_u - \widehat{\mathbf{H}}_u \right\|_{\mathrm{F}}^2}{\sum_{u=1}^{U} \left\| \mathbf{H}_u \right\|_{\mathrm{F}}^2}, \\ \mathrm{NMSE}(\widehat{\mathbf{G}}) &= \frac{\sum_{u=1}^{U} \left\| \mathbf{G}_u - \widehat{\mathbf{G}}_u \right\|_{\mathrm{F}}^2}{\sum_{u=1}^{U} \left\| \mathbf{G}_u \right\|_{\mathrm{F}}^2}. \end{aligned} \tag{49}$$

In our experiments, we evaluate the accuracy of channel estimation in terms of the NMSE metric for different values of signal-to-noise ratio (SNR), number of transmission (P) and reception (Q) beams, number of training subcarriers (K), and length of the pilot sequences (T).

Figures 5 and 6 show the NMSE as a function of the number of transmission (P) and reception (Q) beams for different values of SNR, and fixed values $U = 2$, $T = 16$, and $K = 25$. According Figure 6, the proposed method outperforms the classical framework to the UL channel estimation, while the DL performance is worse in all the simulated SNR ranges as shown in Figure 5. From this experiment, we can observe the trade-off between DL channel estimation accuracy and computational complexity. In other words, the proposed framework concentrates most of the processing burden for channel estimation at the BS side, while a better performance of the DL channel estimation comes at the expense of a high computational cost to complex channel estimation processing at the MS side when the conventional framework is

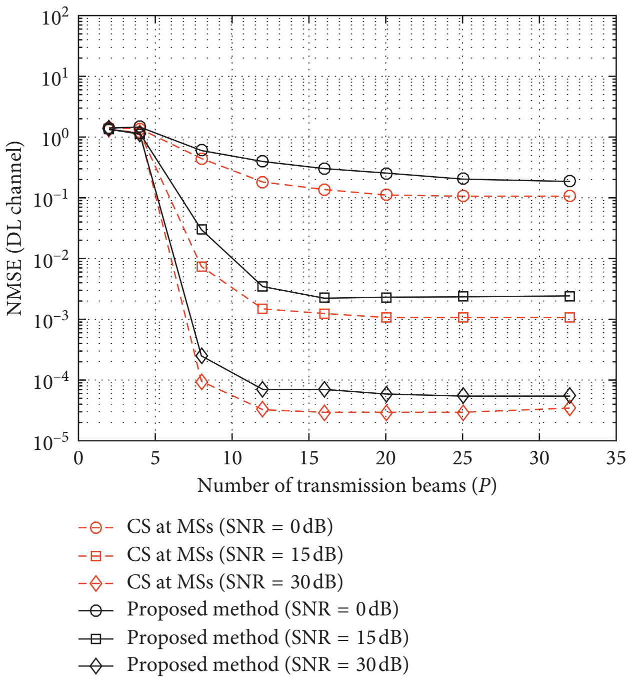


Figure 5: NMSE vs. number of transmission beams P for the DL channel estimation: $U = 2$, $T = 16$, and $K = 25$.

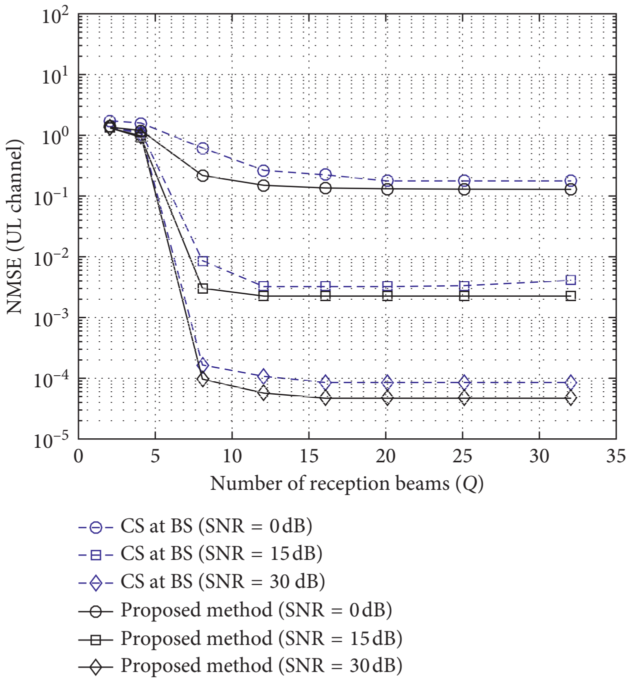


Figure 6: NMSE vs. number of reception beams Q for the UL channel estimation: $U = 2$, $T = 16$, and $K = 25$.

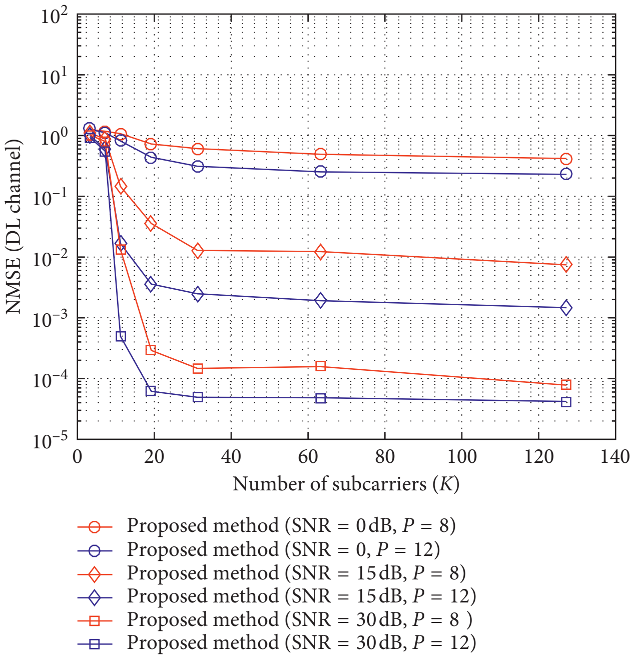


Figure 7: NMSE vs. number of subcarriers K for the DL channel estimation: $U = 2$ and $T = 16$.

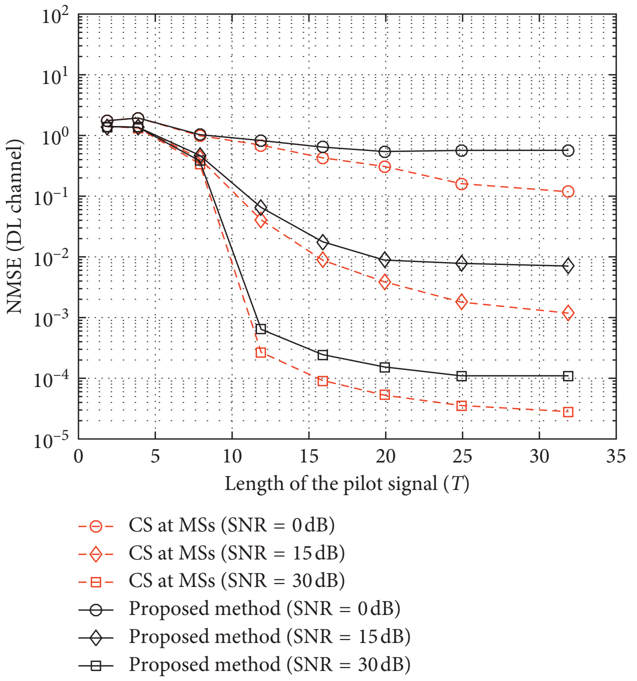


Figure 9: NMSE vs. length of the training sequence T for the DL channel estimation: $U = 2$, $P = 8$, and $K = 25$.

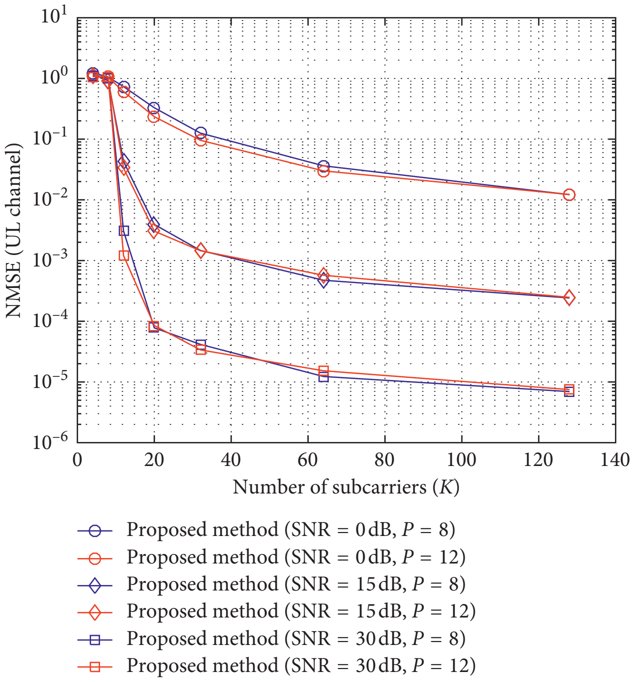


Figure 8: NMSE vs. number of subcarriers K for the UL channel estimation: $U = 2$ and $T = 16$.

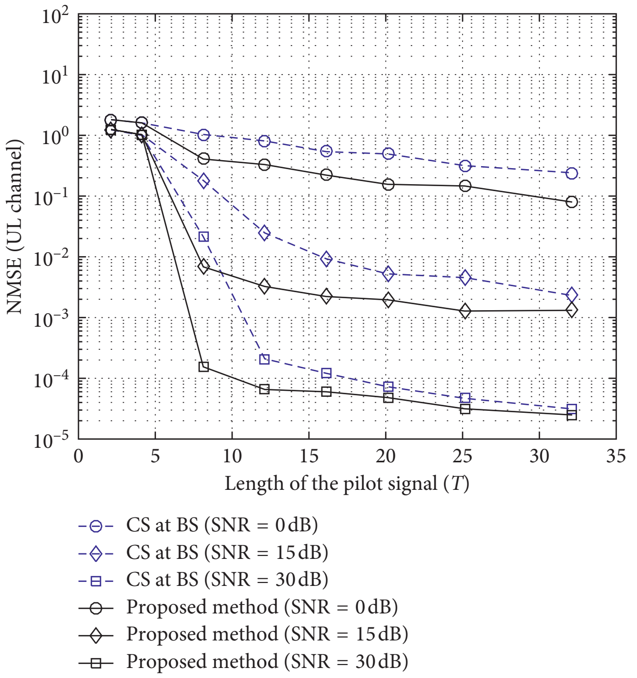


Figure 10: NMSE vs. length of the training sequence T for the UL channel estimation: $U = 2$, $P = 8$, and $K = 25$.

utilized. On the contrary, the performance loss at DL is compensated with more accurate estimations at UL. In addition, the NMSE performance is not influenced by the number of RF chains when P and Q are greater than 12. This result reveals that the proposed framework provides a good channel estimation accuracy even when the BS is equipped with a few number of RF chains, which is the case in HB architectures.

In Figures 7 and 8, the NMSE performance is evaluated as a function of the number of subcarriers (K). An increase of K leads to an improved performance only until $K = 32$ subcarriers in the DL channel estimation, while for the UL channel estimation this value is approximately equal to $K = 64$ subcarriers. This result shows that the proposed closed-loop channel training framework can operate with few frequency resources to jointly estimate the MS channels with high accuracy.

Figures 9 and 10 depict the NMSE performance in terms of the length of the pilot sequence (T). In this experiment, we also set $U = 2$, $P = 8$, and $K = 25$. Here, we conclude that short pilot sequences are necessary to estimate the DL and UL channels from the proposed method. Low variability in the NMSE values is observed when $T > 15$. For a massive MIMO scenario, this result implies in a substantial reduction in the pilot overhead to joint DL and UL channel estimation.

7. Conclusion and Perspectives

In this paper, we have addressed the joint DL and UL channel estimation problem for multiuser FDD massive MIMO systems with HB architecture. As contributions of this work, we firstly proposed a novel closed-loop and multifrequency-based channel training framework that concentrates most of the processing burden for channel estimation at the BS side. We have shown that making use of the proposed framework, the received closed-loop signal follows a third-order PARAFAC model, which can be exploited by two tensor-based semiblind receivers followed by compressed sensing recovery of the channel parameters. Additionally, we have also provided an identifiability study. We have compared our proposed approach with the conventional channel training framework, where the DL and UL channel estimation problems are treated as two decoupled problems, i.e., solved by the MSs and BS, separately. Compared to the conventional framework, the proposed receivers have shown a superior performance in the estimation of the UL channel, while the performance of the DL channel estimation exhibits some degradation. It is worth noting that such a degradation is the price to pay for the complexity reduction at the MS by transferring the processing burden associated with the DL channel estimation to the BS. Perspectives include the extension of the proposed modeling to frequency- and time-selective channels.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Ericsson Research, Technical Cooperation Contract UFC.47. The authors also thank the partial support of CAPES/PROBRAL (grant no. 88887.144009/2017-00), CNPq and FUNCAP.

References

[1] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic, and R. W. Heath, “MIMO precoding and combining solutions for millimeter-wave systems,” *IEEE Communications Magazine*, vol. 52, no. 12, pp. 122–131, 2014.

[2] V. Venkateswaran and A.-J. van der Veen, “Partial beamforming to reduce ADC power consumption in antenna array systems,” in *Proceedings of the IEEE 9th Workshop on Signal Processing Advances in Wireless Communications*, pp. 146–150, Recife, Brazil, July 2008.

[3] A. Alkhateeb, Y.-H. Nam, J. Zhang, and R. W. Heath, “Massive MIMO combining with switches,” *IEEE Wireless Communications Letters*, vol. 5, no. 3, pp. 232–235, 2016.

[4] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, “An overview of signal processing techniques for millimeter wave MIMO systems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, 2016.

[5] R. Mendez-Rial, C. Rusu, N. Gonzalez-Prelcic, A. Alkhateeb, and R. W. Heath, “Hybrid MIMO architecture for millimeter wave communications: phased shifters or switches?,” *IEEE Access*, vol. 4, pp. 247–267, 2016.

[6] A. Hajimiri, H. Hashemi, A. Natarajan, X. Guan, and A. Komijani, “Integrated phased array systems in silicon,” *Proceedings of the IEEE*, vol. 93, no. 9, pp. 1637–1655, 2005.

[7] G. Wang, H. Ding, W. Woods, and E. Mina, “Wideband on-chip RF MEMS switches in a BiCMOS technology for 60 GHz applications,” in *Proceedings of the International Conference on Microwave and Millimeter Wave Technology*, vol. 3, pp. 1389–1392, Nanjing, China, April 2008.

[8] J. Brady, N. Behdad, and A. M. Sayeed, “Beamspace MIMO for millimeter-wave communications: system Architecture, modeling, analysis, and measurements,” *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 7, pp. 3814–3827, 2013.

[9] L. Liang, W. Xu, and X. Dong, “Low-complexity hybrid precoding in massive multiuser MIMO systems,” *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 653–656, 2014.

[10] A. Alkhateeb, R. W. Heath, and G. Leus, “Achievable rates of multiuser millimeter wave sustems with hybrid precoding,” in *Proceedings of the IEEE International Conference on Communication Workshop*, pp. 1232–1237, London, UK, June 2015.

[11] R. A. S. Gallacher and M. S. Rahman, “Multi-user MIMO strategies for a millimeter wave communication system using hybrid beamforming,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 2437–2443, London, UK, June 2015.

[12] C. Huang, L. Liu, C. Yuen, and S. Sun, “A LSE and sparse message passing-based channel estimation for mmwave MIMO systems,” in *Proceedings of the IEEE Globecom Workshops*, pp. 1–6, Washington, DC, USA, December 2016.

[13] D. Zhu, J. Choi, and R. W. Heath, “Auxiliary beam pair enabled AoD and AoA estimation in closed-loop large-scale millimeter-wave MIMO systems,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4770–4785, 2017.

[14] H. Ghauch, T. Kim, M. Bengtsson, and M. Skoglund, “Subspace estimation and decomposition for large millimeter-wave MIMO systems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 528–542, 2016.

[15] X. Li, J. Fang, H. Li, and P. Wang, “Millimeter wave channel estimation via exploiting joint sparse and low-rank structures,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1123–1133, 2018.

[16] A. Liao, Z. Gao, Y. Wu, H. Wang, and M.-S. Alouini, “2D unitary ESPRIT based super-resolution channel estimation for millimeter-wave massive MIMO with hybrid precoding,” *IEEE Access*, vol. 5, pp. 24747–24757, 2017.

[17] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, 2014.

[18] K. Venugopal, A. Alkhateeb, R. W. Heath, and N. G. Prelcic, "Time-domain channel estimation for wideband millimeter wave systems with hybrid architecture," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6493–6497, New Orleans, LA, USA, March 2017.

[19] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and S. Li, "Channel estimation for millimeter-wave multiuser MIMO systems via PARAFAC decomposition," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7501–7516, 2016.

[20] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 7, pp. 1524–1538, 2017.

[21] D. C. Araújo and A. L. F. de Almeida, "Tensor-based compressed estimation of frequency-selective mmwave MIMO channels," in *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Guadeloupe, West Indies, March 2018.

[22] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 40–44, Pacific Grove, CA, USA, November 1993.

[23] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[24] R. A. Harshman, "Foundations of the PARAFAC procedure: models and conditions for an explanatory multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 10, pp. 1–84, 1970.

[25] W. Shen, L. Dai, Y. Shi, B. Shim, and Z. Wang, "Joint channel training and feedback for FDD massive MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8762–8767, 2016.

[26] R. Bro, *Multi-way analysis in the food industry: models, algorithms and applications*, Ph.D dissertation, University of Amsterdam, Amsterdam, Netherlands, 1998.

[27] F. Roemer and M. Haardt, "Tensor-based channel estimation and iterative refinements for two-way relaying with multiple antennas and spatial reuse," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5720–5735, 2010.

[28] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[29] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[30] 3GPP TS 38.211, *NR; Physical Channels and Modulation, Version 15.2.0 Release 15, Technical Specification, 3GPP*, Sophia-Antipolis, France, 2018.

[31] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis*, Wiley, Hoboken, NJ, USA, 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 8458016, 17 pages
https://doi.org/10.1155/2019/8458016

Research Article

Estimating Network Flow Length Distributions via Bayesian Nonnegative Tensor Factorization

Barış Kurt,[1] **Ali Taylan Cemgil**,[1] **Güneş Karabulut Kurt**,[2] **and Engin Zeydan**[3]

[1]*Department of Computer Engineering, Bogazici University, Istanbul, Turkey*
[2]*Istanbul Technical University, Istanbul, Turkey*
[3]*Centre Tecnològic de Telecomunicacions de Catalunya, Castelldefels, 08860, Spain*

Correspondence should be addressed to Barış Kurt; bariskurt@gmail.com

Received 28 December 2018; Revised 7 June 2019; Accepted 16 July 2019; Published 18 August 2019

Guest Editor: C. Alexandre R. Fernandes

In this paper, we develop a framework to estimate network flow length distributions in terms of the number of packets. We model the network flow length data as a three-way array with day-of-week, hour-of-day, and flow length as entities where we observe a count. In a high-speed network, only a sampled version of such an array can be observed and reconstructing the true flow statistics from fewer observations becomes a computational problem. We formulate the sampling process as matrix multiplication so that any sampling method can be used in our framework as long as its sampling probabilities are written in matrix form. We demonstrate our framework on a high-volume real-world data set collected from a mobile network provider with a random packet sampling and a flow-based packet sampling methods. We show that modeling the network data as a tensor improves estimations of the true flow length histogram in both sampling methods.

1. Introduction

Monitoring network statistics is crucial for the maintenance and infrastructure planning for network service providers. Statistical information about traffic patterns helps a service provider to characterize its network resource usage and user behavior, to infer future traffic demands, to detect traffic and usage anomalies, and to provide insights to improve the performance of the network [1]. However, network monitoring has become a difficult task due to increasingly high-volume and high-speed data over modern networks, and in most cases, it requires special hardware. For this reason, sampling [2] becomes a viable approach for extracting statistics from such high-speed networks. In this work, we are interested in one of the most important network statistics, the flow length distribution (FLD).

A network flow is defined as a set of Internet protocol (IP) packets with the same signature observed within a limited time period. The flow signature is composed of the IP and port pairs of both the source and destination nodes together with level-3 protocol types such as transport control protocol (TCP) or user datagram protocol (UDP). A flow starts with the arrival of the first packet and terminated when the interpacket timeout is exceeded. The total number of packets in a flow is referred to as the flow length and the length distribution of a set of flows that are terminated in a time window is called flow length distribution.

In this work, we are using one of the most popular methods for collecting per-flow information, i.e., passive measurement. In this method, network packets are processed as they pass through a passive measurement beacon connected to the network, e.g, router. The beacon keeps a look-up table for flow identification. The beacon processes a packet by searching its corresponding flow inside the look-up table using its signature. If such a flow is found, its statistics are updated. Otherwise, the packet is treated as the first packet of a new flow, and the new flow is inserted into the table. Once a flow is terminated, its statistics are transferred to a storage.

The flow length histogram can be calculated exactly by processing every packet that passes through the measurement beacon. In order to implement such a direct method, the monitoring beacon needs to maintain a table to hold information for all active flows on the network. However, substantial amount of concurrent flows with very short packet interarrival times of current high-speed networks (on the order of 10 Gbps to 100 Gbps inside carrier's network today) make this brute-force counting method very costly to implement. First of all, this method would require a large amount of memory to record the flow table. Secondly, in a high-speed link, the interarrival times between packets, which may be as small as 8 nanoseconds in an OC-768 link, may be smaller than the time required to process flow hash operations such as identifying the packet and updating the flow statistics.

The characteristics of the network traffic data inevitably lead to the development of alternative methods for measurement such as random sampling, where a fraction of the network traffic is randomly selected and processed. The simplest sampling method is the uniform packet sampling [3–6], used in commercial systems [7, 8]. In uniform sampling, each packet is selected with a predefined constant probability. This approach is easy to implement since it does not require the flow identification of each packet. However, recovering the true flow length distribution from the random packet sampled traffic is a challenging problem. The unbiased estimator of the original flow length n for sampling probability π is $\hat{n}(m) = m/\pi$, where m is observed flow length. The relative error of this estimator, calculated as $\sqrt{1/(\pi - 1)}$ [3], grows unboundedly for short flows as the sampling rate gets smaller. The high error on the small flow lengths comes from the fact that most of the samples are collected from longer flows.

Flow-based adaptive sampling methods [9–14] were proposed for more accurate flow length estimation. In these methods, each incoming packet is processed and then sampled with a probability that is a function of the current sampled length of the flow that the packet belongs to. Here, the main idea is to use a smaller memory by compressing the flow statistics counters in the router. However, these methods need to be implemented on specialized and expensive hardware due to the mandatory packet identification and look-up step.

Both packet-based and flow-based adaptive sampling methods rely on numerical methods to recover the true FLD. In this work, we propose a framework that can be used to recover the true FLD from the sampled observation obtained by any sampling method. This framework uses a variant of the nonnegative tensor factorization NTF model, which we call the thin nonnegative tensor factorization (ThinNTF), where the "thin" prefix emphasizes that the factorization is applied directly to the samples, or namely "thinned" data.

In our framework, the network traffic data is modeled as a 3-way array, containing the number of flow length observations, with dimensions interpreted as (1) flow length, (2) hour-of-day, and (3) day-of-week to capture the hourly and daily periodicity in the data. The nonnegative factorization of this tensor basically gives us estimates in the form of a nonparametric mixture model. Therefore, our model is an improvement of the nonparametric flow length models in [3, 6] by having the capability of modeling data with an arbitrary amount of mixture components and use the periodicity.

While the ordinary NTF model [15] factorizes an observation tensor, the ThinNTF directly factorizes its sampled version and recovers the original tensor from the estimated factors. We take a fully Bayesian approach here and provide a generative model for the TNTF and a variational Bayes algorithm for inference. The contributions in this paper can be listed as follows:

(i) We model one week of flow length observations as a 3-dimensional tensor and observe the periodic behavior.

(ii) We propose a novel tensor factorization scheme, ThinNTF, which is able to find the factors of a latent tensor from its sampled counterpart. By doing so, we also solve the reconstruction problem.

(iii) We apply ThinNTF to real-world data sampled with two different sampling methods: uniform random packet sampling and flow-based adaptive sampling.

The structure of the paper is as follows. In Section 2, we provide the related works on network sampling and tensor factorization. In Section 3, we describe our real-world data and how we visualize it as a tensor. Additionally, we describe the sampling methods that we used to sample the data. In Section 4, we describe our ThinNTF model and the variational Bayes algorithm for estimating the factors. In Section 5, we describe our real-world data collection architecture. In Section 6, we present our synthetic and real-world experiments and results. Finally, in Section 7, we draw our conclusions.

2. Related Work

Sampling methods have long been applied to network traffic monitoring. A survey on fundamental network sampling strategies is given in [2]. Uniform packet sampling is extensively studied by various authors. Duffield et al. [3] propose the first nonparametric model for flow length distribution and provide a maximum likelihood estimation to recover the flow lengths. Riberio et al. [4] show that using protocol specific information gives better flow length distribution estimates in TCP flows. Yang and Michailidis [6] adopt the maximum likelihood approach to estimate both flow length and flow volume (number of bytes in a flow) distributions. Additionally, they model the data with a nonparametric mixture model of two components, where the first component models small flows and the second models large ones.

Flow-based sampling methods are proposed as alternatives to the uniform packet sampling since packet sampling has theoretical limitations when recovering true flow statistics [5]. These methods process every incoming packet and apply sampling conditionally. Kumar et al. [13, 16] propose two different algorithms where the flow size counters are compressed statistically. They also propose a nonuniform packet sampling algorithm based on sketch counting [12]. Hu et al. [10, 14] propose another nonuniform packet sampling algorithm, called adaptive nonlinear sampling (ANLS), for estimating flow lengths per each flow and then adopt this method to flow volume [11]. In our experiments, we are going to use ANLS as an example of flow-based sampling methods since it is the current state-of-the-art nonuniform sampling method.

Nonnegative tensor factorization is the generalization of the nonnegative matrix factorization (NMF) [17] to multiway arrays. In NMF, a nonnegative matrix is approximated with a multiplication of two nonnegative matrices. Minimizing the Kullback–Leibler divergence between the initial matrix and multiplied factors is a popular formulation of this method and can be solved with fixed-point iterations [18] or full Bayesian methods [19]. NMF has been used in many applications such as spectral data analysis [20], face recognition [17], and document clustering [21].

Modeling the flow length distribution as a mixture of distributions is first proposed by [6]. However, according to our best knowledge, there is no previous work that models a large volume of flow length data as a tensor. This work fills a gap in the literature by introducing tensor factorization methodology to network monitoring.

3. Problem Description

We describe our problem as a tensor thinning problem, where the count entities of the original flow lengths are stored in a tensor. We formulate the sampling process as a matrix multiplication operated on this data tensor. In order to do that, each sampling model should be represented as a matrix that transforms the original data tensor to a sampled one. We provide matrices for two sampling models: uniform packet sampling and ANLS flow-based packet sampling.

3.1. Notation and Indexes. For a clear notation, the scalar values are denoted by lightface letters, such as the index variable j and its maximum value J. The vectors are represented by boldface lower case letters, such as vector $\mathbf{x}$. Boldface upper case letters represent matrices, such as $\mathbf{F}$, $\mathbf{H}$, and $\mathbf{D}$, and the tensors are represented with calligraphic upper case letters, i.e, $\boldsymbol{\mathcal{X}}$. The individual entries in matrices and tensors are written like scalars, i.e., $f_{i,r}$ and $x_{i,j,k}$. The index denotes all the entries in the given dimension. For example $s_{i,:}$ is the i^{th} row of the $\mathbf{S}$ matrix and $\mathcal{X}_{i,:,:}$ is the i^{th} slice of the tensor $\mathcal{X}$ in the first dimension.

The index parameters are also fixed for clarity. The list of indexes and their ranges and semantic descriptions are given in Table 1. For example, the i index always represents an original flow length, while ν presents a sampled flow length. The range of ν starts from 0, since all of the packets of a flow may be discarded during the sampling process yielding a zero-length sampled flow, which is never observed.

3.2. Data Tensor. The original flow length data is represented in an $I \times J \times K$ tensor $\mathcal{X}$, with individual elements $x_{i,j,k}$, regarded as the number of flows that has length i measured at the hour j of the day k. In this setup, I is the maximum flow length value, J is the hours of the day, and K is the days of the week. For our real-world data, collected continuously for 1 week, these values are $I = 2,000,000$, $J = 24$, and $K = 7$.

Table 1: Indexes in the model.

Index	Range	Description
I	$[1, I]$	Original flow lengths
N	$[0, I]$	Sampled flow lengths
J	$[1, J]$	Hours of day
K	$[1, K]$	Days
R	$[1, R]$	Components

Working with large maximum flow size is not feasible for two reasons. First, learning a mixture model where each flow component has 2 million parameters is not a good formulation for this problem. Secondly, 99.9% of flows in our data have less than 100 packets, which means the tensor $\boldsymbol{\mathcal{X}}$ will be very sparse for $i > 100$. The clamping process can be defined as

$$\overline{\mathcal{X}}_{i,j,k} = \begin{cases} \mathcal{X}_{i,j,k}, & \text{for } i < I_{\max}, \\ \sum\limits_{l \geq I_{\max}} \mathcal{X}_{l,j,k}, & \text{for } i = I_{\max}, \end{cases} \tag{1}$$

where $\overline{\mathbf{X}}$ is the clamped tensor. The clamping does not require any change in the model and inference equations that are given in Section 4. Therefore, for notational clarity, we only use $\mathcal{X}$ as the generic original data tensor.

Figure 1 shows the vertical slices of our unsampled real-world data tensor $\mathcal{X}$, collected at the backbone of a mobile operator during a one week period, from Monday to Sunday, and clamped at $I_{\max} = 50$. The intensity images are generated from the logarithm of the flow length counts. The daily and hourly patterns are easily recognizable in the original FLD data.

3.3. Sampling Methods. Independent of the sampling method, we can define an $I \times (I + 1)$ size $\mathbf{S}$ matrix, where I is the maximum flow length with entries $s_{i,\nu}$ interpreted as the probability of sampling ν packets from an original flow of length i. Naturally, $\mathbf{S}$ is a lower diagonal matrix of the form:

$$\mathbf{S} = \begin{bmatrix} s_{1,0} & s_{1,1} & 0 & 0 & \cdots & 0 \\ s_{2,0} & s_{2,1} & s_{2,2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ s_{I,0} & s_{I,1} & s_{I,2} & s_{I,3} & \cdots & s_{I,I} \end{bmatrix}, \tag{2}$$

where its l^{th} row defines a probability distribution for the sampled flow length of a flow of size l. Given a flow size distribution $\mathbf{x} \in \mathbb{Z}^{\geq I}$, where $\mathbb{Z}^{\geq}$ is the set of nonnegative integers, the expected sampled flow length distribution would be given by $\widehat{y} = \mathbf{S}^{T}\mathbf{x}$. It immediately follows that the sampled flow size distribution y has length $I + 1$, with y_0 being the proportion of sampled flows with none of their packets sampled. During the sampling process, this value will never be observed naturally since the flow identification is performed only on selected packets. In all experiments throughout this paper, the $\mathbf{y}$ vector (or $\boldsymbol{\mathcal{Y}}$ tensor, which will be described later on) will be element-wise multiplied with a binary mask vector

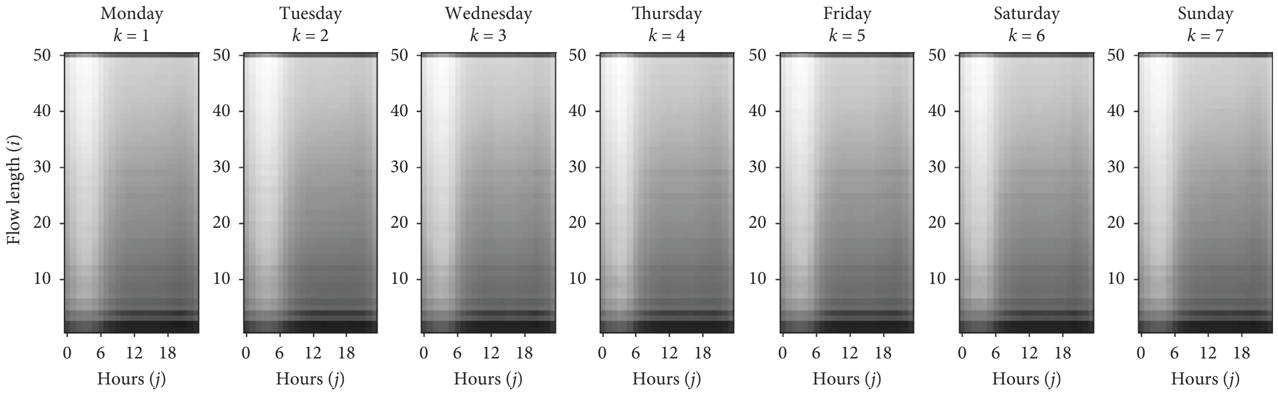


Figure 1: Slices of the original flow length tensor $\mathcal{X}$.

$\mathbf{m}$ (or binary mask tensor $\mathcal{M}$), whose entries are set to 1 except the ones corresponding to zero-sampled flow lengths, in order to simulate the real-life scenario.

For any given sampling method, we can calculate the $\mathbf{S}$ matrix directly if a closed-form expression is available. Otherwise, it can be approximated by simulating the sampling process and counting the sampling statistics. In this paper, both uniform random sampling and the ANLS provide closed-form expressions for the calculation of $\mathbf{S}$ matrix.

An important practical issue is that, if the original tensor $\mathcal{X}$ is clamped at $I_{\max}$, the $\mathbf{S}$ matrix must also be clamped. In that case, a last row entry $s_{I_{\max},\nu}$ must present the probability of selecting ν packets from a flow of length greater or equal to $I_{\max}$. This clamping operation can be done by calculating a full-size $\mathbf{S}$ matrix first and setting $\bar{s}_{I_{\max},\nu} \propto \sum_{i=I_{\max}}^{I} s_{i,\nu}$ with a naive assumption that after $I_{\max}$, the original flow sizes are uniformly distributed.

3.3.1. Uniform Sampling Method. In uniform sampling, each packet is processed with a fixed probability of π, irrespective of the flow it belongs to. In this method, the sampling matrix entries $s_{i,\nu}$ are calculated directly through Binomial distribution with i trials and π success probability, i.e.,

$$s_{i,\nu} = \begin{cases} \binom{i}{\nu} \pi^{\nu} (1-\pi)^{i-\nu}, & \text{for } \nu \leq i, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Algorithm 1 describes how the flow table is updated with uniform sampling upon the arrival of a new packet. The algorithm uniformly draws a random number in interval $[0, 1]$, and if it is less than π, the packet is processed; otherwise, the packet is discarded. For processing the packet, a look-up operation is performed on the flow table to find and update the flow that the packet belongs to. If no such flow is found, a new flow is created using the packet's signature.

Figure 2(a) shows the top 10×11 entries of the lower diagonal $\mathbf{S}$ matrices with different sampling probabilities. As the sampling probability π gets smaller, fewer packets from a flow gets observed, and the flow may even be missed when none of its packets are observed. The rightmost sampling matrix shows the case when $\pi = 1/64$, where the matrix has a very high concentration of zero-length sampled flows.

Figure 2(b) shows the original Monday data (the leftmost matrix) and its sampled versions under uniform sampling with the probabilities shown on the top sampling matrices. Here, we see that for $\pi = 1/64$, the observed flow lengths are mostly less than 10, while the majority are not observed at all.

3.3.2. ANLS Sampling Method. The ANLS [10] will be used as the representative of the flow-based adaptive sampling methods. In ANLS, each packet is sampled according to the number of packets previously sampled from its corresponding flow. If a sampled flow has length x, the probability of its next packet to be sampled ($p(x; u)$) is calculated as

$$f(x; u) = \frac{[(1+u)^{x} - 1]}{u}, \quad (4)$$

$$p(x; u) = \frac{1}{[f(x-1; u) - f(x; u)]}. \quad (5)$$

Here, $f(x; u)$ is a monotonically increasing function of flow length x, parametrized with u, which makes $p(x; u)$ monotonically decreasing. u determines the tendency of sampling packets. As u gets smaller, more packets are sampled and estimating original flow lengths gets easier.

The ANLS method is described in details in Algorithm 2. Prior to the sampling, $\mathbf{f}$ and $\mathbf{p}$ are calculated in the SetupANLS function, according to equations (4) and (5). During sampling, for each incoming packet, a look-up operation is performed unconditionally. If the corresponding flow is found, it is updated with probability relative to its current observed size. Otherwise, a new flow is created, ensuring that every flow is observed with at least one packet.

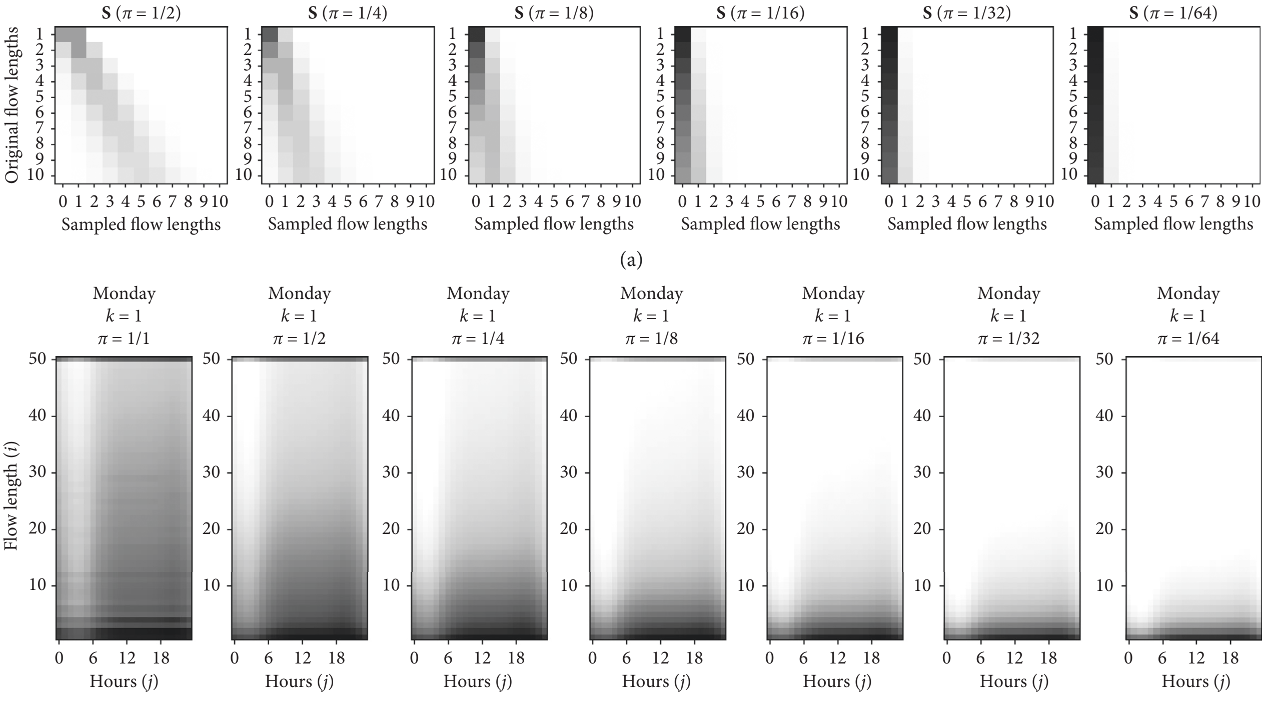
We calculate the sampling matrix $\mathbf{S}$ recursively for ANLS. In this method, the first packet is always sampled since $p(1, u) = 1$ independent of u. We start by assigning all zero sampling probabilities as $s_{:,0} = 0$ and $s_{1,1} = 1$. The recursive equation for calculating the sampling matrix can be deduced as

$$s_{i,\nu} \propto s_{i-1,\nu-1} p(i-1, u) + s_{i-1,\nu} (1 - p(i-1, u)). \quad (6)$$

Figure 3 shows the ANLS sampling matrices for first 10 flow lengths and the Monday data sampled with them, respectively, similarly to Figure 2. First, we can see that when u is small, the $\mathbf{S}$ matrix looks like identity and as u gets larger,

```
function Sample Uniformly Random $\pi$, flow_table, packet
    if $\pi >$ rand_double (0, 1) then
        flow = flow_table·look-up (packet)
        if flow is null then
            flow = new Flow (packet)
        else
            flow·length+ = 1
        flow_table·insert_or_update (flow)
```

Algorithm 1: Uniform packet sampling algorithm.



(a)

(b)

Figure 2: Sampling matrices for two different sampling schemes.

```
function Setup ANLSu
    $\mathbf{f}[0] = 0$
    for $i \in [1, I]$ do
        $f[i] = ((1+u)^i - 1)/u$
        $p[i] = 1/(f[i-1] - f[i])$
    return f, p
function SampleWithANLS (p, flow_table, packet)
    flow = flow_table·look-up (packet)
    if flow is null then
        flow = new Flow (packet)
    else if p[flow·length] > rand_double (0, 1) then
        flow·length+ = 1
    flow_table·insert_or_update (flow)
```

Algorithm 2: ANLS sampling algorithm.

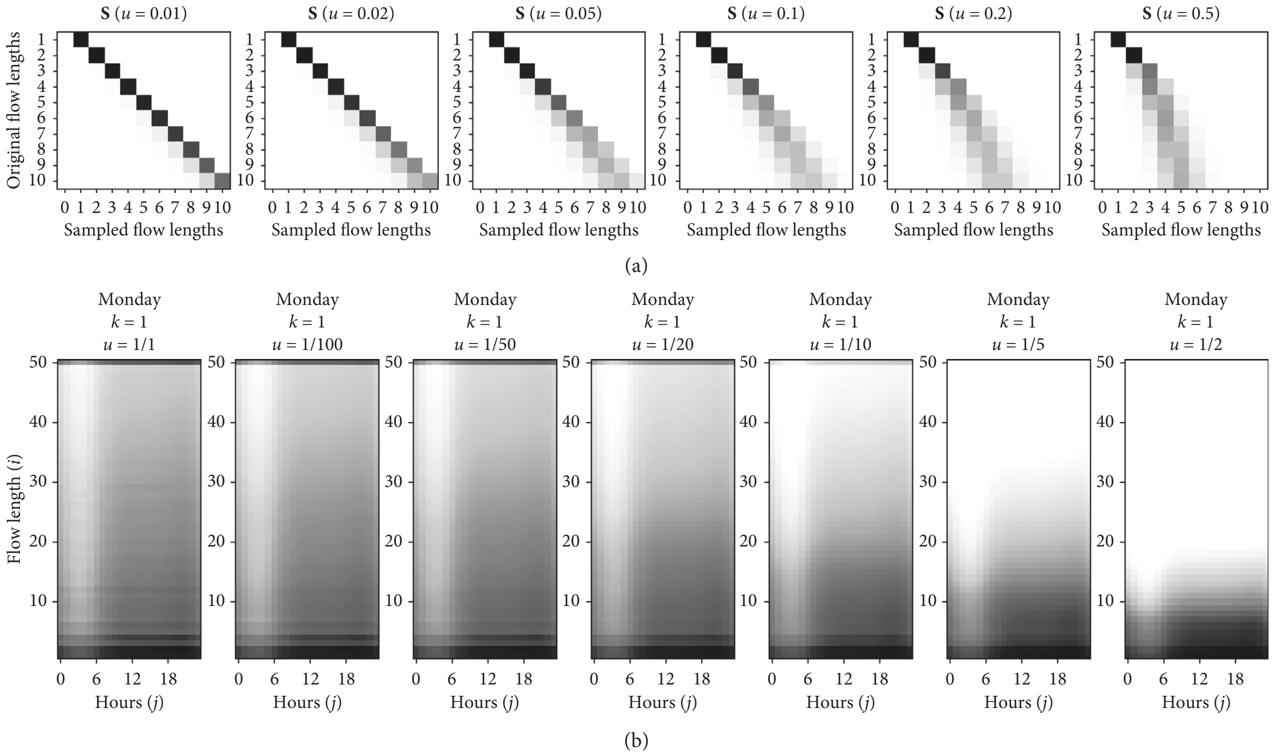


Figure 3: Visualization of the Monday slice with uniform and ANLS sampling methods.

the sampling probability of large flows decreases. Secondly, compared to uniform sampling, the ANLS method has much higher sampling ratios than uniform sampling. However, operating with such high sampling ratios would require specialized hardware in real time.

4. Methodology

Our methodology is based on the nonnegative factorization of the data tensor. Our model, which we call ThinNTF, introduces the sampling matrix as a constant factor to the original NTF with the Poisson–Gamma observation model. The rationale for using factorization for recovering true flow sizes is that the flow size distributions have daily periodic behavior, as we show in Section 3. Inferring the factors, instead of the original matrix, requires less parameter estimation and results in smoother estimates compared to the standard maximum likelihood estimation described in [3]. In this section, we first describe the original tensor factorization model and we provide our novel version: the ThinNTF. At the end of the section, we present the full-Bayesian variational Bayes algorithm for the inference.

4.1. Nonnegative Tensor Factorization. NTF is the generalization of the 2-dimensional NMF model to multiple dimensions. In NTF, an N-dimensional tensor is approximated by the multiplication of lower dimensional factors. Unlike NMF, tensor factorization can be done in multiple ways. In this work, we are going to use the PARAFAC [22–24] factorization scheme. In PARAFAC, an $I_1 \times I_2 \times \cdots \times I_N$ tensor is approximated by $I_n \times R$ matrices for $n \in [1, N]$. Here, R is the number of components, i.e., the number of clusters in the data. Figure 4 shows the PARAFAC factorization of our FLD tensor $\mathcal{X}$, into 3 factors: an $I \times R$ factor $\mathbf{F}$ for representing the flow length clusters, a $J \times R$ factor $\mathbf{H}$ for representing hourly behavior, and a $K \times R$ factor $\mathbf{D}$ for representing the daily behavior of the data. Every single entry of the $\mathcal{X}$ tensor is approximated by

$$x_{i,j,k} \approx \widehat{x}_{i,j,k} = \sum_r f_{i,r} h_{j,r} d_{k,r}. \tag{7}$$

Bro [25] explains that the PARAFAC factorization is unique under certain circumstances, where uniqueness is defined as begin unable to rotate the factorization without loss of fit. NMF and NTF are statistical models that impose nonnegativity constraint without uniqueness property. The uniqueness may be important if individual factors are of special interest. In our case, we are concerned with the estimation of the original data tensor $\mathcal{X}$ from sampled tensor $\mathcal{Y}$, but not the individual factors for any interpretation. Our problem is more close to a missing value imputation problem; hence, uniqueness is not a requirement.

4.2. Thin Nonnegative Tensor Factorization. ThinNTF is basically an NTF with an additional constant factor, which in our case is the sampling matrix $\mathbf{S}$. Figure 5 shows the graphical models of the NTF and the ThinNTF models for factorizing original and sampled flow length observations. In the graphical models, the shaded nodes are the observed entities and the unshaded ones are the latent entities.

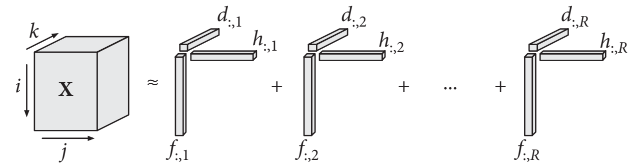
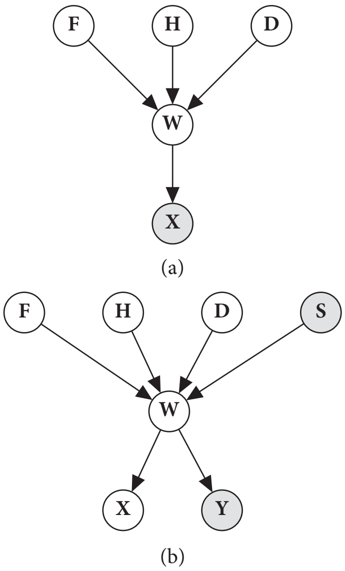


Figure 4: PARAFAC factorization.



(a)

(b)

Figure 5: Graphical models representing the dependency structure of NTF and ThinNTF models in PARAFAC scheme.

In Section 3.3, we have described the sampling process as a matrix multiplication operation with a sampling matrix **S**. In ThinNTF, this sampling matrix operates on the original tensor $\mathscr{X}$ and creates a thinned version of it, which we call $\mathscr{Y}$, by down-sampling its entries according to a sampling scheme, as shown in Figure 6. The entries of $\mathscr{Y}$ tensor $y_{v,j,k}$ presents the number of flows of sampled-length v, at hour j at day k. The $\otimes_1$ operation denotes the 1-mode product of matrix $\mathbf{S}^T$ and tensor $\mathscr{X}$, which corresponds to the set of matrix multiplications $\mathscr{Y}_{:,:,k} = \mathbf{S}^T \mathscr{X}_{:,:,k}$ for $k \in [1, K]$.

In this scheme, one can immediately suspect that $\boldsymbol{x}$ can be estimated by $(\mathbf{S}^T)^{-1} \otimes_1 \mathscr{Y}$. However, this solution is not feasible for several reasons. First, the **S** matrix is not square and hence not invertible. Instead, its pseudoinverse can be calculated, but this does not impose nonnegativity. Moreover, the top slice of the $\mathscr{Y}$ tensor, which stores the number of flows with zero-sampled size, is never observed and hence must be estimated. Therefore, we need a solid statistical model and an inference method to estimate $\mathscr{X}$ under this model.

In ThinNTF, we observe the $\mathscr{Y}$ tensor, but try to factorize the $\mathscr{X}$ tensor, which is latent (Figure 5(b)). In the end, the factors of $\mathscr{X}$ are going to provide us an approximation $\widehat{\mathscr{X}}$ which solves the original flow length distribution reconstruction problem. We mathematically express this approximation as

$$y_{v,j,k} \approx \widehat{y}_{v,j,k} = \sum_{i,r} s_{v,i} f_{i,r} h_{j,r} d_{k,r}, \tag{8}$$

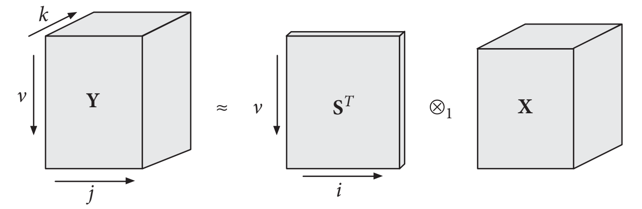


Figure 6: ThinNTF model.

where **F**, **H**, and **D** are described in exactly the same way in the original NTF case. In Subsections 3.3.1 and 3.3.2, we described two different **S** matrices for two different schemes. ThinNTF model can be employed with any sampling method as long as it is described with a sampling matrix.

4.3. Generative Model. Taking the Bayesian approach, we first provide a generative model for the ThinNTF and then describe how we can estimate the posterior probabilities of model parameters (in this case, the factor matrices) conditioned on the sampled flow length observations $\mathscr{Y}$ and the sampling matrix **S** using the well-known Bayes rule. Table 2 contains all tensors and matrices used in the model together with their index sets.

The original and latent data tensor $\mathscr{X}$ and the sampled and observed data tensor **Y** have nonnegative integer entries. The natural probability distribution for this type of count data is the Poisson distribution. We assume that each entry of a latent 5-dimensional tensor $\boldsymbol{\mathscr{W}}$ is drawn from a Poisson distribution whose parameters are functions of sampling matrix **S** and factors **F**, **H**, and **D**, such as

$$w_{v,i,j,k,r} \sim \mathscr{P}\mathscr{O}\left(w_{v,i,j,k,r}; s_{v,i} f_{i,r} h_{j,r} d_{k,r}\right), \tag{9}$$

where the Poisson distribution is defined as

$$\mathscr{P}\mathscr{O}(w; \lambda) = \exp(w \log \lambda - \lambda - \log \Gamma(w+1)). \tag{10}$$

We choose the prior distributions for the factor entries as the Gamma distribution since it is the conjugate prior of Poisson distribution [26]. For each entry of factor **F**, we write

$$f_{i,r} \sim \mathscr{G}\left(f_{i,r}; a^f_{i,r}, \frac{b^f_{i,r}}{a^f_{i,r}}\right), \tag{11}$$

where the Gamma distribution is described as

$$\mathscr{G}(f; \kappa, \Theta) = \exp\left((\kappa - 1)\log f - \frac{f}{\Theta} - \kappa \log \Theta - \log \Gamma(\kappa)\right), \tag{12}$$

with shape parameter κ and scale parameter θ. In our generative model, the parameters for Gamma distributions are $\kappa = a^f_{i,r}$ and $\Theta = b^f_{i,r}/a^f_{i,r}$ respectively. This means that the mean of $f_{i,r}$ is $\kappa\Theta = b^f_{i,r}$, which is independent of $a^f_{i,r}$. The variance of $f_{i,r}$ becomes $\kappa\Theta^2 = (b^f_{i,r})^2/a^f_{i,r}$, which means that as $a^f_{i,r}$ gets smaller, the factors gets sparser.

In order to avoid repetition, we are going to omit the equations regarding the factors **H** and **D** throughout the paper. These factors behave exactly like factor **F**, and it is easy

Table 2: Tensors in the model and their corresponding index sets.

Tensor	Index set	Description
$\mathcal{X}$	$i,\ j,\ k$	Original flow length tensor
$\mathcal{Y}$	$v,\ j,\ k$	Sampled flow length tensor
$\mathbf{M}$	$v,\ j,\ k$	Mask tensor
$\boldsymbol{\mathcal{W}}$	$v,\ i,\ j,\ k,\ r$	Latent variable tensor
$\mathbf{F}$	$i,\ r$	Flow length factor
$\mathbf{H}$	$j,\ r$	Hour of day factor
$\mathbf{D}$	$k,\ r$	Day of week factor
$\mathbf{S}$	$i,\ v$	Sampling matrix
$\mathbf{A}^F$, $\mathbf{B}^F$	$i,\ r$	Gamma priors for $\mathbf{F}$
$\mathbf{A}^H$, $\mathbf{B}^H$	$j,\ r$	Gamma priors for $\mathbf{H}$
$\mathbf{A}^D$, $\mathbf{B}^D$	$k,\ r$	Gamma priors for $\mathbf{D}$

to derive equations related to these factors once their corresponding equation for $\mathbf{F}$ is given.

Finally, we generate $\mathcal{X}$ and $\mathcal{Y}$ tensor from $\boldsymbol{\mathcal{W}}$. Each entry $w_{v,i,j,k,r}$ of $\boldsymbol{\mathcal{W}}$ can be interpreted as the number of original flows of length i, generated on hour j, day k, by cluster r, and observed as length v. By summing $\boldsymbol{\mathcal{W}}$ over dimensions cluster (r) and original lengths (i), we get the sampled observations tensor $\mathcal{Y}$. Similarly, by summing $\boldsymbol{\mathcal{W}}$ over dimensions cluster (r) and sampled lengths (v), we get the original flow length tensor $\mathcal{X}$. The whole generative process is summarized in Algorithm 3. The set of all indexes and tensors in the model are summarized in Tables 1 and 2, respectively.

4.4. Variational Bayes. After defining the generative model, we can inter the factors $\mathbf{F}, \mathbf{H}$, and $\mathbf{D}$ of a sampled flow length observation tensor $\boldsymbol{\mathcal{Y}}$. In the original NMF paper, Lee, and Seung [17] provide fixed-point update equations for inferring the factors. Bro [25] gives similar fixed-point equations for updating the factors in PARAFAC factorization. Cemgil [19] shows that these updates correspond to the Kullback–Leibler minimization between the original matrix (or tensor $\boldsymbol{\mathcal{X}}$) and the approximated one ($\hat{\boldsymbol{\mathcal{X}}}$) and also provides a full Bayesian variational algorithm for the matrix factorization. Ermis et al. [15] provide a similar variational algorithm for the Gamma–Poisson tensor factorization.

We start our Bayesian inference by calculating the posterior distributions over the factors $\mathbf{F}, \mathbf{H}$, and $\mathbf{D}$ conditioned on observed tensor $\boldsymbol{\mathcal{Y}}$. For notational clarity, we introduce $\theta = (\mathbf{A}^F, \mathbf{B}^F, \mathbf{A}^H, \mathbf{B}^H, \mathbf{A}^D, \mathbf{B}^D)$ as the list of model hyperparameters. The log-likelihood observing $\boldsymbol{\mathcal{Y}}$ under the model parameters θ is written as

$$\log p(\boldsymbol{\mathcal{Y}} \mid \theta, \mathbf{S}) = \log \int_{\mathbf{F},\mathbf{H},\mathbf{D}} d\mathbf{F}\, d\mathbf{H}\, d\mathbf{D} \sum_{\boldsymbol{\mathcal{W}}} p(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \theta, \mathbf{S}). \tag{13}$$

This log-likelihood is intractable due to the integration over the latent factors, but it is lower bounded as

$$\begin{aligned} \log p(\boldsymbol{\mathcal{Y}} \mid \theta, \mathbf{S}) &\leq \mathcal{L}_\theta \\ &= \langle \log p(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \theta, \mathbf{S}) \rangle q(\boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D}) \\ &\quad + \mathcal{H}_{q(\boldsymbol{\mathcal{W}},\mathbf{F},\mathbf{H},\mathbf{D})}, \end{aligned} \tag{14}$$

where q is an auxiliary joint distribution of latent factors. This bound is tight when $q(\boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D}) = p(\boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \mathcal{Y}, \theta, \mathbf{S})$. However, this is also intractable to calculate. Instead, we use a variational approximation [27] for q such that

$$\begin{aligned} q(\boldsymbol{\mathcal{W}}) &\propto \exp\left(\langle \log p(\mathcal{Y}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \theta) \rangle_{q(\mathbf{F},\mathbf{H},\mathbf{D})}\right), \\ q(\mathbf{F}) &\propto \exp\left(\langle \log p(\mathcal{Y}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \theta) \rangle_{q(\boldsymbol{\mathcal{W}},\mathbf{H},\mathbf{D})}\right), \\ q(\mathbf{H}) &\propto \exp\left(\langle \log p(\mathcal{Y}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \theta) \rangle_{q(\boldsymbol{\mathcal{W}},\mathbf{F},\mathbf{D})}\right), \\ q(\mathbf{D}) &\propto \exp\left(\langle \log p(\mathcal{Y}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \theta) \rangle_{q(\boldsymbol{\mathcal{W}},\mathbf{F},\mathbf{H})}\right), \end{aligned} \tag{15}$$

where we iteratively update the posterior distribution of each factor by calculating the expectation of the logarithm of the full joint likelihood $p(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D})$ under the posteriors of all other latent factors.

4.5. Update Equations. Here, we provide the update equations for $q(\boldsymbol{\mathcal{W}})$ and $q(\mathbf{F})$. The updates of $q(\mathbf{H})$ and $q(\mathbf{D})$ can be easily deduced from the update equations of $q(\mathbf{F})$. The full joint likelihood whose expectation will be calculated at each step is

$$\begin{aligned} \mathcal{J}_\theta &= \log p(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} | \theta) \\ &= \log p(\boldsymbol{\mathcal{Y}} \mid \boldsymbol{\mathcal{W}}) + \log p(\boldsymbol{\mathcal{W}} \mid \mathbf{F}, \mathbf{H}, \mathbf{D}) + \log p(\mathbf{F} \mid \theta) \\ &\quad + \log p(\mathbf{H} \mid \theta) + \log p(\mathbf{D} \mid \theta), \end{aligned} \tag{16}$$

where $\boldsymbol{\mathcal{Y}} \mid \boldsymbol{\mathcal{W}}$ is a degenerate distribution ($\delta()$) to make sure the summation of $\sum_{i,r} \boldsymbol{\mathcal{W}}$ equals $\boldsymbol{\mathcal{Y}}$. By inserting the necessary Poisson and Gamma distributions given in the generative model in the equation (16), we get the following expression:

$$\begin{aligned} \mathcal{J}_\theta = {} & \sum_{v,j,k} m_{v,j,k} \log \delta\left(y_{v,j,k} - \sum_{i,r} w_{v,i,j,k,r} \right) \\ & + \sum_{i,r} \sum_{v,j,k} m_{v,j,k} \Big(w_{v,i,j,k,r} \log s_{v,i} f_{i,r} h_{j,r} d_{k,r} - s_{v,i} f_{i,r} h_{j,r} d_{k,r} \\ & - \log \Gamma\left(w_{v,i,j,k,r} + 1 \right) \Big) \\ & + \sum_{i,r} \left(\left(a^f_{i,r} - 1 \right) \log f_{i,r} - f_{i,r} \frac{a^f_{i,r}}{b^f_{i,r}} - a^f_{i,r} \log \frac{b^f_{i,r}}{a^f_{i,r}} - \log \Gamma\left(a^f_{i,r} \right) \right) \\ & + \sum_{j,r} \left(\left(a^h_{j,r} - 1 \right) \log h_{j,r} - h_{j,r} \frac{a^h_{j,r}}{b^h_{j,r}} - a^h_{j,r} \log \frac{b^h_{j,r}}{a^h_{j,r}} - \log \Gamma\left(a^h_{j,r} \right) \right) \\ & + \sum_{k,r} \left(\left(a^d_{k,r} - 1 \right) \log d_{k,r} - d_{k,r} \frac{a^d_{k,r}}{b^d_{k,r}} - a^d_{k,r} \log \frac{b^d_{k,r}}{a^d_{k,r}} - \log \Gamma\left(a^d_{k,r} \right) \right). \end{aligned} \tag{17}$$

4.5.1. Update Rule for $q(\boldsymbol{\mathcal{W}})$. Considering the terms in the log-likelihood expression in equation (17), that only includes $w_{v,i,j,k,r}$, we find that

```
function RandInit ($\mathbf{S}, \mathbf{A}^F, \mathbf{B}^F, \mathbf{A}^H, \mathbf{B}^H, \mathbf{A}^D, \mathbf{B}^D$)
        //Sample factor $\mathbf{F}$ from Gamma ($\mathbf{A}^F, \mathbf{B}^F$)
        for all $i \in [1, I]$, $r \in [1, R]$ do
            $f_{i,r} \sim \mathscr{G}(f_{i,r}\,;\, a^f_{i,r}, (b^f_{i,r}/a^f_{i,r}))$
        //Sample factor $\mathbf{H}$ from Gamma ($\mathbf{A}^H, \mathbf{B}^H$)
        for all $k \in [1, K]$, $r \in [1, R]$ do
            $h_{j,r} \sim \mathscr{G}(h_{j,r}\,;\, a^h_{j,r}, (b^h_{j,r}/a^h_{j,r}))$
        //Sample factor $\mathbf{D}$ from Gamma ($\mathbf{A}^D, \mathbf{B}^D$)
        for all $j \in [1, J]$, $r \in [1, R]$ do
            $d_{k,r} \sim \mathscr{G}(d_{k,r}\,;\, a^d_{k,r}, (b^d_{k,r}/a^d_{k,r}))$
        //Sample latent tensor $\mathscr{W}$ from Poisson distributions
        for all $\nu \in [1, I+1]$, $i \in [1, I]$, $j \in [1, J]$, $k \in [1, K]$, $r \in [1, R]$ do
            $w_{\nu,i,j,k,r} \sim \mathscr{PO}(w_{\nu,i,j,k,r}\,;\, s_{\nu,i} f_{i,r} h_{j,r} d_{k,r})$
        return {F, H, D, $\mathscr{W}$}
function GenerateData ($\mathbf{S}, \mathbf{A}^F, \mathbf{B}^F, \mathbf{A}^H, \mathbf{B}^H, \mathbf{A}^D, \mathbf{B}^D$)
        //Randomly initialize factors and latent tensor
    {F, H, D, $\mathscr{W}$} $\longleftarrow$ RandInit ($\mathbf{S}, \mathbf{A}^F, \mathbf{B}^F, \mathbf{A}^H, \mathbf{B}^H, \mathbf{A}^D, \mathbf{B}^D$)
        //Generate original tensor $\boldsymbol{\mathscr{X}}$
    for all $i \in [1, I]$, $j \in [1, J]$, $k \in [1, K]$ do
        $\boldsymbol{\mathscr{X}}_{i,j,k} = \sum_{\nu,r} \mathscr{W}_{\nu,i,j,k,r}$
        //Generate sampled tensor $\mathscr{Y}$
    for all $\nu \in [1, I+1]$, $j \in [1, J]$, $k \in [1, K]$ do
        $y_{\nu,j,k} = \sum_{i,r} w_{\nu,i,j,k,r}$
    return {F, H, D, $\mathscr{W}$, $\mathscr{X}$, $\mathscr{Y}$}
```

Algorithm 3: TNTF generative model.

$$
\begin{aligned}
q\left(w_{\nu,:,j,k,:}\right) \propto\; & \exp m_{\nu,j,k} \log \delta\left(y_{\nu,j,k} - \sum_{i,r} w_{\nu,i,j,k,r} \right) \\
& + \sum_{i,r} m_{\nu,j,k}\left(w_{\nu,i,j,k,r} \log s_{\nu,i} f_{i,r} h_{j,r} d_{k,r} \right. \\
& \left. - \log \Gamma\left(w_{\nu,i,j,k,r} + 1 \right) \right) \\
& \propto \text{multinomial}\left(w_{\nu,j,k,:}, x_{i,j,k}, p_{\nu,i,j,k,r} \right)^{m_{\nu,j,k}},
\end{aligned}
\tag{18}
$$

where, $w_{\nu,j,k,:,:}$ becomes multinomial distributed. The expectation of $\mathscr{W}$ is calculated as

$$
\begin{aligned}
p_{\nu,i,j,k,r} &= \frac{\exp\left(s_{\nu,i} + \langle \log f_{i,r} \rangle + \langle \log h_{j,r} \rangle + \langle \log d_{k,r} \rangle \right)}{\sum_{i,r} \exp\left(s_{\nu,i} + \langle \log f_{i,r} \rangle + \langle \log h_{j,r} \rangle + \langle \log d_{k,r} \rangle \right)}, \\
\langle w_{\nu,i,j,k,r} \rangle &= y_{\nu,j,k} p_{\nu,i,j,k,r}.
\end{aligned}
\tag{19}
$$

4.5.2. Update Rule for $q(\mathbf{F})$. Similarly, considering the terms in log-likelihood equation (17) that only includes $f_{i,r}$, we find that

$$
\begin{aligned}
q\left(f_{i,r}\right) \propto & \left(\sum_{\nu,j,k} m_{\nu,j,k} \langle w_{\nu,i,j,k,r} \rangle + a^f_{i,r} - 1 \right) \log f_{i,r} \\
& - \left(\sum_{\nu,j,k} m_{\nu,j,k} s_{\nu,i} \langle h_{j,r} \rangle \langle d_{k,r} \rangle + \frac{a^f_{i,r}}{b^f_{i,r}} \right) f_{i,r} \\
& \propto \text{Gamma}\left(f_{i,r}; \alpha^f_{i,r}, \beta^f_{i,r} \right),
\end{aligned}
\tag{20}
$$

where $f_{i,r}$ becomes Gamma distributed with shape and scale parameters

$$
\alpha^f_{i,r} = a^f_{i,r} + \sum_{\nu,j,k} m_{\nu,j,k} \langle w_{\nu,i,j,k,r} \rangle, \tag{21}
$$

$$
\beta^f_{i,r} = \left(\frac{a^f_{i,r}}{b^f_{i,r}} + \sum_{\nu,j,k} m_{\nu,j,k} s_{\nu,i} \langle h_{j,r} \rangle \langle d_{k,r} \rangle \right)^{-1}. \tag{22}
$$

We calculate the expectation of $f_{i,r}$ and the logarithm of $f_{i,r}$ as

$$
\langle f_{i,r} \rangle = \alpha^f_{i,r} \beta^f_{i,r}, \tag{23}
$$

$$
\langle \log f_{i,r} \rangle = \Psi\left(\alpha^f_{i,r} \right) + \log \beta^f_{i,r}. \tag{24}
$$

The variational Bayes algorithm that uses the above equations is given in Algorithm 4. The calculation of the lower bound is given in Appendix. The exact derivations of all equations can be found in [28].

4.6. Computational Complexity. The nonnegative tensor factorization is an NP-hard problem [29]. The variational Bayes algorithm we introduced in Algorithm 4 is an iterative solution that converges to a local maximum solution. The complexity of each iteration is determined by the leading term, which is the equation (19). In general, calculating a ThinNTF model with R components for a κ-dimensional tensor with all dimensions of length N has $O(\kappa N^{(\kappa+1)} R)$ complexity for a single iteration.

```
function ThinNTF_VB ($\mathcal{Y}$, $\mathbf{S}$, $\mathbf{A}^F, \mathbf{B}^F, \mathbf{A}^H, \mathbf{B}^H, \mathbf{A}^D, \mathbf{B}^D$)
        //Randomly initialize factors and latent tensor
    $\{\mathbf{F}, \mathbf{H}, \mathbf{D}, \mathcal{W}\} \longleftarrow$ RandInit ($\mathbf{S}, \mathbf{A}^F, \mathbf{B}^F, \mathbf{A}^H, \mathbf{B}^H, \mathbf{A}^D, \mathbf{B}^D$)
    repeat
        Calculate $\alpha^f_{i,r}, \beta^f_{i,r}$ and $\langle f_{i,r} \rangle$ as in equations (21)–(23).
        Calculate $\alpha^h_{j,r}, \beta^h_{k,r}$ and $\langle h_{j,r} \rangle$ similarly.
        Calculate $\alpha^d_{k,r}, \beta^d_{k,r}$ and $\langle d_{k,r} \rangle$ similarly.
        Calculate $\langle \log f_{i,r} \rangle$ as in equation (24).
        Calculate $\langle \log f_{i,r} \rangle$ similarly.
        Calculate $\langle \log f_{i,r} \rangle$ similarly.
        Calculate $\langle w_{\nu,i,j,k,r} \rangle$ as in equation (19).
        Calculate lower bound
    until Max iterations are reached or lower bound converged
    return F, H, D, $\mathcal{X}$
```

Algorithm 4: Variational Bayes algorithm.

5. Data Collection

Applying a tensor model to the flow length estimation problem requires high-volume data collected over a long period, to capture the timely behavior of the network. The already available online data sets do not fulfill this requirement. Therefore, we collected our own real-world data from a mobile network service provider in Turkey [30]. The architecture of our system and the description of the data we collected are presented as follows.

5.1. System Architecture. The system architecture of a mobile operator's general packet radio service (GPRS) network infrastructure, including radio access and core network elements, is illustrated in Figure 7. IP traffic generated or received by mobile devices between mobile station (MS) and packet data network PDN, e.g., IP Multimedia Subsystem (IMS), is tunneled in the core network of mobile operators through serving GPRS support node (SGSN) and gateway GPRS support node (GGSN) via the user data part of the GPRS tunneling protocol (GTP) [31]. The GTP message exchanges include information such as the size of the traffic, IP session start and end time, and device and user identifiers.

The Gn interface (Gn is an interface between SGSN and GGSN where GTP is the main protocol for network packets flowing through) carries user packets to be transferred between the mobile users and the Internet together with control packets necessary for the universal mobile telecommunications service (UMTS) core network [32]. All packets in this channel are carried by the GTP, which is an IP-based protocol for carrying GPRS data within UMTS networks, used for data encapsulation in order to keep the core network independent of the protocols that are being used between MS and the packet-switched network.

The Gn interface carries mainly two types of GTP message structures or packets: GTP-C and GTP-U. GTP-C is used for signaling between SGSN and GGSN in core network which carries packet data protocol (PDP) context messages such as activating and deactivating user session, configuring service parameters or updating the session. GTP-U is used for transmitting user data between the radio access network and core network. In our experiments, we filtered out GTP-C packets (since they are not considered to be part of a flow due to flow definition), which makes 10% of the total Gn traffic. Therefore, the sampling is applied to GTP-U packets only. GTP is carried mainly over UDP.

5.2. Data Extraction Process. The mobile operator network consists of several districts with more than 10 regional core areas throughout Turkey. The average total traffic in all regional areas consists of approximately over 15 billion packets in the uplink direction and over 20 billion packets in the downlink direction daily. This corresponds to approximately 80 terabytes of total data flowing in uplink and downlink daily inside the mobile operator's core network as a whole. In this work, the Gn interface which connects the SGSN and GGSN nodes are mirrored, and the network traffic is transferred into a FLD server located at mobile operator's technology center in the core network. A speed of 200 Mbit/sec at peak hours can be observed through one of the mirrored interfaces in the core network.

We monitored the network traffic in one of the servers of a mobile operator continuously for 10 days. We developed a packet extraction tool inside the monitoring server shown in Figure 7, which parses each GTP-U packet and stores the packet signature together with packet length and arrival time, discarding their payload. The stored network data is processed offline for extracting the true flow lengths.

After the data collection and flow extraction, the total number of packets collected is found to be 4×10^{11}, which makes up around 2.5×10^{10} flows. Figure 8 shows the cumulative flow length distribution of the data. We see that most of the flows have less than 5 packets, and 99.9% of them have less than 100 packets.

6. Experiments and Results

We designed two sets of experiments in order to verify our model: synthetic and real-world experiments. In each set, we sampled the original data with both uniform and ANLS models with different sampling parameters. Then, we tried to recover the original tensor with ThinNTF models. The

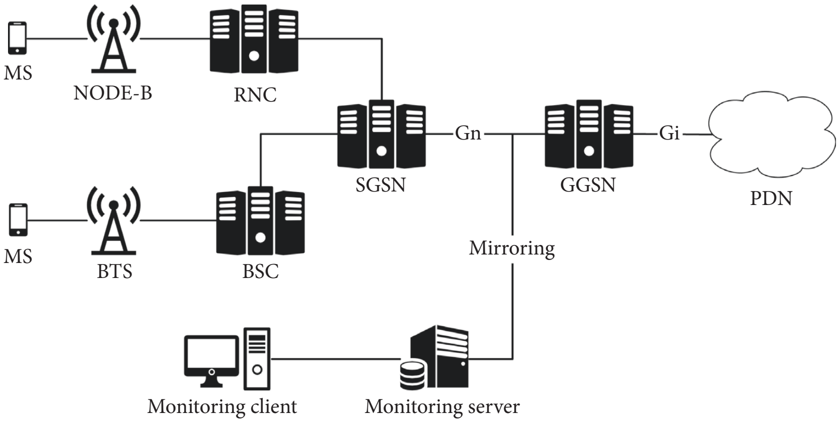


Figure 7: Placement of our monitoring server inside the premises of the mobile operator running a commercial cellular network.

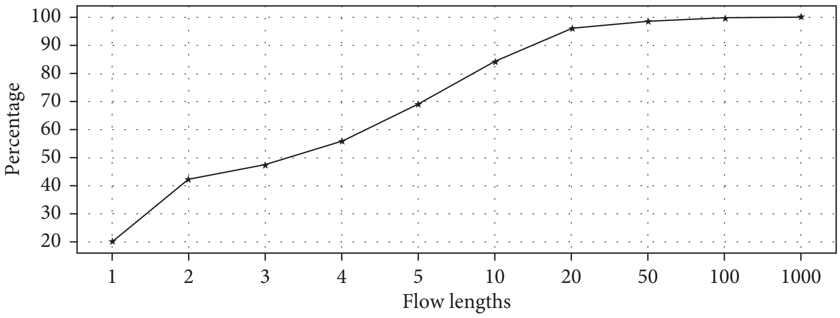


Figure 8: Cumulative flow lengths in the real-world data.

ThinNTF model takes a single parameter R which is the number of components in each factor. Additionally, we also represented data as $I \times JK$ matrix by unfolding the $\mathcal{X}$ tensor in the first dimension as described in [28] and applied the 2-dimensional version of ThinNTF, which we simply call thin nonnegative matrix factorization (ThinNMF). For Uniform sampling, we used the maximum likelihood estimation (MLE) defined in [3] as the baseline. For ANLS, we used both MLE and its own unbiased estimator of the model as baselines.

Both ThinNMF and ThinNTF models explain the data as a linear combination of R flow length distributions, stored in the columns of $\mathbf{F}$ matrix. In the ThinNMF model, we have JK coefficient sets for this combination. On the other hand, in ThinNTF, we have J coefficients for hour-of-day and K coefficients for day-of-week. The Cartesian products of these coefficient sets make a total of JK coefficients and create a dependency between the hour and day attributes. Therefore, we expect that ThinNTF captures the periodicity and give better estimates.

During the experiments, we always run the stochastic algorithms, i.e., ThinNMF, ThinNTF, and MLE, for 10 times and keep the parameters of the model with the highest lower bound value. Then, we reported the success of our algorithm with the weighted mean relative distance (WMRD) metric as this was used in all previous flow size estimation works. The WMRD is a metric which gives more weights to the relative differences that occur with larger frequency. It is formulated as

$$\operatorname{wmrd}(\mathbf{x}, \widehat{\mathbf{X}}) = \frac{\sum_i |x_i - \widehat{x}_i|}{\sum_i (x_i + \widehat{x}_i)/2}, \qquad (25)$$

where $\mathbf{x}$ is an original flow size distribution measured at the end of the hour and $\widehat{x}$ is its estimated version. For the whole tensor $\mathcal{X}$, we calculate the average WMRD value.

Additionally, we report the Kullback–Leibler divergence between the original and the estimated tensors, since this is the metric minimized during the variational Bayes algorithm. The KL divergence between two distributions $\mathbf{x}$ and $\widehat{X}$ is calculated as

$$\operatorname{KL}(\mathbf{x}, \widehat{x}) = \sum_i x_i \log\left(\frac{\widehat{x}_i}{x_i}\right). \qquad (26)$$

6.1. Experiments on Synthetic Data. We prepared our synthetic experiments to test the validity of our models. In this experiment set, we used the generative model of the ThinNTF model as described in Algorithm 3 to generate a small network with maximum flow size $I = 10$, $J = 24$ and $K = 7$. The original synthetic flow length distribution $\mathcal{X}$ is generated by a generative model with 3 components, where each component is a column in the $\mathbf{F}$ factor. We selected these 3 components as exponential, inverted exponential, and uniform random distributions. Therefore, in experiments, we used ThinNMF-R3 and ThinNTF-R3 models, where the suffix R3 shows that the model has 3 components.

We sampled the synthetic data with uniform and ANLS sampling methods with different sampling parameters. The sampling was done simply by randomly drawing a sampled size for each flow according to the sampling probabilities in the **S** matrix. By this way, we ignored the flow splitting problem, and this gave us an ideal data for the ThinNTF model. We report and compare the mean standard deviation of these WMRD values for all experiments.

The ThinNTF model always performed best with the uniform sampling model, as shown in Table 3 as expected. On ANLS sampling, the MLE and the ANLS estimators performed better with high sampling probabilities, when $u \in (0.01, 0.02, 0.05)$, as shown in Table 4. On the other hand, when the sampling probability of the ANLS model decreases, the ThinNMF helped with better estimations. From the initial results, we conclude that the factorization is definitely helpful for more difficult uniform sampling method and helps lower the sampling probabilities in flow-based packet sampling. The results are also visible in Figure 9.

Table 3: Uniform sampling results on synthetic data.

Period	ThinNMF-R3	ThinNTF-R3	MLE
2	0.53	0.49	0.88
4	0.63	0.59	1.20
8	0.65	0.61	1.29
16	0.68	0.61	1.41
32	0.74	0.61	1.37
64	0.85	0.61	1.50

Table 4: ANLS sampling results on synthetic data.

U	ThinNMF-R3	ThinNTF-R3	MLE	ANLS
0.01	0.29	0.27	0.09	0.15
0.02	0.31	0.29	0.17	0.27
0.05	0.33	0.32	0.36	0.28
0.1	0.35	0.34	0.51	0.38
0.2	0.38	0.36	0.59	0.67
0.5	0.48	0.47	0.72	0.70

6.2. Experiments on Real-World Data. The original data collected from a mobile network provider as we described in Section 5 are sampled with both sampling methods. However, this time, we simulated the real network offline by feeding the packets one by one to the monitoring server, as described in Section 5, This way, we were able to create the actual conditions on a sampler installed at a network provider's backbone. This also created the flow splitting problem, since we applied a 30 seconds timeout in our flow hash. We set the maximum flow length as $I = 100$, meaning that $\mathcal{X}_{100,:,:}$ entries show the count of flows that have more than 99 packets. This clamping decision was made according to the cumulative distribution of flow lengths as shown in Figure 8. We also clamped the sampling matrices **S** so that they exactly match the model.

Since the number of components in the original flow distribution is unknown, we run our experiments with $R \in [2, 3, 4]$ components for ThinNMF and ThinNTF. The rest of the experiment is similar to the synthetic one. The sampled Y matrix with shape $100 \times 24 \times 7$ is factorized and $\widehat{\mathcal{X}}$ is reconstructed with the estimated factors. We reported and compared the mean and standard deviation of 24×7 WMRD and KL values.

The factorization models, both ThinNMF and ThinNTF helped lower the WMRD score in both uniform and ANLS sampling methods. ThinNTF-R4 model consistently gave lower error than the MLE baseline for uniform model as shown in Table 5 and Figure 10. Indeed, our factorization framework improved results overall for uniform sampling. However, since recovering true estimates in uniform sampling is quite difficult, we see less impact of the factorization as the sampling ratio increases.

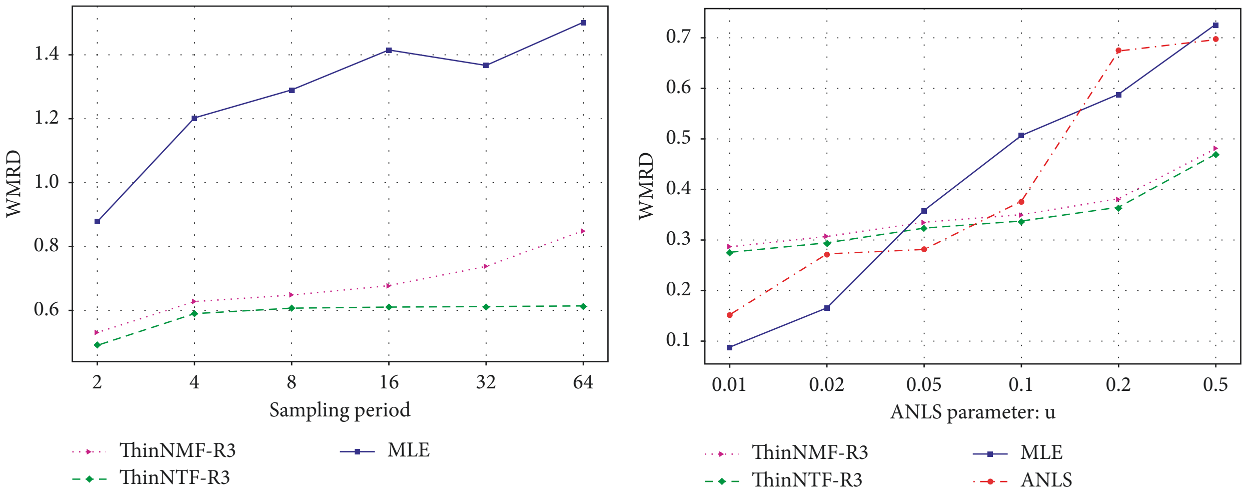
Figure 10 also gives the KL values between the true and estimated flow length distributions. While the scale of this metric is different, it gives consistent results with the WMRD. This shows that our model, which minimizes the KL metric, also minimizes the commonly used WMRD metric; hence, the model is suitable for this problem.

Another important issue is that for uniform sampling, 3-way factorization is more successful than the 2-way factorization. The periodicity information which is captured by the ThinNTF model helps improve the estimates and makes it a more successful model for this sampling method.

In ANLS, all our factorization models gave lower error values than the MLE and unbiased estimator of ANLS as shown in Table 6 and Figure 11. Since ANLS is a more powerful sampling method than uniform sampling, the effect framework is slightly less for small sampling parameter u. However, both ThinNMF and ThinNTF gave better result while sampling smaller number of packets (when u is large). Furthermore, since we are trying to recover the same original data in both experiments, we can compare our ThinNMF and ThinNTF models under two sampling methods. We see that, in both methods, as the number of components increases, the models gave lower error rates. However, with the uniform sampling method, 3-dimensional methods give better results, while with ANLS, 2-dimensional models perform slightly better.

6.3. Effect of Clamping. The choice of where to clamp the data can be given by multiple factors. First of all, one can set the clamping value $I_{\max}$ according to a value of special interest. Otherwise, we would like to choose a small $I_{\max}$ so that we deal with a dense tensor and less parameters. On the other hand, we would like to set $I_{\max}$ as high as possible so that the clamped portion of the data is as small as possible.

We run the best algorithms found in previous section for uniform and ANLS sampling methods with $I_{\max} \in \{25, 50, 75, 100\}$. The WMRD values are given in Figure 12. In both methods, $I_{\max} = 25$ gave relatively poor performance and $I_{\max} = 100$ was generally the best choice. Also the results with $I_{\max} \geq 50$ are closer to each other. This is consistent with the graph in Figure 8, where the cumulative flow lengths do not change much after $I_{\max} = 50$. A final remark from this experiment is that as the clamping value increases,



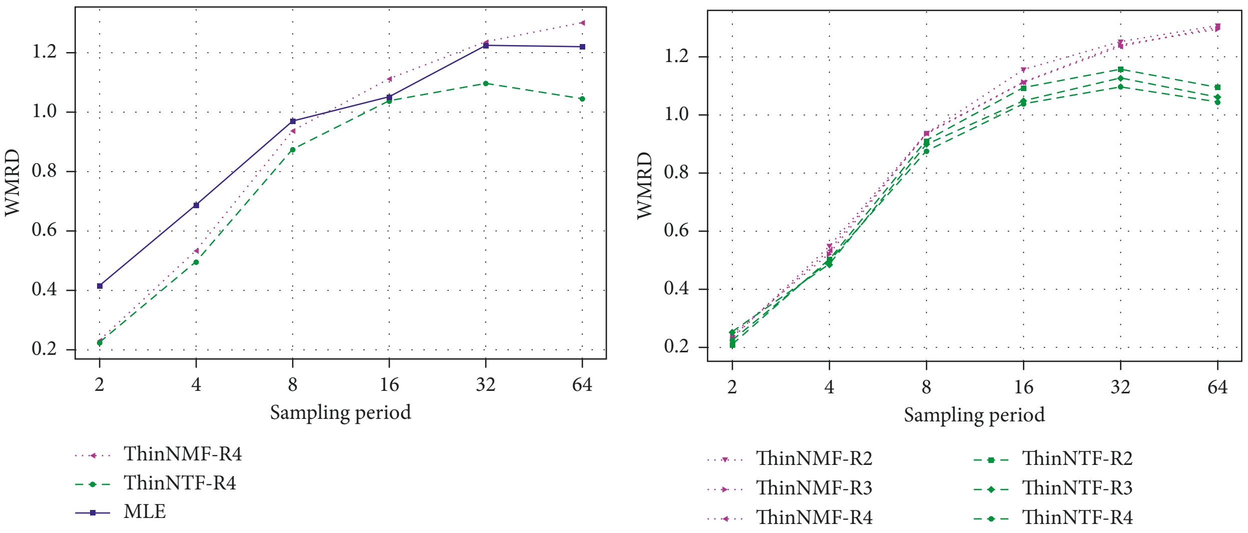
(a)

(b)

Figure 9: Synthetic experiment results.

Table 5: Uniform sampling results on real data.

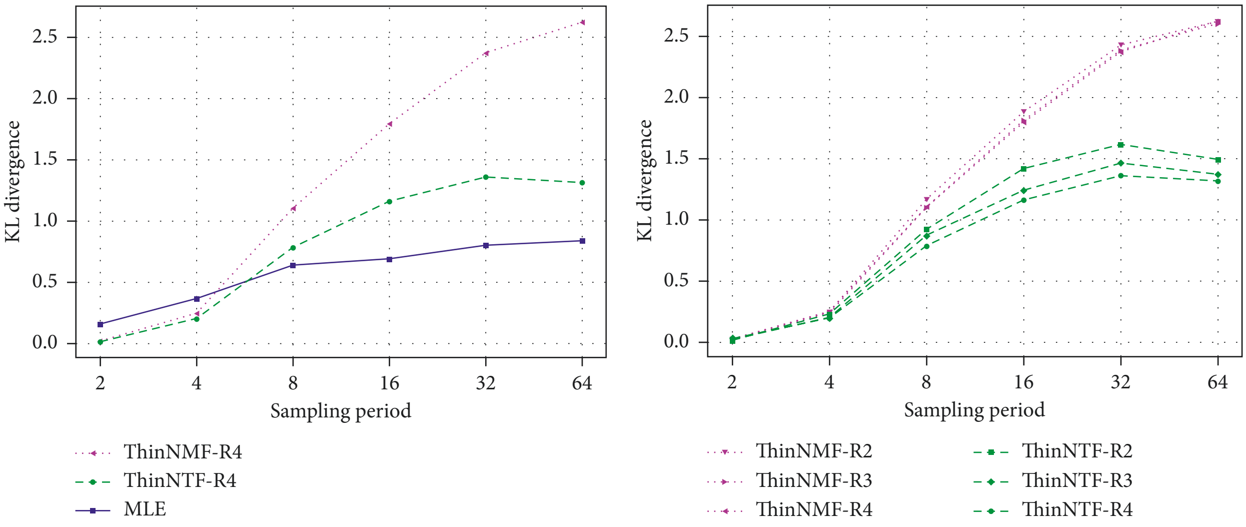
Period	ThinNMF			ThinNTF			MLE
	$R = 2$	$R = 3$	$R = 4$	$R = 2$	$R = 3$	$R = 4$	
2	0.23	0.24	0.23	0.21	0.25	0.22	0.41
4	0.55	0.52	0.53	0.50	0.48	0.49	0.69
8	0.94	0.93	0.94	0.91	0.90	0.87	0.97
16	1.15	1.11	1.11	1.09	1.05	1.04	1.05
32	1.25	1.24	1.24	1.16	1.13	1.10	1.22
64	1.31	1.29	1.30	1.09	1.06	1.04	1.22



(a)

(b)

Figure 10: Continued.



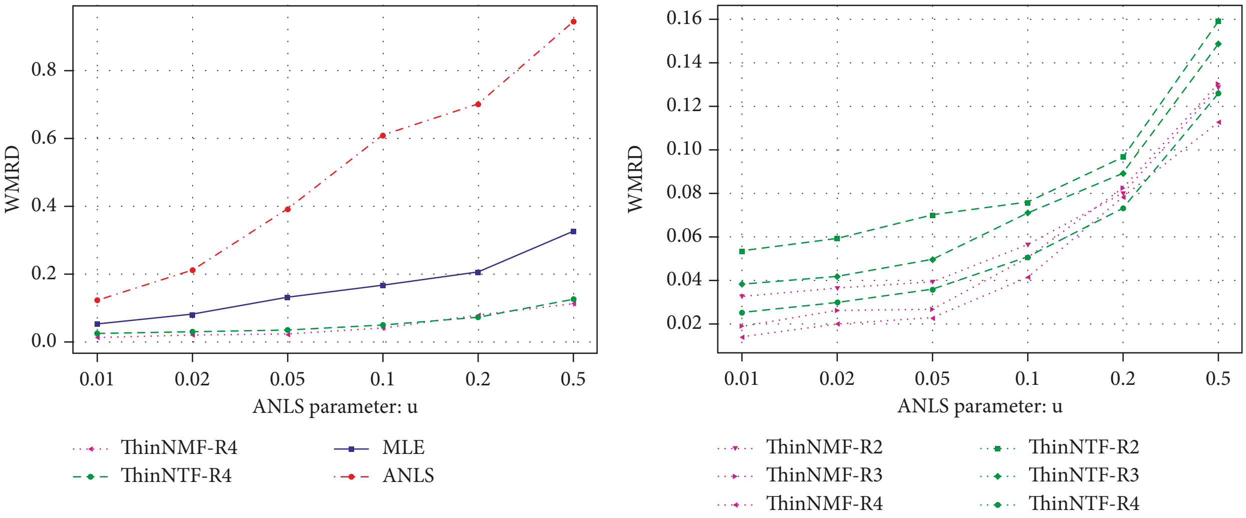
(c)

(d)

Figure 10: Real-world data results with Uniform sampler.

Table 6: ANLS sampling results on real data.

U	ThinNMF			ThinNTF			MLE	ANLS
	$R = 2$	$R = 3$	$R = 4$	$R = 2$	$R = 3$	$R = 4$		
0.01	0.03	0.02	0.01	0.05	0.04	0.03	0.05	0.12
0.02	0.04	0.03	0.02	0.06	0.04	0.03	0.08	0.21
0.05	0.04	0.03	0.02	0.07	0.05	0.04	0.13	0.39
0.1	0.06	0.05	0.04	0.08	0.07	0.05	0.17	0.61
0.2	0.08	0.08	0.08	0.10	0.09	0.07	0.21	0.70
0.5	0.13	0.13	0.11	0.16	0.15	0.13	0.33	0.94



(a)

(b)

Figure 11: Continued.

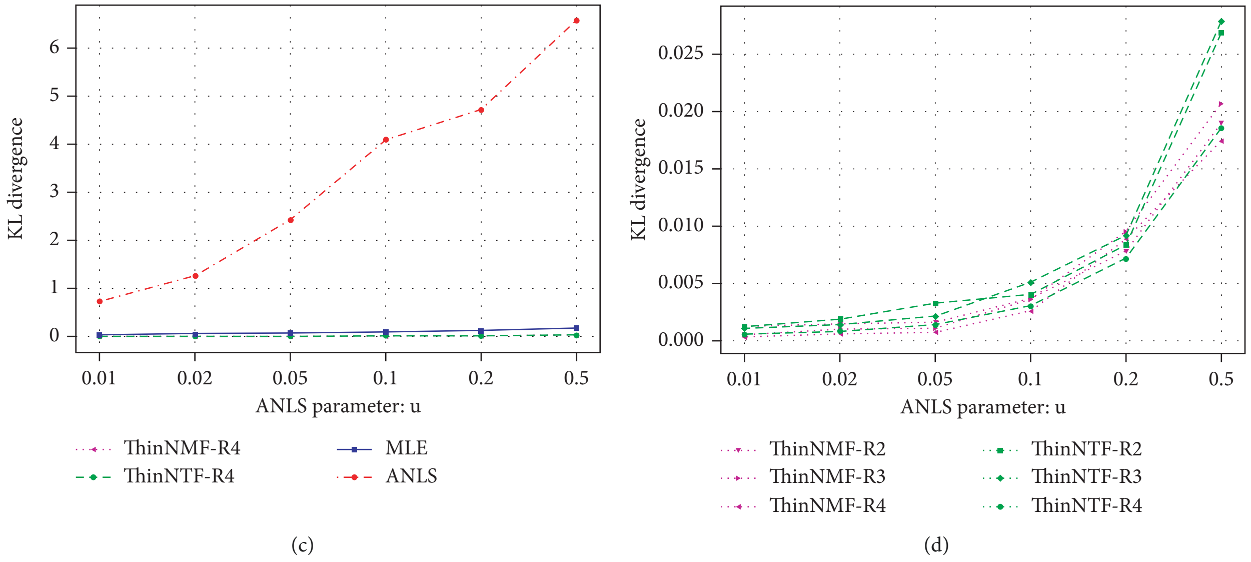


Figure 11: Real-world data results with ANLS sampler.

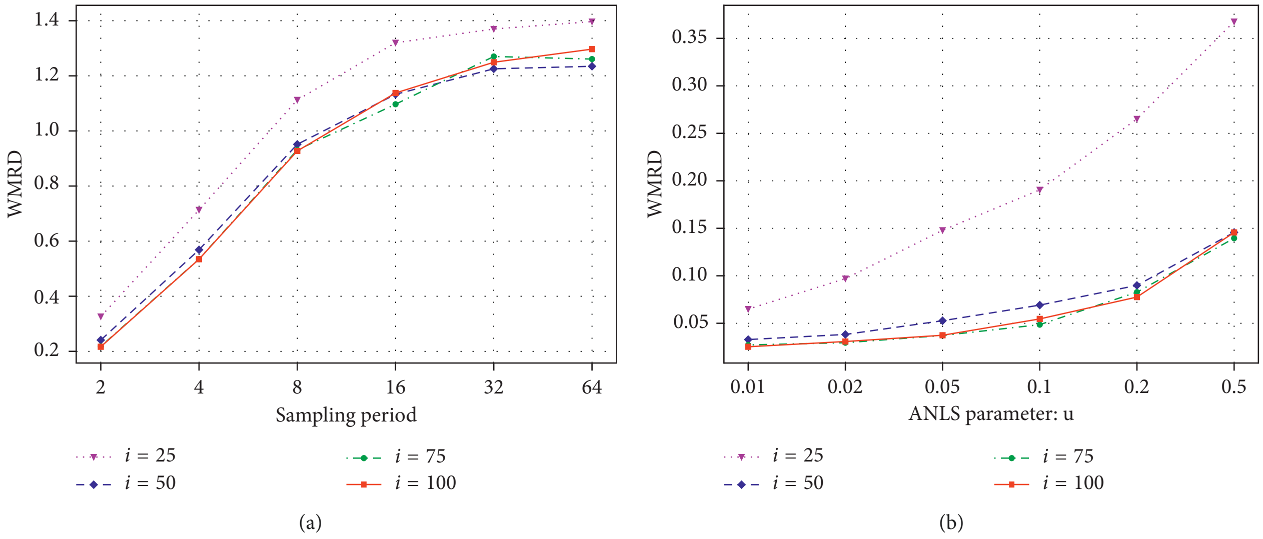


Figure 12: Clamping experiments.

estimation becomes harder with small sampling rates. This explains the results in uniform sampling with sampling rate 1/64.

7. Conclusions

In this work, we introduced a novel nonnegative tensor factorization model called ThinNTF, which extends the classic nonnegative tensor factorization with an additional constant factor that can represent a network packet sampling method. We showed that this model can be employed to improve the current reconstruction algorithms in recovering the original flow length distributions.

We tested our model with two different types of sampling methods: the uniform packet sampling method and a flow-based packet sampling method, called ANLS. We described how to use these methods by showing how to build their sampling matrices.

In order to test our model, we collected high-volume data from a mobile network provider for a long period in order to observe the periodical behavior of the flow length distribution. In experiments on synthetic and real-world data, our models gave promising results by lowering the estimation errors compared to the baselines of each sampling method. We conclude that our model can be used to decrease estimation errors or to decrease the sampling probabilities without increasing the estimation error.

An important issue left as future work is the online execution of the ThinNTF model. Theoretically, the ThinNTF model can be used online once sufficient data from the target network is collected, and the flow length distribution components, i.e., the **F** factor, are inferred. The power of our model is that this inference can be done directly from the sampled observations. Once the **F** factor is estimated, for each incoming observation, the corresponding entries in other factors can be inferred by keeping **F** constant during the inference. Moreover, **F** can be updated periodically, say weekly, in a sliding window fashion and kept up to date with the networks flow length behavior.

Appendix

Variational Lower Bound Calculation

The calculation of the lower bound includes a few arithmetic tricks. We provide a Bayesian nonnegative matrix factorization [28] tutorial for the detailed derivation and coding tricks. The final form of the lower bound equation is

$$\begin{aligned}
\mathscr{L}_{\theta} &= \langle \log p(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}}, \mathbf{F}, \mathbf{H}, \mathbf{D} \mid \theta) \rangle_{q(\boldsymbol{\mathcal{W}},\mathbf{F},\mathbf{H},\mathbf{D})} + H_{q(\boldsymbol{\mathcal{W}},\mathbf{F},\mathbf{H},\mathbf{D})} \\
&= -\sum_{\nu,i,j,k,r} m_{\nu,j,k} \left(s_{\nu,i} \langle f_{i,r} \rangle \langle h_{j,r} \rangle \langle d_{k,r} \rangle \right) \\
&\quad + \sum_{i,r} \left(-\langle f_{i,r} \rangle \frac{a^{f}_{i,r}}{b^{f}_{i,r}} - a^{f}_{i,r} \log \frac{b^{f}_{i,r}}{a^{f}_{i,r}} - \log \Gamma\left(a^{f}_{i,r}\right) \right) \\
&\quad + \sum_{j,r} \left(-\langle h_{j,r} \rangle \frac{a^{h}_{j,r}}{b^{h}_{j,r}} - a^{h}_{j,r} \log \frac{b^{h}_{j,r}}{a^{h}_{j,r}} - \log \Gamma\left(a^{h}_{j,r}\right) \right) \\
&\quad + \sum_{k,r} \left(-\langle d_{k,r} \rangle \frac{a^{d}_{k,r}}{b^{d}_{k,r}} - a^{d}_{k,r} \log \frac{b^{d}_{k,r}}{a^{d}_{k,r}} - \log \Gamma\left(a^{d}_{k,r}\right) \right) \\
&\quad - \sum_{\nu,j,k} m_{\nu,j,k} \log \Gamma\left(y_{\nu,j,k} + 1\right) \\
&\quad - \sum_{\nu,i,j,k,r} m_{\nu,j,k} \langle w_{\nu,i,j,k,r} \rangle \log \left(\frac{p_{\nu,i,j,k,r}}{s_{\nu,i}} \right) \\
&\quad + \sum_{i,r} \left(\alpha^{f}_{i,r} \left(\log \beta^{f}_{i,r} + 1 \right) + \log \Gamma\left(\alpha^{f}_{i,r}\right) \right) \\
&\quad + \sum_{j,r} \left(\alpha^{h}_{j,r} \left(\log \beta^{h}_{k,r} + 1 \right) + \log \Gamma\left(\alpha^{h}_{j,r}\right) \right) \\
&\quad + \sum_{k,r} \left(\alpha^{d}_{k,r} \left(\log \beta^{d}_{k,r} + 1 \right) + \log \Gamma\left(\alpha^{d}_{k,r}\right) \right).
\end{aligned} \tag{A.1}$$

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] G. Varghese and C. Estan, "The measurement manifesto," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 1, pp. 9–14, 2004.

[2] N. Duffield, "Sampling for passive internet measurement: a review," *Statistical Science*, vol. 19, no. 3, pp. 472–498, 2004.

[3] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 933–946, 2005.

[4] B. F. Ribeiro, D. F. Towsley, T. Ye, and J. Bolot, "Fisher information of sampled packets: an application to flow size estimation," in *Proceedings of the 6th ACM SIGCOMM on Internet measurement*, pp. 15–26, Rio de Janeriro, Brazil, October 2006.

[5] N. Hohn and D. Veitch, "Inverting sampled traffic," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, pp. 222–233, ACM Press, Miami Beach, FL, USA, October 2003.

[6] L. Yang and G. Michailidis, "Sampled based estimation of network traffic flow characteristics," in *Proceedings of the 26th IEEE International Conference on Computer Communications*, pp. 1775–1783, Anchorage, AK, USA, May 2007.

[7] Cisco netflow, http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html.

[8] ntop, http://www.ntop.org/.

[9] A. Kumar, M. Sung, J. Xu, and E. W. Zegura, "A data streaming algorithm for estimating subpopulation flow size distribution," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 61–72, 2005.

[10] C. Hu, B. Liu, S. Wang, J. Tian, Y. Cheng, and Y. Chen, "ANLS: adaptive non-linear sampling method for accurate flow size measurement," *IEEE Transactions on Communications*, vol. 60, no. 3, pp. 789–798, 2012.

[11] C. Hu, B. Liu, H. Zhao et al., "Discount counting for fast flow statistics on flow size and flow volume," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 970–981, 2014.

[12] A. Kumar and J. J. Xu, "Sketch guided sampling-using on-line estimates of flow size for adaptive data collection," in *Proceedings of the 2006 IEEE INFOCOM*, Barcelona, Spain, April 2006.

[13] A. Kumar, J. Jun Xu, and J. Jia Wang, "Space-code bloom filter for efficient per-flow traffic measurement," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2327–2339, 2006.

[14] C. Hu, S. Wang, J. Tian, B. Liu, Y. Cheng, and C. Yan, "Accurate and efficient traffic monitoring using adaptive nonlinear sampling method," in *Proceedings of the 2008 IEEE INFOCOM*, Washington, DC, USA, March 2008.

[15] B. Ermis and A. T. Cemgil, "A bayesian tensor factorization model via variational inference for link prediction," 2014, https://arxiv.org/abs/1409.8276.

[16] A. Kumar, M. Sung, J. Xu, and J. Wang, "Data streaming algorithms for efficient and accurate estimation of flow size distribution," *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 177–188, 2004.

[17] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[18] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, no. 1, pp. 556–562, 2001.

[19] A. T. Cemgil, "Bayesian inference for nonnegative matrix factorisation models," *Computational Intelligence and Neuroscience*, vol. 2009, Article ID 785152, 17 pages, 2009.

[20] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[21] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 267–273, Toronto, Canada, July 2003.

[22] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1–4, pp. 164–189, 1927.

[23] J. Douglas Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[24] R. A. Harshman, "Foundations of the parafac procedure: model and conditions for an"explanatory"multi-mode factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1969.

[25] R. Bro, "Parafac. tutorial and applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, no. 2, pp. 149–171, 1997.

[26] N. Johnson, A. Kemp, and S. Kotz, "Univariate discrete distributions," *Wiley Series in Probability and Statistics*, Wiley, Hoboken, NJ, USA, 2005.

[27] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: a review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[28] Bayesian nonnegative matrix factorization tutorial, https://github.com/bariskurt/bptf.

[29] S. A. Vavasis, "On the complexity of nonnegative matrix factorization," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1364–1377, 2010.

[30] B. Kurt, E. Zeydan, U. Yabas, I. A. Karatepe, G. K. Kurt, and A. T. Cemgil, "A network monitoring system for high speed network traffic," in *Proceedings of the 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–3, London, UK, June 2016.

[31] 3GPP, "3GPP evolved packet system (EPS); evolved general packet radio service (GPRS) tunnelling protocol for control plane (GTPV2-C); stage 3," Tech. Rep. 3GPP TS 29.274 13.4.0, ETSI, Sophia Antipolis, France, 2015.

[32] A. Springer and R. Weigel, *The UMTS (Universal Mobile Telecom Standard) Physical Layer Basics, Standard, and Frontend Matters*, Springer-Verlag, Berlin, Germany, 2002.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 5437908, 14 pages
https://doi.org/10.1155/2019/5437908

Research Article

Low Cost Antenna Array Based Drone Tracking Device for Outdoor Environments

Marcos T. de Oliveira,[1] **Ricardo K. Miranda**,[2] **João Paulo C. L. da Costa**,[3,4] **André L. F. de Almeida**,[5] **and Rafael T. de Sousa Jr.**[2]

[1] *Department of Mechanical Engineering, Faculty of Technology, University of Brasília, Brazil*
[2] *Department of Electrical Engineering, Faculty of Technology, University of Brasília, Brazil*
[3] *Department of Electrical Engineering and Department of Mechanical Engineering, Faculty of Technology, University of Brasília, Brazil*
[4] *Elektronische Fahrwerksysteme GmbH, Ingolstadt, Germany*
[5] *Wireless Telecom Research Group, Federal University of Ceará, Fortaleza, CE, Brazil*

Correspondence should be addressed to Marcos T. de Oliveira; socramteix@gmail.com

Received 1 March 2019; Revised 18 May 2019; Accepted 2 June 2019; Published 27 June 2019

Academic Editor: Miguel Garcia-Pineda

Applications of direction of arrival (DoA) techniques have dramatically increased in various areas ranging from the traditional wireless communication systems and rescue operations to GNSS systems and drone tracking. Particularly, police forces and security companies have drawn their attention to drone tracking devices, in order to provide the safeness of citizens and of clients, respectively. In this paper, we propose a low cost antenna array based drone tracking device for outdoor environments. The proposed solution is divided into hardware and software parts. The hardware part of the proposed device is based on off-the-shelf components such as an omnidirectional antenna array, a 4-channel software defined radio (SDR) platform with carrier frequency ranging from 70 MHz to 6 GHz, a FPGA motherboard, and a laptop. The software part includes algorithms for calibration, model order selection (MOS), and DoA estimation, including specific preprocessing steps and a tensor-based estimator to increase the DoA accuracy. We evaluate the performance of our proposed low cost solution in outdoor scenarios. According to our measurement campaigns, we show that when the array is in the front fire position, i.e., with a DoA ranging from -60° to 60°, the maximum and the average DoA errors are 6° and $1{,}9^{\circ}$, respectively.

1. Introduction

Applications of direction of arrival (DoA) techniques have dramatically increased in various areas ranging from the traditional wireless communication systems [1, 2] and rescue operations [3] to GNSS systems [4–7] and drone tracking in public and private events. In the last years Unmanned Aerial Vehicles (UAVs) have been a major concern of airspace control bodies and military due to possible terrorist attacks and illegal activities. In 2015, there were more than nine hundred incidents involving drones and aircrafts in the United States [8], whereas, in April 2016, a UAV reached an aircraft landing at the Heathrow airport in London [9]. In 2016 in Dubai, four drones invaded the airport interrupting the landings and take-offs, causing an estimated loss of one million dollars [10]. In October 2017 in Canada, the first reported collision of a drone and a commercial airplane has occurred [11]. Recently police forces and security companies have drawn their attention to drone tracking devices in order to provide the safeness of citizens and of clients, respectively. In this sense, the development of low cost devices for drone tracking is fundamental to fit such demands.

In general, the DoA estimation techniques can be broadly classified into conventional beamforming techniques, maximum likelihood techniques, and subspace-based techniques [12–14]. In [15] the authors proposed to estimate the DoA of a signal impinging the Electronically Steerable Parasitic Array Radiator (ESPAR) antenna with twelve parasitic elements by using a support vector machine (SVM) technique. In the anechoic chamber the result of the experiments reaches 0.67° estimation error. No hardware details were provided.

To overcome the effects of multipath propagation on the performance of DoA estimation, the authors of [16] proposed a frequency domain multipath resolution subspace-based approach, which makes RSS-based DoA estimation applicable in multipath scenarios for small-size and low-power sensor networks. This approach was verified with Monte Carlo simulations with high SNR.

Aiming at the growth of connected cars systems, the authors in [17] developed a 4×4 MIMO antenna system and proposed the DoA function for a circular phased array antenna. Since the focus of this paper is on the methodology and basic characteristics of DoA function, no measurement or hardware information was provided.

In [18], DoA estimation using an ESPAR with 12 parasitic elements and one active monopole is carried out for wireless sensor network (WSN) applications. The authors calibrated the ESPAR array using an anechoic chamber. Since the focus of [18] is on the calibration, no outdoor or indoor measurement campaigns were performed by the authors.

An improvement of ESPAR antennas is proposed in [19]. The authors developed a Multiple Beam Parasitic Array Radiator (MBPAR) antenna that can realize six beams at the same time without the use of diodes, which increases the communication capacity. To validate the proposed design, a prototype was fabricated at 2.45 GHz. The antenna has the efficiency from 94.2 % to 95.7 % over the 2.4G-WLAN bands. No DoA measurements were provided.

In [20], a square shaped 16 element antenna array is connected to switches so that a four-channel SDR can select four antennas at each side of the square, allowing a 360° DoA estimation in outdoor environments. Each side of the square performs a ±45° azimuth estimation. According to the authors, three Yagi antennas were used as sources at specific points, and a maximum DoA error of 5° is achieved. No information is provided about the real distance between the sources and the receive array.

In [21], a four-element quasi-Yagi antenna array system is applied for DoA estimation using the MUSIC algorithm, whereas the Minimum Description Length (MDL) criterion is used to estimate the number of dominant multipath components. Only two measurements were performed for two specific positions, showing an error of 1°. However, no information is provided about the experimental scenario.

In [22], several DoA estimation techniques are compared considering a horizontal uniform linear array (ULA) with 12 elements inside an anechoic chamber. The measurements were conducted varying the DoA from −20° to 20° in steps of 4°. The DoA estimation errors were smaller than 2°. The Min-Norm approach [12] outperformed MUSIC [23] although it has a higher standard deviation.

Finally, in [24], the authors developed system using five-port reflectometers that allow simultaneously measuring the DoA and Time of Arrival (ToA) of coherent and incoherent signals, connected to seven quasi-Yagi antennas, with one reflectometer for each antenna. The MUSIC algorithm is applied for the DoA estimation, providing an error of 2° for one source and 0,5° for two sources. The measurements were performed in a nonreflective environment.

In order to detect the presence of drones and to track them, there is a variety of mechanical, optical, or antenna array based solutions in the market. For instance, the mechanical solution in [25] detects a drone within 3 km for targets up to 55 cm in diameter and classifies the model of the drone within 1.1 km. The position accuracy (azimuth) in [25] is 1°. In [26], a rechargeable portable drone tracking device can detect and indicate the direction of a drone in a 360° plane even with weak line of sight (LoS) component. The device in [26] allows the communication with other devices by using an Application Programming Interface (API) framework. No technical information and patent about the principles behind the device in [26] and its DoA accuracy were provided. In [27], an antenna array based system is shown to detect with a 1 km range and with 1° accuracy or with a 7 km range and with 3° accuracy.

In [28], a mobile application (app) is proposed for drone detection. According to the developers, the app has an average range of 106 meters. The system allows the detection of almost 95 % of all types of drones. However, the solution in [28] does not indicate the position or the direction of the drone.

In terms of drone tracking, there are recent works on received signal strength (RSS)-based DoA estimation. For example, [29, 30] propose to estimate the DoA using arrays of Yagi-Uda directional antennas for the localization of drones exploiting by their incoming NTSC signal in a measurement campaign. The work [29] proposes a complete hardware and software framework using arrays of directional antennas and formulates a novel DoA estimation correction procedure. In [30], a novel DoA estimation algorithm for the localization of drones is validated by using an AD-FMCOMMS5-EBZ software defined radio (SDR). Finally, [31] implemented tests to detect and locate UAVs at 900MHz. The authors used MUSIC combined with spatial smoothing and MDL. The work [30] has a similar objective to our proposal; however the authors did not concern in to present the accuracy of the system and in use of low cost equipment.

In this paper, we propose a low cost antenna array based drone tracking device for outdoor environments. To the best of our knowledge, there are no state-of-the-art low cost off-the-shelf antenna array based devices applied to drone tracking. The problem of drone tracking is challenging due to the several possible modulation schemes for the data transmission, multipath propagation, and the possible long operational distances. The proposed framework can also exploit tensor-based techniques, such as the Parallel Factor Analysis (PARAFAC). In contrast to the subspace-based methods, the tensor-based approach shows to be robust in real scenarios.

The remainder of this paper is divided as follows. In Section 2, the data model is presented. Next, in Section 3, we propose a low cost antenna array based drone tracking device for outdoor environments, including a complete description of the hardware and software, and the steps involved for assembling, calibration, and signal processing. In Section 4, we validate our proposed solution by means of measurement campaigns in an outdoor scenario. In Section 5, conclusions are drawn.

2. Data Model

We assume d be far-field sources transmitting narrow-band signals. These planar wavefront signals impinge over a receive antenna array with M omnidirectional elements that are uniformly and linearly disposed. The space Δ between two adjacent antennas is equal to $\lambda/2$, where λ is the wavelength of the carrier signal. The received signals at the antenna array can be written in a matrix form as follows:

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N} \in \mathbb{C}^{M\times N}, \tag{1}$$

where $\mathbf{A} \in \mathbb{C}^{M\times d}$ is the steering matrix and its i-th steering vector is given by

$$\mathbf{a}(\mu_i) = \begin{bmatrix} 1 & e^{j\mu_i} & e^{2j\mu_i} & \cdots & e^{j(M-1)\mu_i} \end{bmatrix}^{\mathrm{T}} \in \mathbb{C}^{M\times 1}, \tag{2}$$

where μ_i is the spatial frequency that can be mapped into the direction of arrival of the i-th source, θ_i, by the following expression: $\mu_i = 2\pi\Delta\sin\theta_i/\lambda$. $\mathbf{S} \in \mathbb{C}^{d\times N}$ is the symbol matrix with N being the number of snapshots. $\mathbf{N} \in \mathbb{C}^{M\times N}$ stands for the noise matrix whose elements are assumed to be Complex-Valued Circularly Symmetric Gaussian and identically and independently distributed (i.i.d.) random variables.

Given (1) and assuming that the noise and the signal are uncorrelated, the covariance matrix can be computed by

$$\mathbf{R}_{\mathrm{XX}} = \mathrm{E}\left\{\mathbf{x}\mathbf{x}^{\mathrm{H}}\right\} = \mathbf{A}\mathbf{R}_{\mathrm{SS}}\mathbf{A}^{\mathrm{H}} + \mathbf{R}_{\mathrm{NN}}, \tag{3}$$

where $\mathbf{x}$ is one column vector from $\mathbf{X}$, $(\cdot)^{\mathrm{H}}$ is the Hermitian operator, and $E\{\cdot\}$ is the expected value operator. In practice, the sample covariance matrix is calculated as follows:

$$\widehat{\mathbf{R}}_{\mathrm{XX}} = \frac{\mathbf{X}\mathbf{X}^{\mathrm{H}}}{N} \in \mathbb{C}^{M\times M}. \tag{4}$$

The DoA techniques used along this paper exploit the sample covariance matrix in (4). As shown in Section 3, the matrix $\mathbf{X}$ is preprocessed before we compute the sample covariance matrix $\widehat{\mathbf{R}}_{\mathrm{XX}}$.

The goal of our proposed drone tracking device is to estimate the direction of arrival (DoA) θ_1 of the line of sight (LoS) component from a drone in an outdoor scenario. We assume that there is no obstruction of the LoS component. Therefore, the LoS component is assumed to have the greatest power in comparison with the non-LoS components. Mathematically, we can express it as

$$\left\|\mathbf{a}(\mu_1)\,\mathbf{s}(\mu_1)\right\|_{\mathrm{F}} > \left\|\mathbf{a}(\mu_i)\,\mathbf{s}(\mu_i)\right\|_{\mathrm{F}}, \tag{5}$$

for $i = 2, \ldots, d$. The operator $\|\cdot\|_{\mathrm{F}}$ stands for the Frobenius norm.

3. Proposed Low Cost Antenna Array Based Drone Tracking Device

In this section, we detail the proposed low cost antenna array based drone tracking device. In Section 3.1, we describe the steps for the hardware calibration. The calibration ensures phase alignment for all the four channels of the SDR, allowing the DoA estimation. In Section 3.2, we present the assembling of the hardware components of the proposed drone tracking device. In Section 3.3, we propose a signal processing framework for DoA estimation.

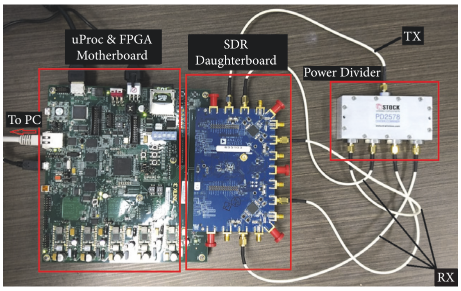


Figure 1: Assembled components for the hardware calibration of four receive channels. The components are a microprocessor (uProc), a FPGA motherboard, a SDR, a power divider, and cables.

3.1. Hardware Assembling for the Calibration. In order to perform the measurements, the SDR should be calibrated, such that all the receive channels become in phase. The phase imbalance may be caused by different time initialization of the oscillators and by hardware imperfections. The hardware vendor provides a software [32] for clock calibration of the local oscillator. However, this software does not perform phase calibration.

Therefore, in order to perform the phase calibration, the hardware components are first assembled according to Figure 1. Note that the SDR transmits the signal from one channel and receives it in four channels that should be calibrated.

As shown in Figure 1, the SDR is a 4×4 MIMO platform named ADFMCOMMS5 [33], with two AD9361 [34] Integrated Circuits (ICs) that contain 2 transmitters and two receivers each, ranging from 70 MHz to 6.0 GHz, and have a channel bandwidth ranging from 200 kHz to 56 MHz. The platform is connected to a microprocessor and a FPGA motherboard [35] that configures the SDR and transmits the SDR data to the PC. As shown in Figure 1, the cables for calibration should have the same length. Moreover, a power division component is included in order to lead the signal to the four receive channels at the same time and to reduce the power of the transmitted signal to avoid damaging.

To compensate the phase errors previously explained, the first step is to extract the phase of the elements of the matrix $\mathbf{X}$. The phase $\phi(m,i)$ is defined as follows:

$$\phi(m,i) = \angle x(m,i) = \arctan\left(\frac{\operatorname{Im}\{x(m,i)\}}{\operatorname{Re}\{x(m,i)\}}\right), \tag{6}$$

where $x(m,i)$ is the element in position m, i of the measured matrix $\mathbf{X}$. The operators $\angle$, $\operatorname{Im}\{\cdot\}$ and $\operatorname{Re}\{\cdot\}$ stand for the phase operator, the imaginary part of a complex number and the real part of a complex number, respectively.

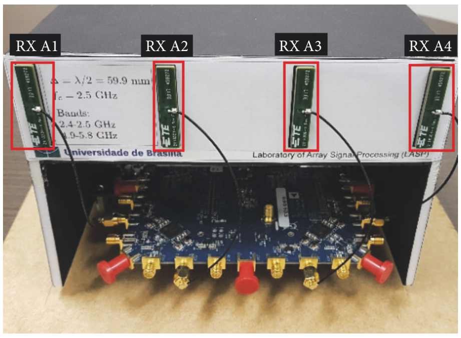


Figure 2: Proposed low cost antenna array based drone tracking device with 4 element ULA and a 4 × 4 MIMO SDR.

Since each channel is related to each line of the matrix **X**, in order to compute the phase shift between two channels, we compute the phase difference of two consecutive antennas

$$\omega(m,:) = \mathbf{\Phi}(m,:) - \mathbf{\Phi}(r,:) \in \mathbb{C}^{1\times N}, \qquad (7)$$

where $\mathbf{\Phi}$ stands for the matrix containing the calculated phases by (6), r indicates the reference channel, and m varies from 1 to M. This reference channel can be randomly selected from 1 to M and is the input of the SDR that is used as a reference to compensate the phase imbalance from the other inputs. Since the vector $\omega(m,:)$ is the m-th row of matrix $\mathbf{\Omega}$, in case $m = i$, the i-th row of $\mathbf{\Omega}$ is filled with zeros. Finally, since the phase difference may slightly vary for different samples in the same row of $\mathbf{\Omega}$ due to the thermal noise, we compute the arithmetic mean of the elements of each row of $\mathbf{\Omega}$, obtaining the vector $\overline{\omega} \in \mathbb{C}^{M\times 1}$ and its m-th element is given by

$$\overline{\omega}(m) = \frac{1}{N}\sum_{n=1}^{N}\mathbf{\Omega}(m,n). \qquad (8)$$

Hence, in order to compensate the phase shift between two different channels, we compute the vector $\mathbf{c} \in \mathbb{C}^{M\times 1}$. The m-th element is given by

$$\mathrm{c}(m) = \mathrm{e}^{-j\overline{\omega}(m)}. \qquad (9)$$

Note that the compensation vector **c** is computed only once for the system initialization. The calibrated outputs of the antenna array are given by the following expression:

$$\mathbf{X}_{\mathrm{c}} = \operatorname{diag}\{\mathbf{c}\}\,\mathbf{X}, \qquad (10)$$

where the operator diag{·} transforms its argument vector into the main diagonal of a diagonal matrix.

3.2. Hardware Assembling for the Drone Tracking Measurement Campaign. After the hardware has been calibrated, the next step is to assembly it in order to perform the measurements.

The four-element omnidirectional antenna array is connected to the calibrated hardware composed of the FPGA motherboard and SDR daughterboard according to Figure 2. Each antenna is dual band (from 2400 MHz to 2483.5 MHz and from 4900 MHz to 5875 MHz) [36] and has linear polarization with 3.7 dBi of gain. The space Δ between two consecutive antennas is equal to 5.99 cm. The operational frequency $f = 2.5$GHz is the maximum frequency that avoids aliasing.

3.3. Framework for DoA Estimation. Here, we first propose a sample selection approach for DoA estimation by automatic phase deviation detection. Then, we formulate a DoA estimation framework exploiting preprocessing techniques and model order selection schemes.

Figure 3 depicts the flowchart of the proposed signal processing solution for DoA estimation.

As shown in Box 2 of the Figure 3, the phase deviation correction proposed in Section 3.1 returns a matrix $\mathbf{X}_{\mathrm{c}}$ that is used in the sample selection scheme in Section 3.3.1.

3.3.1. Sample Selection for DoA Estimation by Automatic Phase Deviation Detection. As exemplified $\mathbf{X}_{\mathrm{c}}$ in Figure 4, empirically we observed that the hardware causes phase deviations on the samples in random time instants. Therefore, we propose an approach to select the samples with phase deviations for the DoA estimation.

Note that the phase compensation proposed in Section 3.1 has been applied to the samples, whose matrix $\mathbf{\Phi}$ containing the phases are depicted in Figure 4. Furthermore, note that there are significant deviations that can degrade the DoA estimation process. The main objective here is to remove these phase deviations.

As shown in Figure 5, such ripples can be better visualized by computing the phase difference in the time dimension according to the following expression:

$$\gamma(m,i) = \left(\phi(m,i+1) - \phi(m,i)\right)^2, \qquad (11)$$

where $\gamma(m,i)$ is the value containing the quadratic difference of two consecutive time samples i and $i+1$ of the m-th channel. The $\phi(m,i)$, from (6), is the element in position (m,i) of the matrix $\mathbf{\Phi}$.

Figure 5 draws the $\mathbf{\Gamma}$ that contains the result of (11). By detecting the peaks, we can identify which samples should be removed. For this task, we apply the approach proposed in [37], which returns the green curve with the value of the threshold. Therefore, the samples whose phase differences are greater than the threshold are removed. The result after the samples removed is presented in the following equation:

$$\begin{aligned}\mathbf{X}_{\mathrm{cs}} = [&\mathbf{X}_{\mathrm{c}}(:,1:N_1) \mid \mathbf{X}_{\mathrm{c}}(:,N_2:N_3) \mid \cdots \mid \\ &\mathbf{X}_{\mathrm{c}}(:,N_{T-1}:N_T)],\end{aligned} \qquad (12)$$

where $\mathbf{X}_{\mathrm{cs}}$ is the matrix with the selected samples. Note that $N_T = N$ and the values of N_t, for $t = 1,\ldots,T$, are found by comparing the phase difference values with the threshold in Figure 5.

3.3.2. DoA Estimation Framework. According to Figure 3, the matrix $\mathbf{X}_{\mathrm{cs}}$ given in (12) is used to improve the DoA estimation with preprocessing schemes. There are several

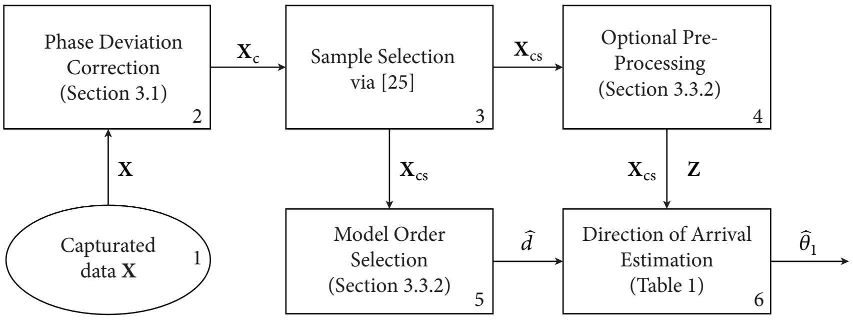

Phase Deviation Correction (Section 3.1)
2
$\mathbf{X}_c$
Sample Selection via [25]
3
$\mathbf{X}_{cs}$
Optional Pre-Processing (Section 3.3.2)
4
$\mathbf{X}$
$\mathbf{X}_{cs}$
$\mathbf{X}_{cs}$
$\mathbf{Z}$
Capturated data $\mathbf{X}$
1
Model Order Selection (Section 3.3.2)
5
$\hat{d}$
Direction of Arrival Estimation (Table 1)
6
$\hat{\theta}_1$

FIGURE 3: Flowchart of the proposed solution for DoA estimation.

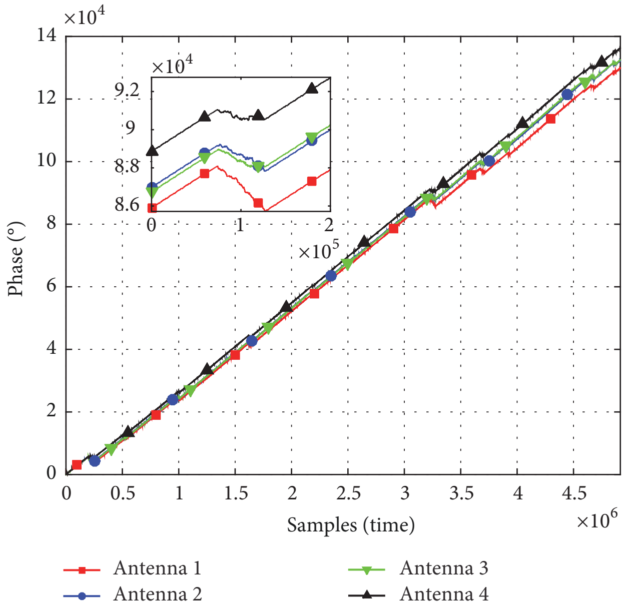


FIGURE 4: Phase of the received data in each antenna. An example of phase deviation region is highlighted in the zoom area.

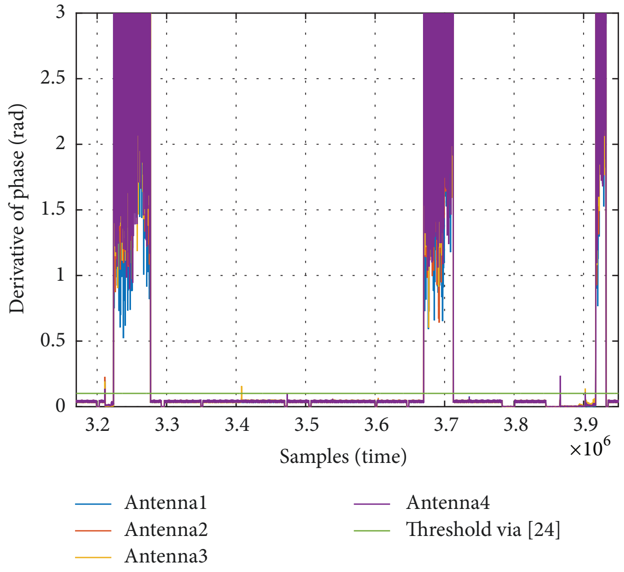


FIGURE 5: Difference between consecutive samples from Figure 4.

DoA estimation schemes in the literature, such as beamforming approaches and subspace-based approaches. Examples of beamforming are Delay and Sum [38] and Capon [39], whereas examples of classical subspace-based approaches are MUSIC [23] and ESPRIT [40].

The DoA estimation schemes assume that the model order d is known. In practice, model order selection techniques should be applied to estimate the model order d, as depicted in Figure 3.

In the flowchart of Figure 3, we adopted the Exponential Fitting Test (EFT) [43, 44] as the model order selection scheme. The EFT has the deflation property that allows us to find suitable thresholds as a function of the Probability of False Alarm (P_{fa}). By exploiting the deflation property and by finding suitable thresholds, the EFT has been the only scheme in the literature to estimate $d = 1$ in the presence of a strong LoS signal and $d = 0$ in the only noise (no signal) measurements. We have compared several schemes in the literature such as Akaike Information Criterion (AIC) [45], Efficient Detection Criterion (EDC) [46], Minimum Description Length (MDL) [47], Stein's unbiased risk estimate (SURE) [48], RADOI [49], ESTimation ERror (ESTER) [50], and Subspace-based Automatic Model Order Selection (SAMOS) [51]. The M-EFT [43, 44, 52] has also been suitable, but an even smaller P_{fa} was required to find the thresholds. The computation of the thresholds of the EFT requires an extremely low P_{fa}. The complexity of such a computation is prohibitive. Therefore, we propose in Appendix A an extrapolation algorithm to compute such thresholds. Note that our proposed extrapolation algorithm has been applied in [53–56], although no details are provided. The reason for extremely low P_{fa} may be related to the noise behaviour as shown in Appendix B. Note that the Ilmenau Package for Model Order Selection (IPM) [57] with MATLAB and Java implementation of the model order selection schemes can be found at the LASP homepage (https://lasp.unb.br/index.php/publications/softwares/).

In order to further improve the accuracy of DoA estimation schemes, preprocessing schemes can be applied beforehand. We consider in this work the Vandermonde Invariation Technique (VIT) [58], Spatial smoothing (SPS) [59], and Forward Backward Averaging (FBA) [60, 61] as a preprocessing schemes. As depicted in Figure 3, after the

Table 1: Selected state-of-the-art DoA estimation schemes.

Delay And Sum [38]	$P_{\mathrm{DS}}(\theta) = \dfrac{\mathbf{w}(\theta)^{\mathbf{H}}\mathbf{R}_{\mathbf{xx}}\mathbf{w}(\theta)}{\mathbf{w}(\theta)^{\mathbf{H}}\mathbf{w}(\theta)}.$	(13)
Capon [39]	$P_{\mathrm{CAP}}(\theta) = \dfrac{\mathbf{1}}{\mathbf{w}(\theta)^{\mathrm{H}}\mathbf{R}_{\mathbf{xx}}^{-1}\mathbf{w}(\theta)}.$	(14)
MUSIC [23]	$P_{MUSIC}(\theta) = \dfrac{\mathbf{1}}{\mathbf{w}^{\mathrm{H}}(\theta)\mathbf{V}_{\mathbf{n}}\mathbf{V}_{\mathbf{n}}^{\mathrm{H}}\mathbf{w}(\theta)},$	(15)
ESPRIT [40]	$\mathbf{\Psi} = \mathbf{J}_1\mathbf{U}_{\mathrm{s}}^{+}\mathbf{J}_2\mathbf{U}_{\mathrm{s}}$	(16)
	$\mathbf{\Psi} \overset{\mathrm{EVD}}{=} \mathbf{E}\Phi\mathbf{E}^{\mathrm{H}},$ with $\mathbf{\Phi} = \mathrm{diag}[\phi_1, \dots, \phi_d]$	(17)
	$\mu_i = \angle(\phi_i),\ i = 1, \dots, d$	(18)

preprocessing step, a matrix $\mathbf{Z}$ is returned and used by the DoA methods summarized in Table 1.

In Table 1, the vector $\mathbf{w}(\theta)$ in (13), (14), and (15) vary according to the candidate values of θ. The value of θ that maximizes the expression in (13), (14), and (15) is the $\widehat{\theta}_1$, since, in Section 2, the data model assumes that the LoS component faces no obstacles. Therefore, the component corresponding to the greatest power should be the same component with DoA equal to $\widehat{\theta}_1$. $\mathbf{U}_{\mathrm{s}} \in \mathbb{C}^{M\times d}$ is signal subspace, which is equal to the d eigenvectors corresponding to the d greatest eigenvalues, whereas $\mathbf{V}_{\mathrm{n}} \in \mathbb{C}^{M\times M-d}$ is a basis for the noise subspace, composed by the $M-d$ eigenvectors associated with to the $M-d$ smallest eigenvalues. In (17) and (18), $\mathbf{\Phi}$ is the diagonal matrix that has the eigenvalues of $\mathbf{\Psi}$. We compute all the spatial frequencies and the one whose component has the greatest power is the $\widehat{\mu}_1$ that can be mapped to $\widehat{\theta}_1$.

3.4. Tensor-Based DoA Estimation. In this subsection, a tensor factorization, namely, the PARAFAC decomposition, is applied. The PARAFAC decomposition generates three factor matrices from a received tensor $\mathcal{X}$, whose structure is detailed in this subsection. One factor matrix corresponds to the estimate of the steering matrix $\mathbf{A}$ containing DoA information. In this subsection $\mathbf{A}$ is extracted from tensor $\mathcal{X}$ to estimate the DoA of the impinging signal.

We first consider an unchanging sequence of N symbols transmitted periodically. Such symbols can be found in a header or footer or even in the payload of a message. Alternatively, repeating sequences of symbols can be extracted from time periods when no data is being transmitted, but the carrier of the transmitter is active. An example is given as follows. Since the oscillators at the transmitter and at the receiver are never exactly the same, a small frequency deviation or constant phase change is observed at the receiver. At an MSK receiver, if the deviation is positive, a sequence of ones can be extracted and if the deviation is negative a sequence of zeros is observed.

The symbols corresponding to the sequence from the i-th source is represented by the vector $\mathbf{c}_i \in \mathbb{C}^{N\times 1}$. Accordingly, we can build a received signal matrix

$$\mathbf{X}_i(p) = \gamma_i(p)\,\mathbf{a}_i(\mu_i)\,\mathbf{c}_i^{\mathrm{T}}, \in \mathbb{C}^{M\times N} \qquad (19)$$

where p is the period corresponding to the sequence transmission. Generalizing for d signals we find

$$\mathbf{X}(p) = \mathbf{A}\mathbf{D}_{\gamma}(p)\,\mathbf{C}^{\mathrm{T}}, \in \mathbb{C}^{M\times N}, \qquad (20)$$

where $\mathbf{D}_{\gamma}(p) = \mathrm{diag}([\gamma_1(p)\ \ \gamma_2(p)\ \ \dots\ \ \gamma_d(p)])^{\mathrm{T}}$ and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \dots\ \mathbf{c}_d]$.

For P transmitted sequences, the symbols can be concatenated along the third dimension to form the received signal tensor

$$\mathcal{X} = [\mathbf{X}(1) \sqcup_3 \mathbf{X}(2) \sqcup_3 \dots \sqcup_3 \mathbf{X}(P)] \in \mathbb{C}^{M\times N\times P}, \qquad (21)$$

where $\sqcup_3$ represents concatenation along the third dimension. Since the slices of $\mathcal{X}$ can be written as (20), $\mathcal{X}$ has a PARAFAC structure and can be decomposed into three factor matrices $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{\Gamma}$, where $\mathbf{\Gamma}$ contains the diagonals of $\mathbf{D}_{\gamma}(p)$ along its rows. To factorize $\mathcal{X}$, we first rewrite it in three different matrix representations or unfoldings

$$\mathcal{X}_{(1)} = \mathbf{A}(\mathbf{C} \diamond \mathbf{\Gamma})^{\mathrm{T}}, \qquad (22)$$

$$\mathcal{X}_{(2)} = \mathbf{C}(\mathbf{\Gamma} \diamond \mathbf{A})^{\mathrm{T}}, \qquad (23)$$

$$\mathcal{X}_{(3)} = \mathbf{\Gamma}(\mathbf{A} \diamond \mathbf{C})^{\mathrm{T}}, \qquad (24)$$

where $\diamond$ is the Khatri-Rao (column-wise Kroenecker) product.

It is known that minimizing (22), (23), and (24) in the least squares sense [62] leads to the following solutions:

$$\widehat{\mathbf{A}} = \mathcal{X}_{(1)}\left[(\mathbf{C} \diamond \mathbf{\Gamma})^{\mathrm{T}}\right]^{+}, \qquad (25)$$

$$\widehat{\mathbf{\Gamma}} = \mathcal{X}_{(2)}\left[(\mathbf{\Gamma} \diamond \mathbf{A})^{\mathrm{T}}\right]^{+}, \qquad (26)$$

$$\widehat{\mathbf{C}} = \mathcal{X}_{(3)}\left[(\mathbf{A} \diamond \mathbf{C})^{\mathrm{T}}\right]^{+}. \qquad (27)$$

We consider the well-known Alternating Least Squares (ALS) algorithm to solve (25), (26), and (27) in an iterative way. Since $\mathbf{C}$ is known, the estimation step (27) is skipped, and the ALS algorithm alternates between the estimations of $\mathbf{A}$ and $\mathbf{\Gamma}$ in a two-step approach [63].

Once convergence is achieved, we use $\widehat{\mathbf{A}}$ to extract the DoA of the i-th source as follows:

$$\widehat{\theta}_i = \underset{\theta}{\mathrm{argmax}}\ \mathbf{a}_i^{\mathrm{H}}(\theta)\,\widehat{\mathbf{a}}_i \qquad (28)$$

Table 2: Number of frames captured at each 10° step of the measurement campaign.

DoA	−90°	−80°	−70°	−60°	−50°	−40°	−30°	−20°	−10°	0°	10°	20°	30°	40°	50°	60°	70°	80°	90°
Frames	31	29	29	30	33	34	32	29	24	30	32	24	30	33	26	31	43	41	31

Table 3: RMSE for the schemes in Table 1 without the preprocessing using the measurements from Figure 8.

Algorithm	DS	CAP.	MUS.	ESP.	TEN.
RMSE	2.3°	1.7°	4.0°	2.8°	1.7°
Variance of the RMSE	4,3°	1.1°	11.5°	4.1°	4.4°

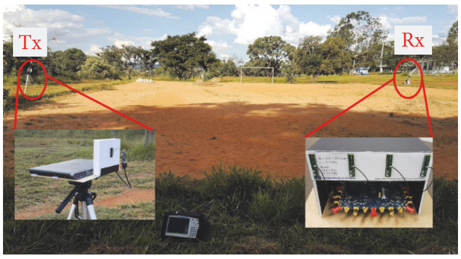


Figure 6: Photo taken during the measurement campaigns of the outdoor scenario pointing out the positions of the transmitter and of the antenna array based receiver.

4. Experiments

In this section, we validate our proposed drone tracking device with measurement campaigns in an outdoor scenario. In Section 4.1, the setup for the measurement campaign is described, whereas, in Section 4.2, we present the obtained results.

4.1. Experimental Setup. In Figure 6, we depict the outdoor scenario used for the measurement campaigns. On the right-hand, we placed our drone tracking device, proposed in Section 3, as the receiver, whereas, on the left-hand, the transmitter is placed. The transmitter is a 2x2 MIMO SDR platform ASPR4 [64], with frequencies ranging from 50 MHz to 6.0 GHz, a channel bandwidth varying from 200 kHz to 56 MHz and a maximum power of 10 dBm at the output port.

As shown in Figure 7, the distance between the transmitter and the receiver is 48 m. Both transmitter and receiver are placed on tripods 115 cm above the ground. Note that the red "X" in Figure 7 is the location from where the photo in Figure 6 has been taken.

Both the transmitter and the receiver were set up using a MSK message signal, at 2.48 GHz carrier frequency and 250 kbps of data rate. Before starting the experiment, the 2.48 GHz frequency was scanned and no noise source was detected. In order to verify that the receiver properly works and measures the Bit Error Rate (BER), we have to decode the signal. To this purpose, the transmitted package must be known and consists of pseudo random sequences of length 1024 bits and a header and footer with 16 bits each defined as 0xFFFF and 0x0000, respectively. Therefore, the total size of the package is 1056 bits. The transmitter uses both sampling frequency and bandwidth of 2MHz. At the receiver side, a 4 MHz sampling frequency and 4 MHz bandwidth are used. Each captured frame at the receiver has 5120 samples.

As shown in Figure 7, the transmitter is fixed and the receiver rotates from +90° to −90° in steps of 10°. According to Table 2, which presents the number of captured frame per DoA, at each 10° step, about 31 frames of size 5120 samples are captured.

4.2. Experimental Results. This subsection shows the performance of the DoA estimation schemes shown in Table 1. During the measurement campaign, the achieved Bit Error Rate (BER) was 10^{-4}.

Figure 8 shows the DoA calculated by rotating the base array over the time. Our proposed device works for a DoA ranging from −60° to 60°. The DS, Capon, MUSIC, ESPRIT, and Tensor combined with preprocessing schemes are shown in Figure 9(b).

In Tables 3 and 4, we present the Root Mean Square Error RMSE for the schemes in Table 1 with and without preprocessing schemes using the measurements from Figure 6, respectively. The equation for RMSE is given by

$$\mathrm{RMSE}(\theta_1) = \sqrt{\frac{1}{Q}\sum_{q=1}^{Q}\left(\theta_1^{(q)} - \hat{\theta}_1^{(q)}\right)^2} \tag{29}$$

where q is one realization of a total of Q realizations for each 10° stage of the measurement campaign. In other words, the acquired data at each 10° degree step is reshaped into small matrices. The computation of the DoA is individually performed for each of these matrices. In our experiment, we chose empirically matrices of size 4×1000. Since, as shown in Table 2, at each 10° step 31 frames of size 5120 samples are captured, in average there are approximately $Q = 158$ matrices depending on the data reduction performed in Section 3.3.1. The variables θ and $\hat{\theta}$ stand for the actual and the estimated DoA, respectively.

Comparing Tables 3 and 4 we can note that, except for the Tensor, the algorithms presented improvement in terms of RMSE after incorporating the preprocessing. CAPON and the Tensor increased the variance with the preprocessing. The smallest RMSE was achieved by the DS approach after preprocessing. Note that the ESPRIT assumes the shift invariance property, while both MUSIC and ESPRIT exploit the property of the orthogonality between signal and noise subspaces. Note that both assumptions are approximations and, therefore, their exploitation may cause additional errors.

Table 4: RMSE for the schemes in Table 1 with the preprocessing using the measurements from Figure 8.

Algorithm	DS	CAP.	MUS.	ESP.	TEN.
RMSE	1,6°	1,7°	1,7°	1,9°	2,7°
Variance of the RMSE	2,3°	2,5°	2,4°	1,8°	23,7°

Table 5: Comparison between the obtained results and the DoA estimation in practical measurements results already available in the literature.

	Drone detect solutions			General DoA solutions				
Solutions	Proposed	[25]	[27]	[20]	[21]	[22]	[24]	[15]
DoA precision	1.9°	1°	1-3°	5°	1°	4°	0.5-2°	2°
Uses anechoic chamber	No	x	x	No	Yes	Yes	Yes	Yes
Low cost	Yes	No	No	Yes	Yes	Yes	Yes	Yes

In Table 5, we compare the DoA estimation results obtained with our framework and with the state-of-the-art approaches. Furthermore, we show that, even without involving anechoic chamber in our tests, precise results are obtained in comparison to references that implemented its tests in a nonreflexive environment. The ‘x’ means that the commercial solutions did not provide such information. Finally, we provide the information about which of the solutions has low cost.

As shown in Table 6, the total cost of the proposed drone tracking solution is US$ 2.222, whereas the solutions in [25] and [27] cost US$ 226.000 and US$ 120.000, respectively. Therefore, our proposed off-the-shelf solution costs less than 2 % of the commercial solutions in [25, 27].

5. Conclusions

In this paper, we have proposed a low cost antenna array based drone tracking device for outdoor environments. The proposed solution is divided into hardware and software parts. The hardware part of the proposed device is based on off-the-shelf components such as an omnidirectional antenna array, a 4-channel SDR platform with carrier frequency ranging from 70 MHz to 6 GHz, a FPGA motherboard, and a laptop. The software part includes algorithms for calibration, model order selection (MOS), and DoA estimation, including specific preprocessing steps to increase the DoA accuracy. We have evaluated the performance of our proposed low cost solution in outdoor scenarios. Our measurement campaigns have shown that when the array is in the front fire position, i.e., with a DoA ranging from −60° to 60°, the maximum and the average DoA errors are 6° and 1,9°, respectively. Our proposed off-the-shelf solution costs less than 2 % of commercial solutions in [25, 27]. In order to further improve our analysis of the proposed system and our results, experiments in an anechoic chamber can be performed. Moreover, the performance of the proposed framework can be improved by incorporating interpolation schemes. Perspectives also include the adoption of a more realistic noise model to simplify the computation of the thresholds of the Exponential Fitting Test (EFT).

Appendix

A. Proposed Extrapolation Algorithm to Find the EFT Thresholds for Extremely Low Probability of False Alarm

In this appendix we propose an extrapolation algorithm to estimate the thresholds of the EFT algorithm in cases that the Probability of False Alarm (P_{fa}) is extremely low.

The EFT is based on the approximation that the profile of the ordered noise eigenvalues has an exponential behaviour. The profile $a(M, N)$ can be expressed as

$$a(M,N) = \sqrt{\frac{1}{2}\left(\frac{15}{M^2+2} - \sqrt{\frac{225}{(M^2+2)^2} - \frac{180M}{N(M^2-1)(M^2+2)}}\right)}. \tag{A.1}$$

Given that $\hat{d} = M - P^*$, our goal is to vary P such that we find P^* that $\hat{\lambda}_{M-P} \ll \lambda_{M-P}$, where $\hat{\lambda}_{M-P}$ and λ_{M-P} stand for predicted $(M-P)^{\text{th}}$ noise eigenvalue and $M - P$ stands for actual eigenvalue, respectively. Note that the EFT assumes that smallest eigenvalue is a noise eigenvalue. Therefore, P varies from 1 to $M - 1$. Using (A.1), [44] has derived the following expression:

$$\hat{\lambda}_{M-P} = (P+1)\cdot\left(\frac{1-a(P+1,N)}{1-a(P+1,N)^{P+1}}\right)\cdot\hat{\sigma}^2, \tag{A.2}$$

$$\hat{\sigma}^2 = \frac{1}{P+1}\sum_{i=0}^{N}\lambda_{M-i}, \tag{A.3}$$

where $\hat{\sigma}^2$ is the estimated noise power.

In order to improve further the performance of the EFT approach, thresholds coefficients η_P are computed using noise-only simulated data following Complex-Valued Circularly Symmetric Gaussian and identically and independently

Table 6: Price table for the hardware of the proposed drone tracker device.

Hardware	Cost (US$)
AD-FMCOMMS5-EBZ-ND [33]	1080
4 × Dual-Band Antenna 2.4 & 5GHz [36]	11
ZYNQ 7000 Zc702 [35]	999
Power Divider 2.4-6GHz 30Watts RoHS IP67 [41]	109
5 × cable 305mm HPP100 SMA [42]	23
Total	2222

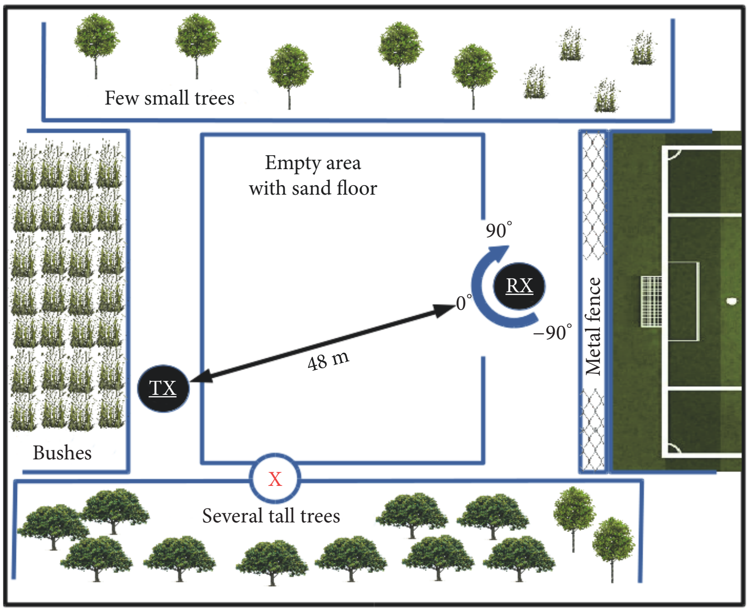


Figure 7: Top view of the outdoor scenario for the measurement campaigns including the positions of the transmitter and of the antenna array based receiver.

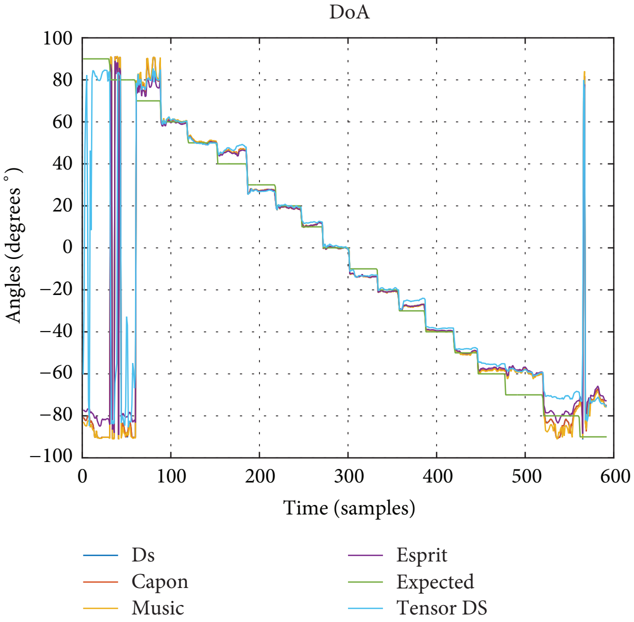


Figure 8: Comparison between the DoA estimation schemes by varying the DoA from +60° to −60° with steps of 10°.

distributed (i.i.d.) as indicated in Section 2. Depending on the η_P, we have two hypothesis:

$$\mathrm{H}_P : \left| \frac{\lambda_{M-P} - \hat{\lambda}_{M-P}}{\hat{\lambda}_{M-P}} \right| \leq \eta_P \qquad \text{(A.4)}$$

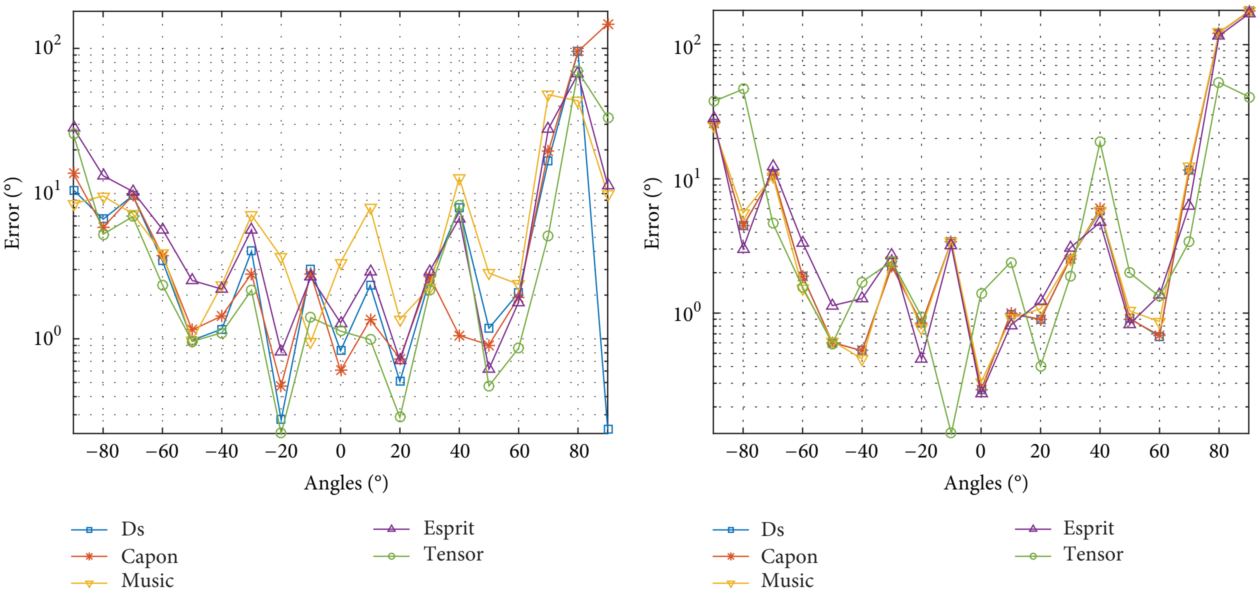
$$\overline{\mathrm{H}}_P : \left| \frac{\lambda_{M-P} - \hat{\lambda}_{M-P}}{\hat{\lambda}_{M-P}} \right| > \eta_P \qquad \text{(A.5)}$$

where $\mathrm{H}_P : \lambda_{M-P}$ is a noise eigenvalue and $\overline{\mathrm{H}}_P : \lambda_{M-P}$ is a signal eigenvalue. In order to have all η_P depending of the P_{fa}, we can define the P_{fa} as

$$\mathrm{P}_{\mathrm{fa}} = \Pr\left[\hat{\mathrm{d}} \neq 0 \mid \mathrm{d} = 0\right]. \qquad \text{(A.6)}$$

Note that the η_P thresholds are obtained by Monte Carlo simulations carried out in the only-noise scenario following the steps in [44] and by choosing the following amount of realizations:

$$\mathrm{I} = \frac{10}{\mathrm{P}_{\mathrm{fa}}}. \qquad \text{(A.7)}$$



(a) Without preprocessing

(b) With preprocessing

FIGURE 9: Comparison of the DoA estimation error between the DoA estimation schemes from Table 1 by varying the angle from −90° to 90° with steps of 10°.

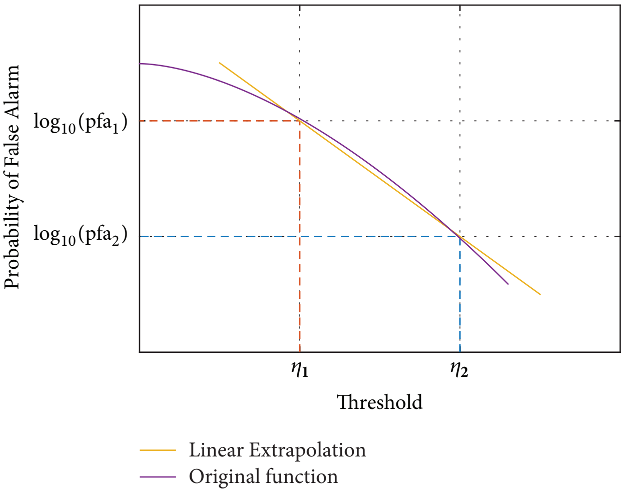


FIGURE 10: Log linear extrapolation based on two points given by P_{fa}s and thresholds η_P.

Depending on the noise behaviour and the parameters of the scenario [54, 56], the thresholds can be extremely low. Therefore, the computational complexity of (A.7) can be prohibitive. In order to overcome such limitation, we propose an extrapolation approach to compute the thresholds for extremely low values of P_{fa}.

Since we wish to estimate values outside the known limits, we can use an extrapolation method and approximate the descending side of the curve as a decreasing exponential. In order to simplify the approximation, we adopt a logarithmic scale as exemplified in Figure 10.

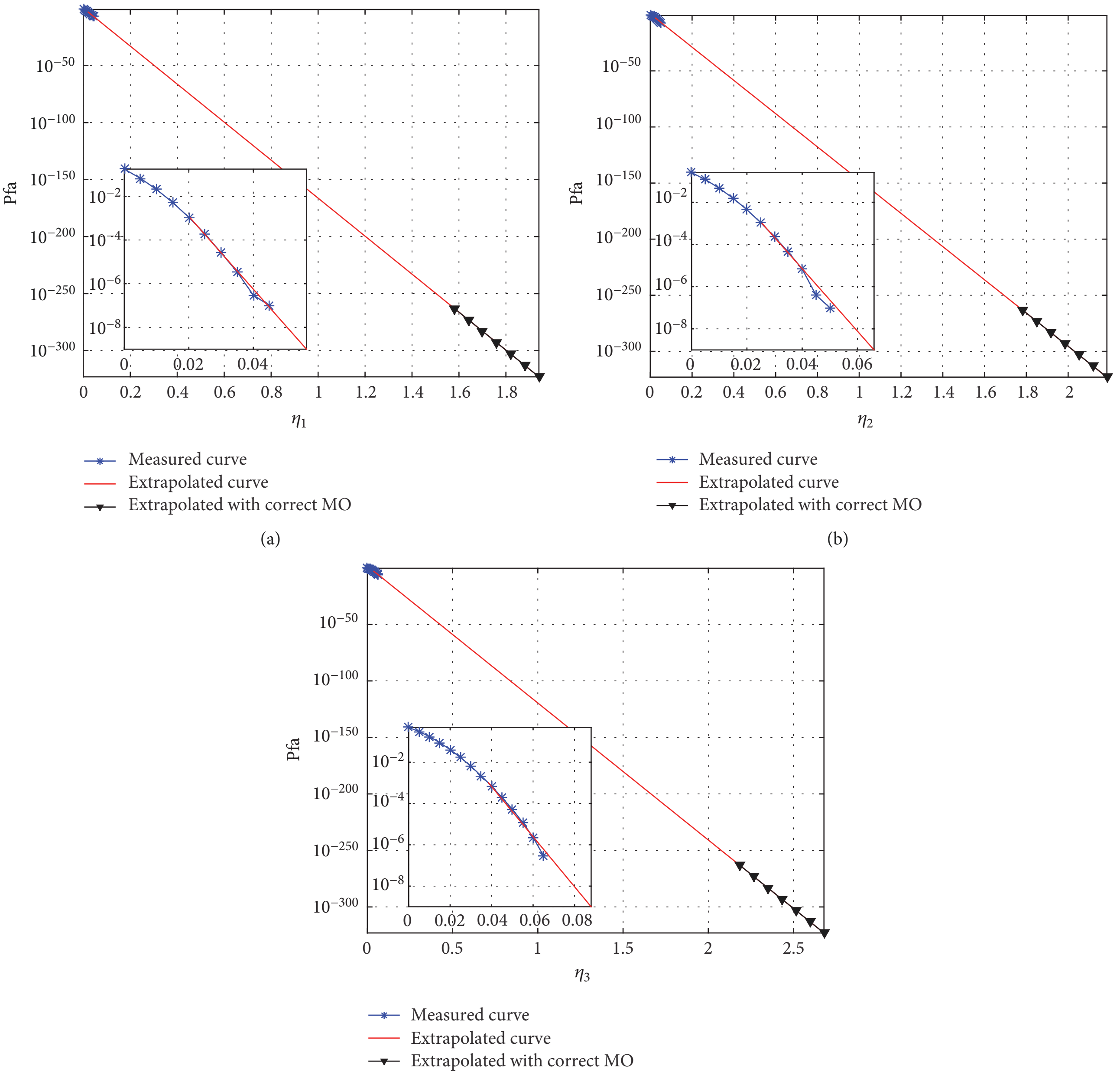
Given the two known points in Figure 10 obtained by Monte Carlo simulations and given the linear extrapolation in (A.8), we can compute the two unknown constants a and b.

$$P_{\mathrm{fa}} = a \cdot \eta_{\mathrm{P}} + b. \tag{A.8}$$

The constants a and b are given by (A.9) and (A.10) by using the two known points $(\log_{10}(P_{\mathrm{fa}1}), \eta_1)$ and $(\log_{10}(P_{\mathrm{fa}2}), \eta_2)$.

$$a = \frac{\log_{10}(P_{\mathrm{fa}2}/P_{\mathrm{fa}1})}{(\eta_2 - \eta_1)} \tag{A.9}$$

$$b = \log_{10}(P_{\mathrm{fa}2}) - \eta_2 \cdot \left(\frac{\log_{10}(P_{\mathrm{fa}2}/P_{\mathrm{fa}1})}{(\eta_2 - \eta_1)}\right) \tag{A.10}$$



(a)

(b)

(c)

FIGURE 11: Thresholds for extrapolated data using computation with M=4 and N=41.

By replacing a and b in (A.8), we obtain the expression for the $\widehat{\eta}_P$ in (A.11).

$$\eta_P = \frac{\log_{10}(\mathrm{P_{fa}}) - (\log_{10}(\mathrm{P_{fa2}}) - (\eta_2/(\eta_2-\eta_1)) \cdot \log_{10}(\mathrm{P_{fa2}}/\mathrm{P_{fa1}}))}{\log_{10}(\mathrm{Pfa}_2/\mathrm{Pfa}_1)/(\eta_2-\eta_1)}. \tag{A.11}$$

Following the framework of Figure 3, we set up the EFT with a $\mathrm{P_{fa}} = 10^{-263}$ and we obtained the following values for the thresholds: $\eta_1 = 1.5810$, $\eta_2 = 1.7810$, and $\eta_3 = 2.1840$.

Note that there are only three thresholds, since the smallest eigenvalue is assumed as a noise eigenvalue in the EFT approach. In Figures 11(a), 11(b), and 11(c), we depict

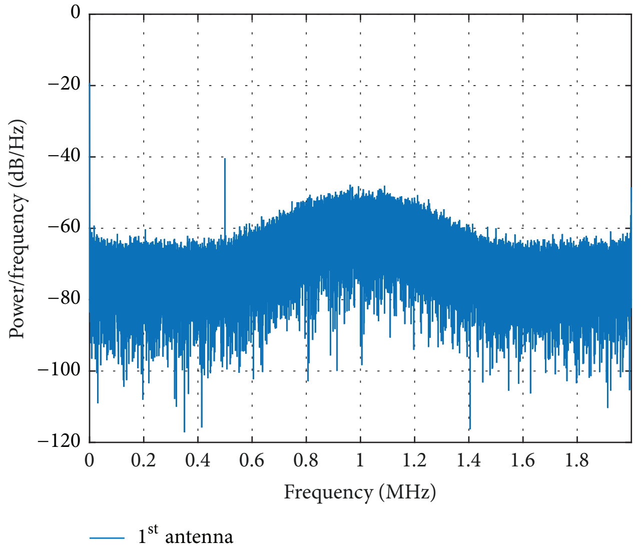


FIGURE 12: Power spectral density estimate of the only-noise samples captured by the 1st antenna of the antenna array.

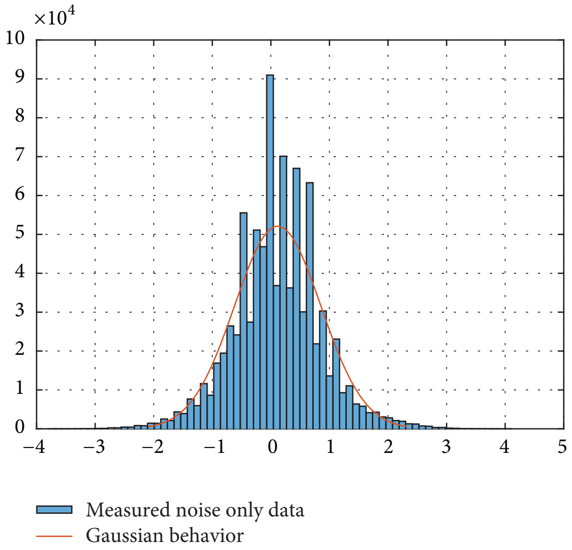


FIGURE 13: Histogram of 3rd antenna with 70 bins, estimated mean = 0.1162, and estimated standard deviation = 0.7377.

the extrapolation curves for the thresholds η_1, η_2, and η_3, respectively.

B. Noise Analysis

In Section 2, the noise is assumed to be Complex-Valued Circularly Symmetric Gaussian and identically and independently distributed (i.i.d.). The EFT relies on these properties of the noise. Due to extremely low values of the P_{fa} in Appendix A, we analyze the noise behaviour.

According to Figure 12, the Power Spectrum Density (PSD) is not flat, indicating that the noise is time correlated.

In Figure 13, we depict the normalized histogram for antenna 3. Note that the Gaussian approximation has errors that can be reduced with an improved model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the colleagues Prof. Dr. Sébastien Roland Marie Joseph Rondineau, Dr.-Ing. Christopher Schirmer, and Mr. Mario Lorenz for their support. This work was partially supported by Research Support Foundation of the Brazilian Federal District (FAPDF), by the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES), and by the Brazilian National Council for Scientific and Technological Development (CNPq).

References

[1] R. K. Miranda, J. P. C. L. Da Costa, and F. Antreich, "High accuracy and low complexity adaptive Generalized Sidelobe Cancelers for colored noise scenarios," *Digital Signal Processing*, vol. 34, pp. 48–55, 2014.

[2] R. Kehrle Miranda, J. P. C. L. da Costa, F. Roemer, L. R. A. X. Menezes, G. Del Galdo, and A. L. F. de Almeida, "Low complexity performance assessment of a sensor array via unscented transformation," *Digital Signal Processing*, vol. 63, pp. 190–198, 2017.

[3] D. Neudert-Schulz, *A contribution to efficient direction finding using antenna arrays, [Ph.D. dissertation]*, Technische Universitat Ilmenau, 2017.

[4] M. A. M. Marinho, F. Antreich, and J. P. Carvalho Lustosa Da Costa, "Improved array interpolation for reduced bias in DOA estimation for GNSS," in *Proceedings of the of Institute of Navigation (ION) GNSS+*, 2014.

[5] D. V. de Lima, J. P. da Costa, J. P. Maranhao, and R. T. de Sousa, "High resolution Time-delay estimation via direction of arrival estimation and Khatri-Rao factorization for multipath mitigation," in *Proceedings of the 21st International ITG Workshop on Smart Antennas (WSA 2017)*, pp. 388–395, Berlin, Germany, 2017.

[6] D. V. de Lima, J. P. C. L. da Costa, J. P. A. Maranhão, and R. T. de Sousa, "Time-delay estimation via procrustes estimation and Khatri-Rao factorization for GNSS multipath mitigation," in *Proceedings of the 2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–7, IEEE, 2017.

[7] D. V. De Lima, J. P. C. L. Da Costa, F. Antreich, R. K. Miranda, and G. Del Galdo, "Time-Delay estimation via CPD-GEVD applied to tensor-based GNSS arrays with errors," in *Proceedings of the 7th International Workshop on Computational Advances*

in Multi-Sensor Adaptive Processing (CAMSAP), pp. 1–5, IEEE, 2017.

[8] F. Karimi, "Hundreds of drones fly dangerously close to manned aircraft," https://edition.cnn.com/2015/12/12/us/drone-aircraft-close-calls, 2015.

[9] K. Rawlinson, "Drone hits plane at Heathrow airport, says pilot," https://www.theguardian.com/uk-news/2016/apr/17/drone-plane-heathrow-airport-british-airways, 2016.

[10] Z. Alkhalisi, "Dubai deploys a 'drone hunter' to keep its airport open," http://money.cnn.com/2016/11/04/technology/dubai-air-port-drone-hunter/index.html?iid=EL , 2016.

[11] J. Goglia, "A small drone hits a commercial airliner and nothing happens," https://www.forbes.com/sites/johngoglia/2017/10/19/a-small-drone-hits-a-commercial-airliner-and-nothing-hap-pens/#78ad9a249ea1, 2017.

[12] H. Krim and M. Viberg, "Two decades of array signal processing research," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[13] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Transactions on Signal Processing*, vol. 36, no. 10, pp. 1553–1560, 1988.

[14] R. Muhamed and T. Rappaport, "Comparison of conventional subspace based DOA estimation algorithms with those employing property-restoral techniques: simulation and measurements," in *Proceedings of the ICUPC - 5th International Conference on Universal Personal Communications*, vol. 2, pp. 1004–1008, 1996.

[15] M. Tarkowski and L. Kulas, "RSS-based DoA estimation for ESPAR antennas using support vector machine," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 4, pp. 561–565, 2019.

[16] T. Nowak, M. Hartmann, J. Thielecke, N. Hadaschik, and C. Mutschler, "Super-resolution in RSS-based direction-of-arrival estimation," in *Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, IEEE, 2018.

[17] K. Honda, D. Iwamoto, and K. Ogawa, "Angle of arrival estimation embedded in a circular phased array 4 × 4 MIMO antenna," in *Proceedings of the 2017 IEEE Asia Pacific Microwave Conference, APMC 2017*, pp. 93–96, IEEE, 2017.

[18] L. Kulas, "RSS-based DoA estimation using ESPAR antennas and interpolated radiation patterns," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 1, pp. 25–28, 2018.

[19] Q. Liang, B. Sun, and G. Zhou, "Multiple Beam Parasitic Array Radiator Antenna for 2.4 GHz WLAN Applications," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 12, pp. 2513–2516, 2018.

[20] C. H. Meng, H. H. Cheng, J. L. Wen, C. Y. Chia, H. Y. Ting, and C. L. Hsin, "Direction-of-arrival estimator using array switching on software defined radio platform," in *Proceedings of the International Symposium on Antennas and Propagation (APSURSI)*, pp. 2821–2824, IEEE, 2011.

[21] V. Y. Vu and A. B. Delai, "Digital solution for inter-vehicle localization system by means of direction-of-arrival," in *Proceedings of the International Symposium on Intelligent Signal Processing and Communications (ISPACS)*, pp. 875–878, 2006.

[22] S. Björklund and A. Heydarkhan, "High resolution direction of arrival estimation methods applied to measurements from a digital array antenna," in *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM '00)*, pp. 464–468, 2000.

[23] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[24] V. Y. Vu, A. J. Braga, X. Begaud, and B. Huyart, "Direction of arrival and time of arrival measurements using five-port reflectometers and quasi-Yagi antennas," in *Proceedings of the 11th International Symposium on Antenna Technology and Applied Electromagnetics, ANTEM 2005*, pp. 1–4, 2005.

[25] "Robin Radar Systems BV, Elvira® Drone Detection Radar," https://www.robinradar.com/markets/drone-detection/.

[26] "Dronelabs LLC, DD610AR Stationary Drone Detector," http://dronedetector.com/stationary-unit/.

[27] "Real-Time RF Drone/UAV and Radar Detection System," http://www.aaronia.com/products/solutions/Aaronia-Drone-Detection-System/.

[28] "Detect, Inc., Drone Watcher," http://www.dronewatcher.com/.

[29] R. K. Miranda, D. A. Ando, J. P. da Costa, and M. T. de Oliveira, "Enhanced direction of arrival estimation via received signal strength of directional antennas," in *Proceedings of the 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 162–167, 2018.

[30] D. A. Ando, R. K. Miranda, J. P. da Costa, and M. T. de Oliveira, "A novel direction of arrival estimation algorithm via received signal strength of directional antennas," in *Proceedings of the 2018 Workshop on Communication Networks and Power Systems (WCNPS)*, pp. 1–5, 2018.

[31] S. Basak and B. Scheers, "Passive radio system for real-time drone detection and DoA estimation," in *Proceedings of the 2018 International Conference on Military Communications and Information Systems, ICMCIS 2018*, pp. 1–6, 2018.

[32] "A. Devices. FMComms5 Plugin Description," https://wiki.analog.com/resources/tools-software/linux-software/fmcomms5_plugin, 2016.

[33] "Eval board software Defined Radio ADFMCOMMS5-EBZ-ND," http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad-fmcomms5-ebz.html.

[34] "A. DEVICES. (2017) AD9361 datasheet," http://www.analog.com/media/en/technical-documentation/data-sheets/AD9361.pdf.

[35] "ZYNQ-7000 ZC702 Eval KIT," https://www.xilinx.com/products/boards-and-kits/ek-z7-zc702-g.html.

[36] "Dual band WiFi, 2.4 & 5GHz, PCB," http://www.te.com/usa-en/product-2118309-1.html.

[37] P. F. C. Lima, R. K. Miranda, J. P. C. L. Da Costa, R. Zelenovsky, Y. Yuan, and G. Del Galdo, "Low complexity blind separation technique to solve the permutation ambiguity of convolutive speech mixtures," in *Proceedings of the 10th International Conference on Signal Processing and Communication Systems, ICSPCS 2016*, pp. 1–10, 2016.

[38] B. D. Van Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[39] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[40] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[41] "RF Power Splitter, combiner, divider - IP67, 4 Way, SMA Female, 2.4 - 6.0 GHz," https://www.instockwireless.com/rf_power_splitter_combiner_divider_pd2578.htm.

[42] “Cable assembly SMA long 305 mm,” https://www.instockwireless.com.

[43] A. Quinlan, J.-P. Barbot, P. Larzabal, and M. Haardt, “Model order selection for short data: an exponential fitting test EFT,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 201, 2007.

[44] J. Grouffaud, P. Larzabal, and H. Clergeot, “Some properties of ordered eigenvalues of a Wishart matrix: Application in detection test and model order selection,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 2463–2466, IEEE, 1996.

[45] H. Akaike, “Information theory as an extension of the maximum likelihood,” in *Proceeding of International Symposium on Information Theory*, pp. 267–281, 1973.

[46] L. C. Zhao, P. R. Krishnaiah, and Z. D. Bai, “On detection of the number of signals in presence of white noise,” *Journal of Multivariate Analysis*, vol. 20, no. 1, pp. 1–25, 1986.

[47] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[48] M. O. Ulfarsson and V. Solo, “Dimension estimation in noisy PCA with SURE and random matrix theory,” *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 5804–5816, 2008.

[49] E. Radoi and A. Quinquis, “A new method for estimating the number of harmonic components in noise with application in high resolution radar,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 8, pp. 1177–1188, 2004.

[50] R. Badeau, B. David, and G. Richard, “Selecting the modeling order for the esprit high resolution method: An alternative approach,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2, pp. ii–1025, IEEE, 2004.

[51] J.-M. Papy, L. De Lathauwer, and S. Van Huffel, “A shift invariance-based order-selection technique for exponential data modelling,” *IEEE Signal Processing Letters*, vol. 14, no. 7, pp. 473–476, 2007.

[52] J. P. C. L. da Costa, M. Haardt, F. Romer, and G. Del Galdo, “Enhanced model order estimation using higher-order arrays,” in *Proceedings of the 2007 IEEE Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers (ACSSC)*, pp. 412–416, IEEE, 2007.

[53] J. P. C. L. da Costa, M. Haardt, F. Romer, and G. Del Galdo, “Enhanced model order estimation using higher-order arrays,” in *Proceedings of the 2007 IEEE Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers (ACSSC)*, pp. 412–416, IEEE, 2007.

[54] J. P. C. L. da Costa, F. Roemer, M. Haardt, and R. T. de Sousa Jr., “Multi-dimensional model order selection,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, article 26, 2011.

[55] J. P. C. L. Da Costa, M. Haardt, and F. Römer, “Robust methods based on the hosvd for estimating the model order in parafac models,” in *Proceedings of the 5th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 510–514, IEEE, Germany, 2008.

[56] J. P. C. L. da Costa, *Parameter Estimation Techniques for Multi-Dimensional Array Signal Processing*, Shaker Verlag, 2010.

[57] J. P. C. L. da Costa, F. Roemer, F. A. de Castro, R. F. Ramos, L. Sabirova, and S. Schwarz, “Ilmenau package for model order selection and evaluation of model order estimation scheme of users of MIMO channel sounders,” in *Proceedings of the XXIX Simpósio Brasileiro de Telecomunicações (SBrT)*, XXIX Simpósio Brasileiro de Telecomunicações (SBrT), Curitiba, Brazil, 2011.

[58] T. P. Kurpjuhn, M. T. Ivrlač, and J. A. Nossek, “Vandermonde invariance transformation,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 2929–2932, IEEE, 2001.

[59] J. E. Evans, J. R. Johnson, and D. F. Sun, “Application of advanced signal processing techniques to angle of arrival estimation in ATC navigation and surveillance systems,” ser. Technical report (Lincoln Laboratory), Massachusetts Institute of Technology, Lincoln Laboratory, 1982, https://books.google.com.br/books?id=lOIjSwAACAAJ.

[60] D. A. Linebarger, R. D. DeGroat, and E. M. Dowling, “Efficient direction-finding methods employing forward/backward averaging,” *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 2136–2145, 1994.

[61] S. U. Pillai and B. H. Kwon, “Forward/backward spatial smoothing techniques for coherent signal identification,” *IEEE Transactions on Signal Processing*, vol. 37, no. 1, pp. 8–15, 1989.

[62] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[63] R. K. Miranda, J. P. C. da Costa, B. Guo, A. L. de Almeida, G. Del Galdo, and R. T. de Sousa, “Low-complexity and high-accuracy semi-blind joint channel and symbol estimation for massive mimo-ofdm,” *Circuits, Systems, and Signal Processing*, pp. 1–23, 2018.

[64] “SDR Starter Kit 2x2 MIMO 50 MHz to 6 GHz PC based,” http://akashkosgi.wixsite.com/agile-solutions/sdr-starter-kit-c1a59.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 3619818, 10 pages
https://doi.org/10.1155/2019/3619818

Research Article

Compressed Sensing PARALIND Decomposition-Based Coherent Angle Estimation for Uniform Rectangular Array

Wu Wei,[1] Xu Le,[1] Zhang Xiaofei ,[1] and Li Jianfeng [1,2]

[1] *College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[2] *College of Computer and Information, Hohai University, Nanjing 211100, China*

Correspondence should be addressed to Li Jianfeng; lijianfengtin@126.com

Received 17 November 2018; Revised 27 March 2019; Accepted 27 May 2019; Published 24 June 2019

Guest Editor: Alex da Silva

In this paper, the topic of coherent two-dimensional direction of arrival (2D-DOA) estimation is investigated. Our study jointly utilizes the compressed sensing (CS) technique and the parallel profiles with linear dependencies (PARALIND) model and presents a 2D-DOA estimation algorithm for coherent sources with the uniform rectangular array. Compared to the traditional PARALIND decomposition, the proposed algorithm owns lower computational complexity and smaller data storage capacity due to the process of compression. Besides, the proposed algorithm can obtain autopaired azimuth angles and elevation angles and can achieve the same estimation performance as the traditional PARALIND, which outperforms some familiar algorithms presented for coherent sources such as the forward backward spatial smoothing-estimating signal parameters via rotational invariance techniques (FBSS-ESPRIT) and forward backward spatial smoothing-propagator method (FBSS-PM). Extensive simulations are provided to validate the effectiveness of the proposed CS-PARALIND algorithm.

1. Introduction

Array signal processing has aroused considerable concerns in recent decades owing to its extensive engineering application in satellite communication, radar, sonar, and some other fields [1–4]. In array signal processing, spectrum estimation, also known as direction of arrival (DOA) estimation, is a crucial issue. Till now, there are already many neoteric algorithms [5–8] proposed for DOA estimation with linear array, which include estimating signal parameters via rotational invariance techniques (ESPRIT) algorithm [5, 6], multiple signal classification (MUSIC) algorithm [7], and propagator method (PM) [8]. Compared with linear array, rectangular array can measure both azimuth angle and elevation angle, and hence 2D-DOA estimation with rectangular array has motivated enormous investigations. Many traditional DOA estimation methods have already been extended to 2D-DOA estimation [9–12]. For example, the 2D-ESPRIT algorithms [9–11] have utilized the invariance property to obtain the 2D-DOA estimation with uniform rectangular array, and the 2D-MUSIC [12] algorithm has also proved to be applicable. Some other algorithms such as angle estimation with generalized coprime planar array [13] and 2D-DOA estimation with nested subarrays [14] have better performance for 2D-DOA estimation as well.

However, in many practical situations, the complex propagation environment usually leads to the presence of coherent signals, which makes it complicated to obtain the valid DOA estimation. Therefore, the research on coherent angle estimation has gained great significance. The algorithms mentioned above are only applicable to noncoherent sources and the coherent sources will lead to severe invalidity for these methods. Some other coherent estimation methods including traditional forward spatial smoothing (FSS) or forward backward spatial smoothing (FBSS) [15, 16] algorithm have good estimation performance. Exploiting coprime multiple-input multiple-output (MIMO) radar, [17] has proposed the DOA estimation method for mixed coherent and uncorrelated targets and [18] obtained the DOA estimation of coherent sources via fourth-order cumulants.

The trilinear decomposition, namely, parallel factor (PARAFAC) technique [19–21], has been widely employed to resolve the problem of angular estimation with rectangular array [22]. However, the decomposition of PARAFAC model fails to work if it contains coherent sources. The parallel profiles with linear dependencies (PARALIND) model [23, 24] discussed in this paper can be regarded as a generalization of PARAFAC model and is efficient to solve the problem of coherent DOA estimation, where the PARAFAC method usually cannot present significant results. In [25, 26], the PARALIND decomposition method has already been successfully applied to obtain the coherent sources estimation with MIMO radar and acoustic vector-sensor array. However, the traditional algorithms based on PARALIND decomposition involve heavy computational burden and huge capacity consumption of data storage.

Compressed sensing (CS) [27, 28] has aroused considerable concern, which is introduced to areas including channel estimation, image, beamforming, and radar [29–32]. Specifically, the angular information of sources can be structured as a sparse vector and hence the CS technique can be directly utilized [33, 34]. By applying CS theory, many novel parameter estimation algorithms have been proposed for different scenarios. Reference [35] proposed a CS-based angle estimation method for noncircular sources which can be applied to MIMO radar, while [33, 34] combined the CS technique with PARAFAC model and proposed an estimation algorithm for joint angle and frequency. Based on the CS PARAFAC model, a 2D-DOA estimation method was presented in [36] with a uniform rectangular array, whereas it works only for noncoherent sources.

In this paper, we propose a CS-PARALIND algorithm for coherent sources to extract the DOA estimates with a uniform rectangular array by generalizing the work presented in [36] and the CS theory for PARALIND decomposition. We first construct the received data model which can be transformed to the PARALIND model. Then, we compress the data model and perform the PARALIND decomposition on it to achieve the estimation of compressed direction matrices. Finally, to acquire the 2D-DOA estimation, we formulate a sparse recovery problem which can be solved by the orthogonal matching pursuit (OMP) method [37]. Note that the compression process only compresses the directional matrix and the source matrix while the coherent matrix remains the same. Therefore, the coherent structure of impinging signals is not destroyed after compression. Comparing to [36], we summarize the main contributions of our research in this paper:

(1) We construct the received data model of coherent signals for uniform rectangular array which is suitable for the PARALIND decomposition.

(2) We generalize the method in [36] and propose the compressed sensing PARALIND (CS-PARALIND) model. Notably, there is no existing report for CS-PARALIND decomposition so far as we know.

(3) We develop the CS-PARALIND decomposition method to obtain the coherent 2D-DOA estimation for uniform rectangular array and give the detailed description of it.

In this paper, the proposed algorithm can obtain autopaired 2D-DOA estimation of coherent signals. In addition, the corresponding correlated matrix can also be obtained. The proposed algorithm can attain the same angle estimation performance as the traditional PARALIND method [25]. Compared to the FBSS-ESPRIT and the FBSS-PM algorithm, our method can achieve better DOA estimation performance. In addition, due to the process of compression, the proposed algorithm consumes lower computational burden and requires limited storage capacity in practical application. The Cramér-Rao bound (CRB) for the DOA estimation with uniform rectangular array is also provided in this paper. A series of simulation results verify the effectiveness of our approach.

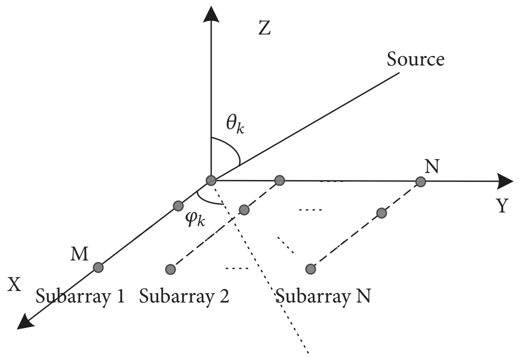


Figure 1: The structure of uniform rectangular array.

The remainder of our paper is organized as follows: Section 2 presents the received data model of coherent signals with uniform rectangular array. Section 3 depicts the detailed derivation of the proposed CS-PARALIND algorithm as well as the uniqueness certification and complexity analysis. Numerical simulations are exhibited in Section 4, and we conclude this paper in Section 5.

Notation. $\oplus$, $\circ$, and $\otimes$, respectively, represent *Hadamard* product, *Khatri–Rao* product, and *Kronecker* product. $(.)^H$, $(.)^T$, $(.)^*$, $(.)^{-1}$, and $(.)^+$ stand for the operations of conjugate-transpose, transpose, conjugation, inverse, and pseudoinverse. The *Frobenius* norm and l_0–norm are denoted by $\| \bullet \|_F$ and $\| \bullet \|_0$. diag($\mathbf{b}$) is a diagonal matrix composed of elements in vector $\mathbf{b}$ while $\text{diag}^{-1}(\mathbf{A})$ produces a column vector composed of the diagonal elements of matrix $\mathbf{A}$. $D_n(\mathbf{A})$ denotes a diagonal matrix which consists of the n-th row of $\mathbf{A}$. *span*($\mathbf{H}$) represents the subspace spanned by the columns of $\mathbf{H}$. *vec*($\mathbf{A}$) means stacking the columns of matrix $\mathbf{A}$. ∂ means partial derivatives.

2. Data Model

Assume that there are K far-field narrow-band signals with DOA (θ_k, φ_k) impinging on a uniform rectangular array consisting of $M \times N$ sensors, where φ_k is the azimuth angle of the k-th signal, θ_k represents the elevation angle, and the distance between any two adjacent elements is d. The sources number K is known and the noise is additive white Gaussian which is assumed to be independent and uncorrelated with received signals. Figure 1 shows the structure of the array.

For the first subarray in the uniform rectangular array, the received data at t-th time can be represented as [36]

$$\mathbf{x}_1(t) = \mathbf{A}_x \mathbf{s}_0(t) + \mathbf{n}_1(t), \tag{1}$$

where $\mathbf{A}_x = [\mathbf{a}_x(\theta_1, \varphi_1), \mathbf{a}_x(\theta_2, \varphi_2), \cdots, \mathbf{a}_x(\theta_K, \varphi_K)]$ denotes the directional matrix of the first subarray with $\mathbf{a}_x(\theta_k, \varphi_k) = [1, \exp(j2\pi d \cos\varphi_k \sin\theta_k/\lambda), \cdots, \exp(j2\pi(M-1)d\cos\varphi_k \sin\theta_k/\lambda)]^T$ and $j = \sqrt{-1}$. The received noise of the first subarray is denoted by $\mathbf{n}_1(t)$, and $\mathbf{s}_0(t)$ is the $K \times 1$ source vector of the K signals. Then it follows that the received data of n-th subarray at t-th time can be written as [36]

$$\mathbf{x}_n(t) = \mathbf{A}_x \mathbf{\Phi}^{n-1} \mathbf{s}_0(t) + \mathbf{n}_n(t), \tag{2}$$

where $\mathbf{\Phi} = \mathrm{diag}(e^{j2\pi d \sin\theta_1 \sin\varphi_1/\lambda}, \cdots, e^{j2\pi d \sin\theta_K \sin\varphi_K/\lambda})$ and $\mathbf{n}_n(t)$ stands for the received noise of the n-th subarray. Therefore, the received data of the entire rectangular array can be denoted by [38]

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \\ \vdots \\ \mathbf{x}_N(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_x \\ \mathbf{A}_x\mathbf{\Phi} \\ \vdots \\ \mathbf{A}_x\mathbf{\Phi}^{N-1} \end{bmatrix} \mathbf{s}_0(t) + \begin{bmatrix} \mathbf{n}_1(t) \\ \mathbf{n}_2(t) \\ \vdots \\ \mathbf{n}_N(t) \end{bmatrix}, \tag{3}$$

or more compactly as

$$\mathbf{x}(t) = \left(\mathbf{A}_y \circ \mathbf{A}_x\right) \mathbf{s}_0(t) + \mathbf{n}(t), \tag{4}$$

where $\mathbf{A}_y = [\mathbf{a}_y(\theta_1, \varphi_1), \mathbf{a}_y(\theta_2, \varphi_2), \cdots, \mathbf{a}_y(\theta_K, \varphi_K)]$ with $\mathbf{a}_y(\theta_k, \varphi_k) = [1, \exp(j2\pi d \sin\varphi_k \sin\theta_k/\lambda), \cdots, \exp(j2\pi(N-1)d\sin\varphi_k \sin\theta_k/\lambda)]$ and $\mathbf{n}(t)$ is the noise of the whole rectangular array with $\mathbf{n}(t) = [\mathbf{n}_1(t)^T, \mathbf{n}_2(t)^T, \ldots, \mathbf{n}_N(t)^T]^T$.

By exploiting J snapshots, the received data can be denoted as [36]

$$\widetilde{\mathbf{X}} = [\mathbf{x}(t_1), \mathbf{x}(t_2), \cdots, \mathbf{x}(t_J)], \tag{5}$$

where $\widetilde{\mathbf{X}}$ stands for the noisy received signals, and rewriting (5) in matrix form, we obtain [38]

$$\begin{aligned} \widetilde{\mathbf{X}} = \begin{bmatrix} \widetilde{\mathbf{X}}_1 \\ \widetilde{\mathbf{X}}_2 \\ \vdots \\ \widetilde{\mathbf{X}}_N \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_x D_1(\mathbf{A}_y) \\ \mathbf{A}_x D_2(\mathbf{A}_y) \\ \vdots \\ \mathbf{A}_x D_N(\mathbf{A}_y) \end{bmatrix} \mathbf{S}_0 + \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \\ \vdots \\ \mathbf{N}_N \end{bmatrix} \\ &= \left[\mathbf{A}_y \circ \mathbf{A}_x\right] \mathbf{S}_0 + \mathbf{N}, \end{aligned} \tag{6}$$

where $\mathbf{S}_0 = [\mathbf{s}_0(t_1), \mathbf{s}_0(t_2), \cdots, \mathbf{s}_0(t_J)] \in \mathbf{C}^{K\times J}$ and $\widetilde{\mathbf{X}}_n = \mathbf{A}_x D_n(\mathbf{A}_y)\mathbf{S}_0 + \mathbf{N}_n$. The noise of J snapshots is $\mathbf{N} = [\mathbf{n}(t_1), \mathbf{n}(t_2), \cdots, \mathbf{n}(t_J)] \in \mathbf{C}^{MN\times J}$.

The received data matrix shown in (6) is a traditional PARAFAC model [19–22], the decomposition of which is usually nonunique if there exist coherent signals [19]. Assume that among the K received sources, there are K_1 groups of coherent signals; then (6) can be represented as [25]

$$\widetilde{\mathbf{X}} = \left[\mathbf{A}_y \circ \mathbf{A}_x\right] \mathbf{\Gamma}\mathbf{S} + \mathbf{N}, \tag{7}$$

where $\mathbf{S} \in \mathbf{C}^{K_1\times J}$ is the source matrix of J snapshots of K_1 noncoherent signals. $\mathbf{\Gamma} \in \mathbf{C}^{K\times K_1}$ is the corresponding correlated matrix satisfying $\mathbf{\Gamma}\mathbf{S} = \mathbf{S}_0$.

3. CS-PARALIND Decomposition-Based Algorithm

The PARALIND algorithm suffers from expensive computational cost, especially in the case of large number of sensors or snapshots. Specifically, the angular information of received sources can be structured as a sparse vector and hence the CS technique can be directly utilized [33]. To counter this problem, we bring in the CS theory by compressing the received signal in (7) to a smaller matrix, followed by the PARALIND decomposition and, finally, acquire the 2D-DOA estimates by exploiting sparse recovery method.

3.1. Compression. We compress the received data $\underline{\mathbf{X}} \in \mathbf{C}^{M\times J\times N}$ into a smaller matrix $\underline{\mathbf{X}'} \in \mathbf{C}^{M'\times J'\times N'}$ with three compression matrices $\mathbf{U} \in \mathbf{C}^{M\times M'}$, $\mathbf{V} \in \mathbf{C}^{N\times N'}$, and $\mathbf{W} \in \mathbf{C}^{J\times J'}$, with $M' < M$, $J' < J$ and $N' < N$, which is illustrated in Figure 2.

The three compression matrices can be constructed via some random special matrices, *e.g.*, the random Gaussian, Bernoulli, and partial Fourier matrices or the Tucker3 decomposition introduced in [33, 39].

Applying $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ to (7), we can obtain the compressed received data $\widetilde{\mathbf{X}}' \in \mathbf{C}^{(M'N'\times J')}$ as [27]

$$\begin{aligned} \widetilde{\mathbf{X}}' &= \left(\mathbf{V}^T \otimes \mathbf{U}^T\right) \widetilde{\mathbf{X}}^{(MN\times J)} \mathbf{W} \\ &= \left(\mathbf{V}^T \otimes \mathbf{U}^T\right) \left[\mathbf{A}_y \circ \mathbf{A}_x\right] \mathbf{\Gamma}\mathbf{S}\mathbf{W} + \left(\mathbf{V}^T \otimes \mathbf{U}^T\right) \mathbf{N}\mathbf{W}. \end{aligned} \tag{8}$$

And according to the property of Khatri–Rao product [33], we have $(\mathbf{V}^T \otimes \mathbf{U}^T)(\mathbf{A}_y \circ \mathbf{A}_x) = (\mathbf{V}^T\mathbf{A}_y) \circ (\mathbf{U}^T\mathbf{A}_x)$, and then (8) can be further simplified to

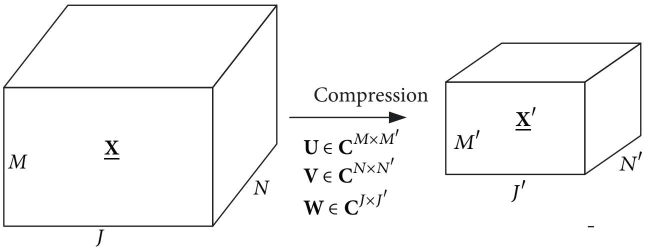
$$\widetilde{\mathbf{X}}' = \left[\mathbf{A}'_y \circ \mathbf{A}'_x\right] \mathbf{\Gamma}\mathbf{S}' + \mathbf{N}', \tag{9}$$

where ${\mathbf{A}'_x}^{(M'\times K)} = \mathbf{U}^T\mathbf{A}_x$, ${\mathbf{A}'_x}^{(M'\times K)} = \mathbf{U}^T\mathbf{A}_x$, ${\mathbf{A}'_y}^{(N'\times K)} = \mathbf{V}^T\mathbf{A}_y$, ${\mathbf{S}'}^{(K\times J')} = \mathbf{S}\mathbf{W}$, and $\mathbf{N}' = (\mathbf{V}^T \otimes \mathbf{U}^T)\mathbf{N}\mathbf{W}$.

3.2. PARALIND Decomposition. To transform (9) to PARALIND model, we define $\widetilde{\mathbf{Y}} = {\widetilde{\mathbf{X}}}'^{T}$, or equivalently,

$$\widetilde{\mathbf{Y}} = \left(\mathbf{\Gamma}\mathbf{S}'\right)^T \left(\mathbf{A}'_y \circ \mathbf{A}'_x\right)^T + {\mathbf{N}'}^T = \mathbf{Y} + {\mathbf{N}'}^T, \tag{10}$$

where $\mathbf{Y} = (\mathbf{\Gamma}\mathbf{S}')^T(\mathbf{A}'_y \circ \mathbf{A}'_x)^T$ denotes the noise-free compressed received matrix.



Compression
$\mathbf{U} \in \mathbf{C}^{M\times M'}$
$\mathbf{V} \in \mathbf{C}^{N\times N'}$
$\mathbf{W} \in \mathbf{C}^{J\times J'}$
$\underline{\mathbf{X}}$
M
N
J
$\underline{\mathbf{X}}'$
M'
N'
J'

FIGURE 2: Compression processing.

Obviously, (10) is equivalent to the PARALIND model [20] and the least fitting for it amounts to [23]

$$\min_{\boldsymbol{\Gamma},\mathbf{S},\mathbf{A}'_y,\mathbf{A}'_x} \left\| \widetilde{\mathbf{Y}} - \left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T \left(\mathbf{A}'_y \circ \mathbf{A}'_x\right)^T \right\|_F^2 . \tag{11}$$

Under noise-free condition, (10) can be rewritten as

$$\begin{aligned} \mathbf{Y} = \left[\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_{N'}\right] &= \left[\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_1\left(\mathbf{A}'_y\right)\mathbf{A}'^T_x, \left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T \right. \\ &\left. \cdot D_2\left(\mathbf{A}'_y\right)\mathbf{A}'^T_x, \cdots, \left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_{N'}\left(\mathbf{A}'_y\right)\mathbf{A}'^T_x \right], \end{aligned} \tag{12}$$

where $\mathbf{Y}_n = (\boldsymbol{\Gamma}\mathbf{S}')^T D_n(\mathbf{A}'_y)\mathbf{A}'^T_x \in \mathbf{C}^{J'\times M'}$, $n = 1, 2, \cdots, N'$.

After vectoring $\mathbf{Y}_n$, we have [38]

$$\begin{aligned} vec\left(\mathbf{Y}_n\right) &= vec\left(\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_n\left(\mathbf{A}'_y\right)\mathbf{A}'^T_x\right) \\ &= \left(\mathbf{A}'_x D_n\left(\mathbf{A}'_y\right) \otimes \mathbf{S}'^T\right) vec\left(\boldsymbol{\Gamma}^T\right). \end{aligned} \tag{13}$$

Stacking these vectors by columns

$$\begin{bmatrix} vec\left(\mathbf{Y}_1\right) \\ vec\left(\mathbf{Y}_2\right) \\ \vdots \\ vec\left(\mathbf{Y}_{N'}\right) \end{bmatrix} = \begin{bmatrix} \mathbf{A}'_x D_1\left(\mathbf{A}'_y\right) \otimes \mathbf{S}'^T \\ \mathbf{A}'_x D_2\left(\mathbf{A}'_y\right) \otimes \mathbf{S}'^T \\ \vdots \\ \mathbf{A}'_x D_{N'}\left(\mathbf{A}'_y\right) \otimes \mathbf{S}'^T \end{bmatrix} vec\left(\boldsymbol{\Gamma}^T\right). \tag{14}$$

Equation (14) also can be denoted as

$$vec\left(\mathbf{Y}\right) = \left[\left(\mathbf{A}'_y \circ \mathbf{A}'_x\right) \otimes \mathbf{S}'^T\right] vec\left(\boldsymbol{\Gamma}^T\right). \tag{15}$$

Then under noise environment, $vec(\boldsymbol{\Gamma}^T)$ can be obtained by

$$vec\left(\boldsymbol{\Gamma}^T\right) = \left[\left(\mathbf{A}'_y \circ \mathbf{A}'_x\right) \otimes \mathbf{S}'^T\right]^+ vec\left(\widetilde{\mathbf{Y}}\right). \tag{16}$$

Now we can update the value of $\boldsymbol{\Gamma}$ via transforming $vec(\boldsymbol{\Gamma}^T)$ to its original modality.

According to (11), the least square (LS) update for $\mathbf{S}'^T$ is

$$\mathbf{S}'^T = \widetilde{\mathbf{Y}}\left(\left(\left(\mathbf{A}'_x \circ \mathbf{A}'_y\right)\boldsymbol{\Gamma}\right)^T\right)^+. \tag{17}$$

From (12), it follows that the covariance matrix of $\mathbf{Y}_n$ can be written as

$$\begin{aligned} \sum_{n=1}^{N'} \mathbf{Y}_n{}^H \mathbf{Y}_n &= \left(\sum_{n=1}^{N'} \mathbf{Y}_n{}^H \left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_n\left(\mathbf{A}'_y\right)\right)\mathbf{A}'^T_x \\ &= \mathbf{A}'^*_x \left(\sum_{n=1}^{N'} D_n{}^*\left(\mathbf{A}'_y\right)\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^*\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_n\left(\mathbf{A}'_y\right)\right)\mathbf{A}'^T_x . \end{aligned} \tag{18}$$

As $\mathbf{A}'_x$ is of full column rank and $\sum_{n=1}^{N'} D_n{}^*(\mathbf{A}'_y)(\boldsymbol{\Gamma}\mathbf{S}')^*(\boldsymbol{\Gamma}\mathbf{S}')^T D_n(\mathbf{A}'_y)$ is nonsingular [24], we can obtain $\mathbf{A}'^*_x$ via

$$\begin{aligned} \mathbf{A}'^*_x &= \left(\sum_{n=1}^{N'} \widetilde{\mathbf{Y}}_n{}^H \left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_n\left(\mathbf{A}'_y\right)\right) \\ &\quad \cdot \left(\sum_{n=1}^{N'} D_n{}^*\left(\mathbf{A}'_y\right)\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^*\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_n\left(\mathbf{A}'_y\right)\right)^{-1} \\ &= \left(\sum_{n=1}^{N'} \widetilde{\mathbf{Y}}_n{}^H \left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_n\left(\mathbf{A}'_y\right)\right) \\ &\quad \cdot \left(\sum_{n=1}^{N'} \left(\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^*\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T\right) \oplus \left(\mathbf{A}'^H_y \mathbf{A}'_y\right)\right)^{-1}. \end{aligned} \tag{19}$$

Substitute $\mathbf{Y}_n$ of (12) into the following

$$\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^* \mathbf{Y}_n \mathbf{A}'_x = \left(\boldsymbol{\Gamma}\mathbf{S}'\right)^*\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T D_n\left(\mathbf{A}'_y\right)\mathbf{A}'^T_x \mathbf{A}'_x. \tag{20}$$

Taking the diagonal elements of both sides in (20), we have

$$\begin{aligned} &\operatorname{diag}^{-1}\left(\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^* \mathbf{Y}_n \mathbf{A}'_x\right) \\ &\quad = \left(\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^*\left(\boldsymbol{\Gamma}\mathbf{S}'\right)^T\right) \\ &\qquad \oplus \left(\mathbf{A}'^T_x \mathbf{A}'_x\right)\operatorname{diag}^{-1}\left(D_n\left(\mathbf{A}'_y\right)\right). \end{aligned} \tag{21}$$

By enforcing left multiplication operation of (21), we get

$$\begin{aligned} \operatorname{diag}^{-1}\left(D_n\left(\mathbf{A}_y'\right)\right) &= \left(\left(\left(\mathbf{\Gamma}\mathbf{S}'\right)^{*}\left(\mathbf{\Gamma}\mathbf{S}'\right)^{T}\right)\right. \\ &\quad \left.\oplus\left(\mathbf{A}_x'^{T}\mathbf{A}_x'\right)\right)^{-1}\operatorname{diag}^{-1}\left(\left(\mathbf{\Gamma}\mathbf{S}'\right)^{*}\widetilde{\mathbf{Y}}_n\mathbf{A}_x'\right). \end{aligned} \quad (22)$$

And hence, $\mathbf{A}_y$ can be easily obtained form (22).

For the PARALIND decomposition above, according to (17), (18), and (19), we repeatedly update the estimation of $\mathbf{A}_x'$, $\mathbf{A}_y'$, $\mathbf{S}'$, and $\mathbf{\Gamma}$ until convergence. Define the sum of squared residuals (SSR) of the k-th iteration as $SSR_k = \sum_{i=1}^{N'M'}\sum_{j=1}^{J'}|c_{ij}|^2$,where c_{ij} is the (i, j) element of $\mathbf{C} = \widetilde{\mathbf{Y}} - (\widehat{\mathbf{\Gamma}}\widehat{\mathbf{S}}')^T(\widehat{\mathbf{A}}_y' \circ \widehat{\mathbf{A}}_x')^T$. Define the convergence rate of PARALIND decomposition as $SSR_{rate} = (SSR_k - SSR_{k-1})/SSR_{k-1}$ [40]. When SSR_{rate} is smaller than a certain small value (determined by the noise level), then the above iteration process can be considered convergent.

Based on the PARALIND decomposition, the estimates of $\mathbf{A}_x'$ and $\mathbf{A}_y'$ can be achieved by

$$\widehat{\mathbf{A}}_x' = \mathbf{A}_x'\mathbf{\Pi}\mathbf{\Delta}_x + \mathbf{W}_x = \mathbf{U}^T\mathbf{A}_x\mathbf{\Pi}\mathbf{\Delta}_x + \mathbf{W}_x, \quad (23)$$

$$\widehat{\mathbf{A}}_y' = \mathbf{A}_y'\mathbf{\Pi}\mathbf{\Delta}_y + \mathbf{W}_y = \mathbf{V}^T\mathbf{A}_y\mathbf{\Pi}\mathbf{\Delta}_y + \mathbf{W}_y, \quad (24)$$

where $\mathbf{\Pi}$ stands for the permutation matrix and $\mathbf{\Delta}_y$ and $\mathbf{\Delta}_x$ are the diagonal scaling matrix of $\mathbf{A}_y'$ and $\mathbf{A}_x'$. In addition, $\mathbf{W}_x$ and $\mathbf{W}_y$ denote the estimation error.

Remark 1. In the proposed algorithm, the scale ambiguity can be eliminated by direct normalization while the permutation ambiguity makes no difference in the angle estimation.

3.3. DOA Estimation with Sparsity.

By utilizing $\widehat{\mathbf{A}}_x'$, $\widehat{\mathbf{A}}_y'$, the 2D-DOA estimation can be obtained with sparsity. With noiseless case, we use $\widehat{\mathbf{a}}_{xk}'$ and $\widehat{\mathbf{a}}_{yk}'$ to denote the k-th column of $\widehat{\mathbf{A}}_x'$, $\widehat{\mathbf{A}}_y'$, respectively. From (23) and (24), it follows that [36]

$$\widehat{\mathbf{a}}_{xk}' = \mathbf{U}^H\rho_{xk}\mathbf{a}_{xk}, \quad (25)$$

$$\widehat{\mathbf{a}}_{yk}' = \mathbf{V}^H\rho_{yk}\mathbf{a}_{yk}, \quad (26)$$

where $\mathbf{a}_{xk}$ and $\mathbf{a}_{yk}$ represent the k-th column in the directional matrices $\mathbf{A}_x$ and $\mathbf{A}_y$. In addition, ρ_{xk} and ρ_{yk} denote the scaling coefficients.

Then we construct two Vandermonde matrices $\mathbf{A}_{sx} \in \mathbf{C}^{M\times Q}$ and $\mathbf{A}_{sy} \in \mathbf{C}^{N\times Q}$ $(Q >> M, Q >> N)$, the columns of which consist of the steering vectors corresponding to each potential source location [36].

$$\begin{aligned} \mathbf{A}_{sx} &= \left[\mathbf{a}_{sx1}, \mathbf{a}_{sx2}, \cdots, \mathbf{a}_{sxQ}\right] \\ &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{j2\pi d\mathbf{g}(1)/\lambda} & e^{j2\pi d\mathbf{g}(2)/\lambda} & \vdots & e^{j2\pi d\mathbf{g}(Q)/\lambda} \\ \vdots & \vdots & \vdots & \vdots \\ e^{j2\pi(M-1)d\mathbf{g}(1)/\lambda} & e^{j2\pi(M-1)d\mathbf{g}(2)/\lambda} & \cdots & e^{j2\pi(M-1)d\mathbf{g}(Q)/\lambda} \end{bmatrix}, \end{aligned} \quad (27)$$

$$\begin{aligned} \mathbf{A}_{sy} &= \left[\mathbf{a}_{sy1}, \mathbf{a}_{sy2}, \cdots, \mathbf{a}_{syQ}\right] \\ &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{j2\pi d\mathbf{g}(1)/\lambda} & e^{j2\pi d\mathbf{g}(2)/\lambda} & \vdots & e^{j2\pi d\mathbf{g}(Q)/\lambda} \\ \vdots & \vdots & \vdots & \vdots \\ e^{j2\pi(N-1)d\mathbf{g}(1)/\lambda} & e^{j2\pi(N-1)d\mathbf{g}(2)/\lambda} & \cdots & e^{j2\pi(N-1)d\mathbf{g}(Q)/\lambda} \end{bmatrix}, \end{aligned} \quad (28)$$

where $\mathbf{g}$ is a sampling vector with $\mathbf{g}(q) = -1 + 2q/Q$, $q = 0, 1, \ldots, Q$. $\mathbf{A}_{sx}$ and $\mathbf{A}_{sy}$ can be referred to as the overcomplete dictionary for our 2D-DOA estimation [36]. Then (25), (26) can be converted to

$$\widehat{\mathbf{a}}_{xk}' = \mathbf{U}^T\mathbf{A}_{sx}\mathbf{x}_{sk}, \quad k = 1, \cdots, K, \quad (29)$$

$$\widehat{\mathbf{a}}_{yk}' = \mathbf{V}^T\mathbf{A}_{sy}\mathbf{y}_{sk}, \quad k = 1, \cdots, K, \quad (30)$$

where $\mathbf{x}_{sk}$ and $\mathbf{y}_{sk}$ are sparse, the estimates of which can be cast as an optimization problem, subject to l_0-norm constraint.

$$\begin{aligned} \min &\quad \left\|\widehat{\mathbf{a}}_{xk}' - \mathbf{U}^H\mathbf{A}_{sx}\mathbf{x}_{sk}\right\|_F^2, \\ st. &\quad \left\|\mathbf{x}_{sk}\right\|_0 = 1, \end{aligned} \quad (31)$$

$$\begin{aligned} \min &\quad \left\|\widehat{\mathbf{a}}_{yk}' - \mathbf{V}^H\mathbf{A}_{sy}\mathbf{y}_{sk}\right\|_F^2, \\ st. &\quad \left\|\mathbf{y}_{sk}\right\|_0 = 1. \end{aligned} \quad (32)$$

$\mathbf{x}_{sk}$ and $\mathbf{y}_{sk}$ can be gained by the OMP recovery method [37].

Denote the maximum modulus of elements in $\mathbf{x}_{sk}$ and $\mathbf{y}_{sk}$ as q_{xk} and q_{yk}, respectively. Then the corresponding elements $\mathbf{g}(q_{xk})$ and $\mathbf{g}(q_{yk})$ in $\mathbf{A}_{sx}$ and $\mathbf{A}_{sy}$ are just the $\sin\theta_k\cos\varphi_k$ and $\sin\theta_k\sin\varphi_k$ estimation. Define $r_k = \mathbf{g}(q_{xk}) + j\mathbf{g}(q_{yk})$ and the estimates of elevation angles and azimuth angles can be obtained by

$$\widehat{\theta}_k = \sin^{-1}\left(abs\left(r_k\right)\right), \quad k = 1, 2, \ldots, K, \quad (33)$$

$$\widehat{\varphi}_k = angle\left(r_k\right), \quad k = 1, 2, \ldots, K, \quad (34)$$

where $abs(.)$ outputs the modulus value and $angle(.)$ computes the angular part of a complex number. Finally, the autopaired estimates of elevation and azimuth angles can be attained by employing the paired relationship in the directional matrices obtained by PARALIND decomposition.

3.4. The Procedure of the Proposed Algorithm.

Till now, the angle estimation of received coherent signals with a uniform rectangular array has been acquired and we provide the major steps as follows:

Step 1. Compress the received data $\widetilde{\mathbf{X}}$, construct the PARALIND model $\widetilde{\mathbf{Y}}$ according to (8), and then initialize the value of $\mathbf{A}_x'$, $\mathbf{A}_y'$, $\mathbf{\Gamma}$, and $\mathbf{S}'$.

Step 2. According to (16), (17), (19), and (22), repeatedly update the estimates of $\mathbf{A}_x'$, $\mathbf{A}_y'$, $\mathbf{S}'$, and $\mathbf{\Gamma}$ according to the convergence conditions.

Step 3. According to (31)-(34), the elevation and azimuth angles can be estimated by $\mathbf{A}_{sx}$ and $\mathbf{A}_{sy}$.

4. Performance Analysis

4.1. Complexity Analysis of the Proposed Algorithm. We analyze the complexity of the proposed algorithm in this subsection. For the proposed algorithm, the computational complexity is $O(M'N'MNJ + JM'N'J' + n_1(M'N'J'(2K^2{K_1}^2 + 2KK_1 + 2K + K_1) + M'N'(2{K_1}^2 + K^2 + KK_1 + 2K) + K^2(J'N' + J' + N'^2 + N' + M' + 2) + K^3{K_1}^3 + 2K^3 + {K_1}^3 + KK_1J'(N' + 1))) + KP(M + N))$, where n_1 denotes the time of iterations. With regard to the traditional PARALIND decomposition [23], it has $(n_2(MNJ(2K^2{K_1}^2 + 2KK_1 + 2K + K_1) + MN(2{K_1}^2 + K^2 + KK_1 + 2K) + K^2(JN + J + N^2 + N + M + 2) + K^3{K_1}^3 + 2K^3 + {K_1}^3 + KK_1J(N + 1)) + 2K^2(M+N)+6K^2)$, where n_2 represents the time of iterations. In the comparison of computational complexity, we have the parameters of $M = N = 10$, $K = 3$, and $K_1 = 2$. In addition, assume that $n_1 = n_2 = 20$, $P = 200$, and $M'/M = N'/N = J'/J = 0.5$. The comparison of complexity versus J is depicted in Figure 3, which concludes that the computational burden of the proposed algorithm is decreased compared to the traditional PARALIND algorithm.

4.2. CRB. Define $\mathbf{A} = [\mathbf{a}_y(\theta_1, \varphi_1) \otimes \mathbf{a}_x(\theta_1, \varphi_1), \cdots, \mathbf{a}_y(\theta_K, \varphi_K) \otimes \mathbf{a}_x(\theta_K, \varphi_K)]$ and according to [41], we derive the CRB

$$CRB = \frac{\sigma^2}{2J} \left\{ \mathrm{Re} \left[\mathbf{D}^H \mathbf{\Pi}_{\mathbf{A}}^{\perp} \mathbf{D} \oplus \widehat{\mathbf{P}}_s{}^T \right] \right\}^{-1}, \tag{35}$$

where J is the number of snapshots and $\mathbf{a}_k$ is the k-th column of $\mathbf{A}$. σ^2 denotes the noise power. $\mathbf{\Pi}_{\mathbf{A}}^{\perp} = \mathbf{I}_{MN} - \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H$ and $\mathbf{I}_{MN}$ is an $MN \times MN$ identity matrix. $\mathbf{D} = [\partial \mathbf{a}_1/\partial\theta_1, \partial \mathbf{a}_2/\partial\theta_2, ..., \partial \mathbf{a}_K/\partial\theta_K, \partial \mathbf{a}_1/\partial\varphi_1, \partial \mathbf{a}_2/\partial\varphi_2, ..., \partial \mathbf{a}_K/\partial\varphi_K]$ and $\widehat{\mathbf{P}}_s = (1/J)\sum_{t=1}^{J} \mathbf{s}(t)\mathbf{s}^H(t)$.

4.3. Advantages. We summarize the advantages of the proposed algorithm as follows:

(1) The proposed algorithm possesses lower computational cost and requires smaller data storage, due to the compression operation.

(2) The proposed algorithm can be applied to the coherent signals. Furthermore, the corresponding correlated matrix also can be obtained.

(3) The proposed algorithm can achieve the same angle estimation performance with the traditional PARALIND method and outperforms the FBSS-ESPRIT and FBSS-PM algorithm.

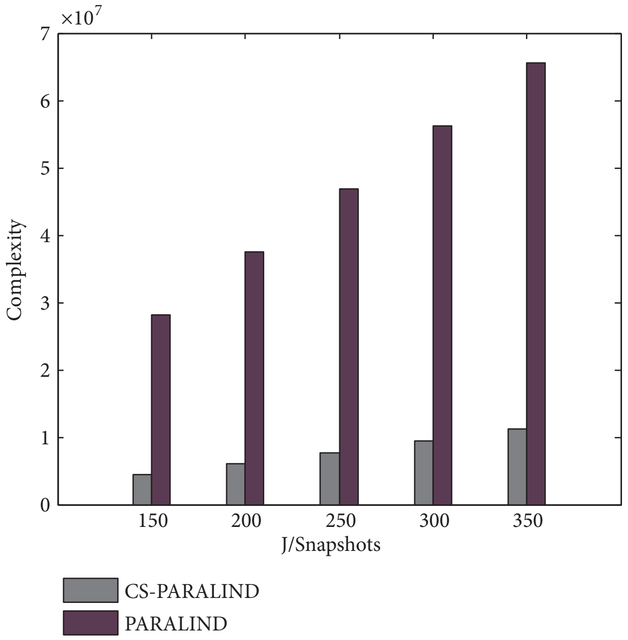


Figure 3: Comparison of complexity versus different snapshots (J).

5. Simulation Results

Suppose that $K = 3$ signals with two coherent signals and one noncoherent signal impinge on a rectangular array, where the correlated matrix is $\mathbf{\Gamma} = \left[\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{smallmatrix}\right]^T$. In the following simulations, we exploit the root mean square error (RMSE) to evaluate the DOA estimation performance and it is presented by

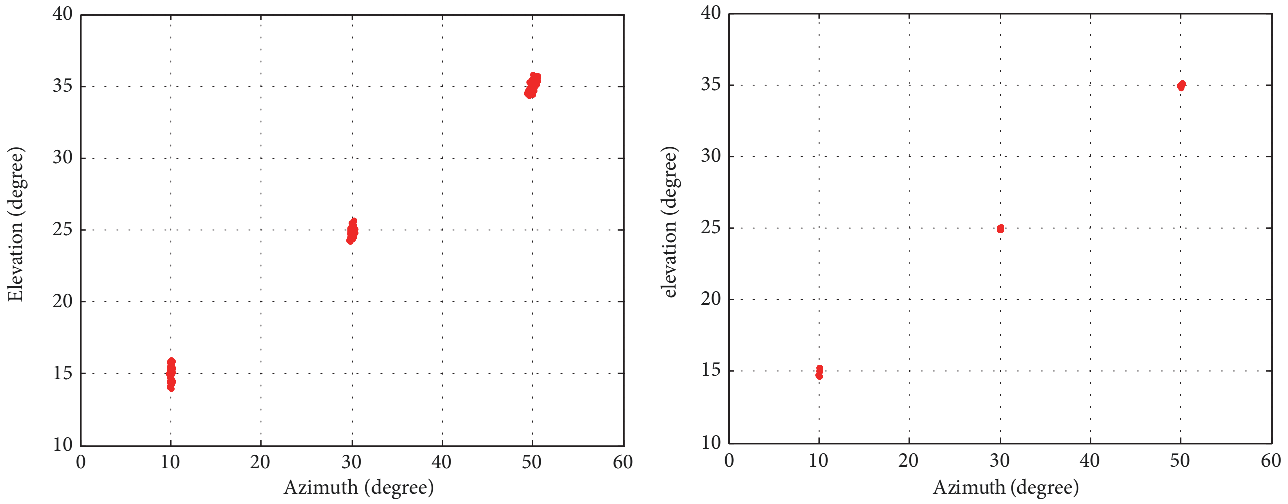
$$RMSE = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{L}\sum_{l=1}^{L}\left(\widehat{\varphi}_{k,l} - \varphi_k\right)^2 + \left(\widehat{\theta}_{k,l} - \theta_k\right)^2}, \tag{36}$$

where $\widehat{\varphi}_{k,l}$, $\widehat{\theta}_{k,l}$ are the estimations of φ_k, θ_k in l-th simulation, and the times of Monte-Carlo simulations are indicated by L. In the following simulations, we set $L = 1000$.

The locations of the three sources are $(\theta_1, \varphi_1) = (15^\circ, 10^\circ)$, $(\theta_2, \varphi_2) = (25^\circ, 30^\circ)$, and $(\theta_3, \varphi_3) = (35^\circ, 50^\circ)$. M, N, J, K are the number of rows and columns of the rectangular array, snapshots, and sources, respectively. And assume that the rectangular array is uniform and the distance of any two adjacent sensors is $\lambda/2$.

Simulation 1. Figure 4 shows the 2D-DOA estimation results of our proposed algorithm with SNR=0dB and SNR=20dB, respectively. The size of PARALIND model is defined as $M \times N \times J$. The received data matrix has the size of $10 \times 10 \times 200$ in this simulation, which becomes $7 \times 7 \times 140(M' \times N' \times J')$ after compression. Figure 4 indicates that the proposed algorithm is effective for 2D-DOA estimation of coherent sources.

Simulation 2. In Figure 5, we exhibit the comparison of the DOA estimation performance among the proposed algorithm, FSS-ESPRIT, FSS-PM, FBSS-ESPRIT, and FBSS-PM algorithm, and the traditional PARALIND algorithm. The



(a)

(b)

Figure 4: DOA estimation of the proposed algorithm with (a) SNR=0dB, (b) SNR=20dB.

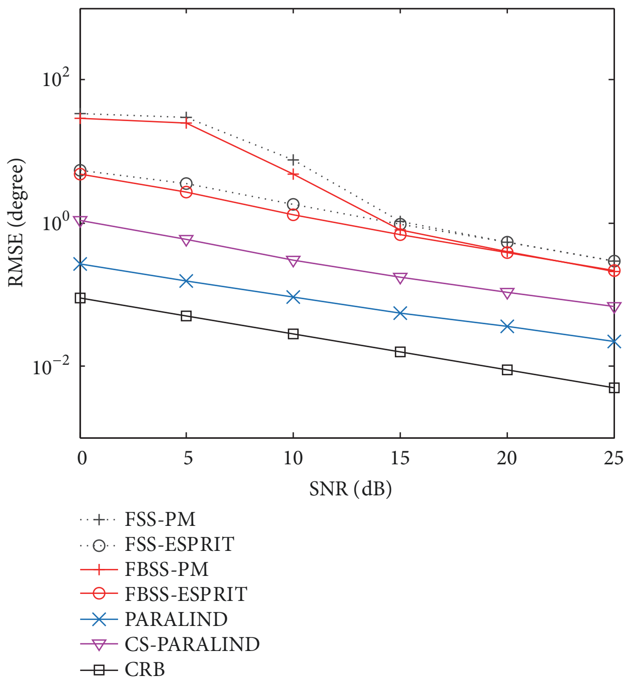


Figure 5: DOA estimation performance comparison.

received data matrix has the size of 10 × 10 × 100 and is compressed to the size of 8 × 8 × 80. It is illustrated clearly in Figure 5 that the approach algorithm can achieve the same DOA estimation performance compared with the traditional PARALIND algorithm and, furthermore, outperforms the other four algorithms.

Simulation 3. The DOA estimation performance result of the proposed algorithm versus compression ratio is provided in Figure 6, where the compression ratio is $P = M'/M = N'/N = J'/J$. In this part, the size of the original PARALIND model is $15 \times 15 \times 200$ with $P = 0.4$, $P = 0.6$, and $P = 0.8$. It is shown in Figure 6 that the estimation performance of our method improves with P getting larger.

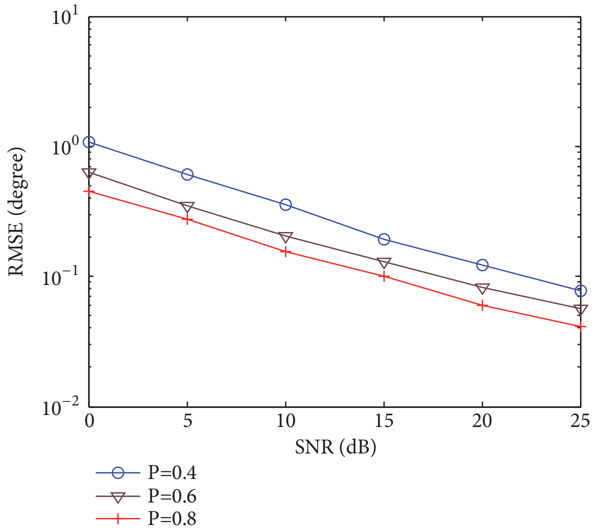


Figure 6: DOA estimation performance under different compression ratio.

Simulation 4. The angle estimation performance of the proposed algorithm versus J is illustrated in Figure 7, where the rectangular array is structured as $M = 10$, $N = 10$ and $M' = N' = 8$, $J'/J = 0.8$. It is verified in Figure 7 that the proposed algorithm can obtain improved DOA estimation accuracy with a larger snapshots.

Simulation 5. Figure 8 shows DOA estimation performance of the proposed algorithm with varied M. Assume that $N = N' = 10$ and $J = J' = 200$ with $M'/M = 2/3$. It is attested

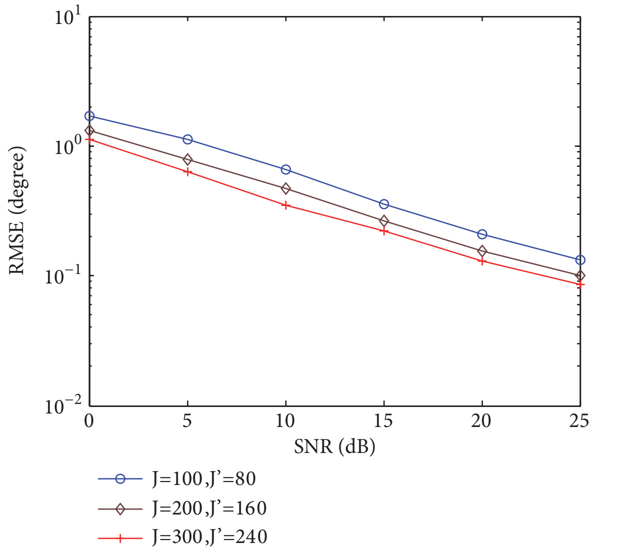


FIGURE 7: DOA estimation performance under different snapshots.

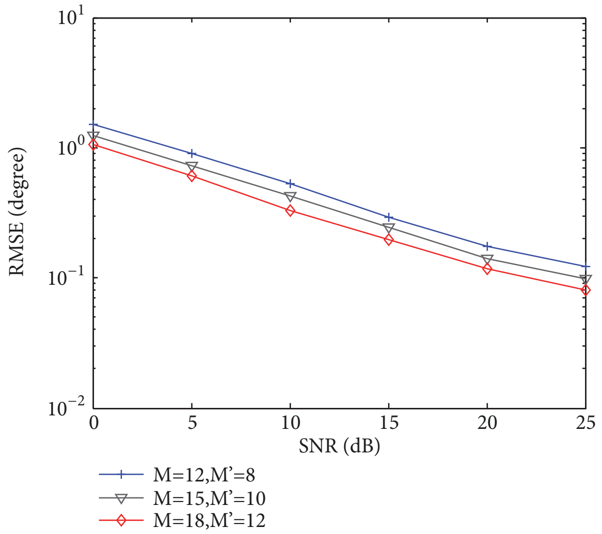


FIGURE 8: DOA estimation performance under different M.

by Figure 8 that the estimation performance of the proposed method gets better with the increasing M.

Simulation 6. The estimation performance of the proposed method versus N is illustrated in Figure 9, where $M = 15$, $J = 200$ with $M'/M = J'/J = N'/N = 0.8$. It is indicated obviously that the proposed method obtains improved performance with larger N.

6. Conclusions

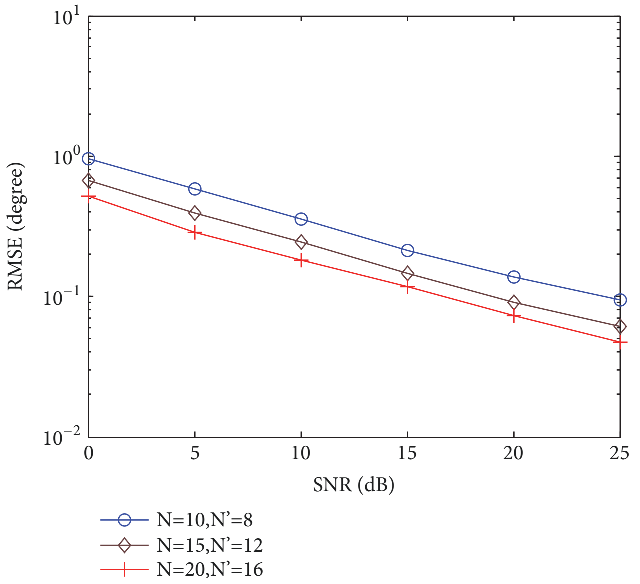


FIGURE 9: DOA estimation performance under different N.

We jointly utilize the CS theory and the PARALIND decomposition in this paper and propose a CS-PARALIND algorithm with a uniform rectangular array to extract the DOA estimates of coherent signals. The proposed algorithm can attain autopaired 2D-DOA estimates, and benefiting from the compression procedure, lower computational cost, and smaller demand for storage capacity can be achieved. Extensive simulations corroborate that the proposed approach obtains superior estimation performance compared to the traditional PARALIND algorithm and, significantly, outperforms the FBSS-PM and FBSS- ESPRIT algorithm.

Data Availability

The simulation data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by China NSF Grants (61371169, 61601167), the Fundamental Research Funds for the Central Universities (NT2019013), Graduate Innovative Base (laboratory) Open Funding of Nanjing University of Aeronautics and Astronautics (kfjj20170412), and Shandong Province NSF Grants (ZR2016FM43).

References

[1] X. Y. Yu and M. Q. Zhou, "A DOA estimation algorithm in OFDM radar-communication system for vehicular application," *Advanced Materials Research*, vol. 926-930, pp. 1853–1856, 2014.

[2] J. Li, X. Zhang, and H. Chen, "Improved two-dimensional DOA estimation algorithm for two-parallel uniform linear arrays using propagator method," *Signal Processing*, vol. 92, no. 12, pp. 3032–3038, 2012.

[3] X. F. Zhang, R. A. Cao, and M. Zhou, "Noncircular-PARAFAC for 2D-DOA estimation of noncircular signals in arbitrarily spaced acoustic vector-sensor array subjected to unknown locations," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, article no. 107, 2013.

[4] X. Zhang, L. Xu, D. Xu et al., "Direction of departure (DOD) and direction of arrival (DOA) estimation in MIMO radar with reduced-dimension MUSIC," *IEEE Communications Letters*, vol. 14, no. 12, pp. 1161–1163, 2010.

[5] R. Roy and T. Kailath, "ESPRIT—estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[6] F. Gao and A. B. Gershman, "A generalized ESPRIT approach to direction-of-arrival estimation," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 254–257, 2005.

[7] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[8] S. Marcos, A. Marsal, and M. Benidir, "The propagator method for source bearing estimation," *Signal Processing*, vol. 42, no. 2, pp. 121–138, 1995.

[9] T. Filik and T. E. Tuncer, "2-D paired direction-of-arrival angle estimation with two parallel uniform linear arrays," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 6, pp. 3269–3279, 2011.

[10] Y.-Y. Wang and S.-C. Huang, "An ESPRIT-based algorithm for 2D-DOA estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94.A, no. 9, pp. 1847–1850, 2011.

[11] C. P. Mathews, M. Haardt, and M. D. Zoltowski, "Performance analysis of closed-form, ESPRIT based 2-D angle estimator for rectangular arrays," *IEEE Signal Processing Letters*, vol. 3, no. 4, pp. 124–126, 1996.

[12] Y. Hua, "A pencil-MUSIC algorithm for finding two-dimensional angles and polarizations using crossed dipoles," *IEEE Transactions on Antennas and Propagation*, vol. 41, no. 3, pp. 370–376, 1993.

[13] W. Zheng, X. Zhang, and H. Zhai, "A generalized coprime planar array geometry for two-dimensional DOA estimation," *IEEE Communications Letters*, vol. 99, p. 1, 2017.

[14] Y.-Y. Dong, C.-X. Dong, Y.-T. Zhu, G.-Q. Zhao, and S.-Y. Liu, "Two-dimensional DOA estimation for L-shaped array with nested subarrays without pair matching," *IET Signal Processing*, vol. 10, no. 9, pp. 1112–1117, 2016.

[15] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Transactions on Signal Processing*, vol. 33, no. 4, pp. 806–811, 1985.

[16] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Transactions on Signal Processing*, vol. 37, no. 1, pp. 8–15, 1989.

[17] S. Qin, Y. D. Zhang, and M. G. Amin, "DOA estimation of mixed coherent and uncorrelated targets exploiting coprime MIMO radar," *Digital Signal Processing*, vol. 61, pp. 26–34, 2017.

[18] Y. Hu, Y. Liu, and X. Wang, "Doa estimation of coherent signals on coprime arrays exploiting fourth-order cumulants," *Sensors*, vol. 17, no. 4, article no. 682, 2017.

[19] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra and its Applications*, vol. 18, no. 2, pp. 95–138, 1977.

[20] R. Cao, X. Zhang, and C. Wang, "Reduced-dimensional PARAFAC-based algorithm for joint angle and doppler frequency estimation in monostatic MIMO radar," *Wireless Personal Communications*, vol. 80, no. 3, pp. 1231–1249, 2014.

[21] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[22] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.

[23] R. Bro, R. A. Harshman, N. D. Sidiropoulos, and M. E. Lundy, "Modeling multi-way data with linearly dependent loadings," *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 324–340, 2009.

[24] M. Bahram and R. Bro, "A novel strategy for solving matrix effect in three-way data using parallel profiles with linear dependencies," *Analytica Chimica Acta*, vol. 584, no. 2, pp. 397–402, 2007.

[25] C. Chen, X. F. Zhang, and D. Ben, "Coherent angle estimation in bistatic multi-input multi-output radar using parallel profile with linear dependencies decomposition," *IET Radar Sonar and Navigation*, vol. 7, no. 8, pp. 867–874, 2013.

[26] X. F. Zhang, M. Zhou, and J. F. Li, "A PARALIND decomposition-based coherent two-dimensional direction of arrival estimation algorithm for acoustic vector-sensor arrays," *Sensors*, vol. 13, no. 4, p. 5302, 2013.

[27] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[28] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[29] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[30] X. Yang, S. A. Shah, A. Ren et al., "Wandering pattern sensing at S-band," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 6, pp. 1863–1870, 2018.

[31] J.-Q. Wu, W. Zhu, and B. Chen, "Compressed sensing techniques for altitude estimation in multipath conditions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1891–1900, 2015.

[32] C.-R. Tsai and A.-Y. Wu, "Structured random compressed channel sensing for millimeter-wave large-scale antenna systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 19, pp. 5096–5110, 2018.

[33] N. D. Sidiropoulos and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 757–760, 2012.

[34] T. Nishimura, Y. Ogawa, and T. Ohgane, "DOA estimation by applying compressed sensing techniques," in *Proceedings of the 2014 IEEE International Workshop on Electromagnetics: Applications and Student Innovation Competition, IEEE iWEM 2014*, pp. 121-122, Japan, August 2014.

[35] C. Guang and L. Qi, "Compressed sensing-based angle estimation for noncircular sources in MIMO radar," in *Proceedings of the 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2014*, pp. 40–44, China, September 2014.

[36] H. X. Yu, X. F. Qiu, X. F. Zhang et al., “Two-dimensional direction of arrival (DOA) estimation for rectangular array via compressive sensing trilinear model,” *International Journal of Antennas & Propagation*, vol. 2015, 10 pages, 2016.

[37] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[38] X. F. Zhang, F. Wang, and H. W. Chen, *Theory and Application of Array Signal Processing*, National Defense Industry Press, Beijing, China, 2012.

[39] S. Li and X. F. Zhang, “Study on the compressed matrices in compressed sensing trilinear model,” *Applied Mechanics and Materials*, vol. 556-562, pp. 3380–3383, 2014.

[40] L. Xu, R. Wu, X. Zhang, and Z. Shi, “Joint two-dimensional DOA and frequency estimation for L-Shaped array via compressed sensing PARAFAC method,” *IEEE Access*, vol. 6, pp. 37204–37213, 2018.

[41] P. Stoica and A. Nehorai, “Performance study of conditional and unconditional direction-of-arrival estimation,” *IEEE Transactions on Signal Processing*, vol. 38, no. 10, pp. 1783–1795, 1990.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 7389306, 12 pages
https://doi.org/10.1155/2019/7389306

Research Article

DOA and Polarization Parameters Estimation by Exploiting Canonical Polyadic Decomposition of Tensors

Long Liu,[1] **Ling Wang**,[1,2] **Jian Xie**,[1,2] **and Zhaolin Zhang**[2]

[1] *School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China*
[2] *Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen 518057, China*

Correspondence should be addressed to Ling Wang; lingwang@nwpu.edu.cn

Received 21 October 2018; Revised 28 November 2018; Accepted 9 January 2019; Published 3 February 2019

Academic Editor: André de Almeida

A new algorithm to estimate the direction of arrival (DOA) and polarization parameters of signals impinging on an array with electromagnetic (EM) vector-sensors is presented by exploiting the canonical polyadic decomposition (CPD) of tensors. In addition to spatial and temporal diversities, further information from the polarization domain is considered and used in this paper. Estimation errors of these parameters are evaluated by the Cramér-Rao lower bound (CRB) benchmark, in the presence of additive white Gaussian noise (AWGN). The superiority of the proposed algorithm is shown by comparing with the derivative algorithms of MUSIC and ESPRIT. In the proposed algorithm, the parameters can be estimated by virtue of the diversities of the spatial and polarization belonging to the factor matrices, rather than the conventional subspace which is the foundation of MUSIC and ESPRIT. Additionally, the classical CPD algorithm based on Alternating Least Squares (ALS) is introduced to verify the efficacy of the proposed CPD algorithm. Results demonstrate that when the number of snapshots is greater than 50, the proposed algorithm requires a smaller number of snapshots to achieve a high level of performance, compared against the subspace-based algorithms and the ALS-based algorithm. Furthermore, in the matter of the array with a small number of sensors, the discovered advantage concerning the Root Mean Square Error (RMSE) in estimating the DOA and the polarization state of the signal is noteworthy.

1. Introduction

Signal parameters estimation is of great significance in many applications such as satellite navigation, wireless communication, radar, and sonar. An array of multiple sensors placed in different spatial locations is used to estimate the signal parameters, which mainly include spatial and polarization parameters of the signal. The spatial parameter refers to the direction of arrival (DOA) of the signal, while the polarization auxiliary angle and the polarization phase difference of the signal can be regarded as the polarization parameters. Among signal parameters estimation algorithms based on matrix operations, some depend on exhaustive search, e.g., multiple signal classification (MUSIC) [1], whereas others do not, e.g., estimation of the signal parameters via rotational invariance techniques (ESPRIT) [2] and root-MUSIC [3]. One of the common features of the matrix-based algorithms is that the matrix can only reflect the two-dimensional diversity of the desired signal. Some methods exploiting the signal-subspace embodied by MUSIC, ESPRIT and their derivative algorithms have been introduced in [4–12], in which the tensor-based data model has been combined in [10–12]. Based on the Higher-Order Singular Value Decomposition (HOSVD) of the measurement tensor, two algorithms of parameters estimation combined with ESPRIT have been developed in [10]. Similarly, in [11, 12], several tensor MUSIC methods based on HOSVD have been derived. The existing research results show that the performance of the subspace-based tensor method is improved compared to the subspace-based matrix method, and the computational complexity is higher in the tensor method than in the matrix method but of the same order. In [13], a greedy algorithm called randomized multiple candidate iterative hard thresholding for DOA estimation has been proposed. A set of potential candidates using the iterative hard thresholding algorithm are first obtained, and then the best candidate can be selected based on the a priori knowledge of the distribution of the signal and noise matrices.

The canonical polyadic decomposition (CPD) of the third-order tensor is a minimal decomposition into a sum of rank-1 tensors [14]. The uniqueness of CPD, which is often called essential uniqueness in engineering papers, has been extensively studied in the field of algebraic geometry. The property of uniqueness makes the CPD a basic tool for signal separation, and it is widely used in telecommunication, array processing, machine learning, etc. [15–20]. In [15], a novel multiuser receiver for joint symbol and channel estimation by capitalizing on a tensor modeling of the end-to-end system has been proposed. The multiple invariance sensor array processing has been linked to the uniqueness of CPD in [20]. It shows the uniqueness of single and multiple invariance ESPRIT, which stems from a deterministic decomposition of the third-order tensor signal model. Moreover, it provides the DOA of the source with less restrictions of signal stationarity than the aforementioned statistical approaches. In [21], a method of DOA estimation for seismic plane waves has been considered from a deterministic perspective using CPD. In addition to temporal and spatial information, the different propagation speed of waves is taken into account by using the multidimensional feature of the tensor. In [22], a CPD-based approach for distinguishing the signals with the same DOA and copolarized state has been proposed. However, the computational complexity of the algorithm is very high due to the complexity of the problem model.

The main purpose of this paper is to explore deterministic decomposition of the third-order tensor signal model by exploiting the uniqueness of CPD and then estimate the signal parameters. The polarization diversity of the signal is considered in this paper, in addition to the temporal and spatial sampling employed by the most current parameters estimation methods. Note that the proposed algorithm focus on not only the DOA estimation but also the estimation of polarization parameters delivered by the signal. The strength of our approach is its excellent performance compared against the traditional subspace-based algorithms such as MUSIC, HOSVD-MUSIC, and HOSVD-ESPRIT given shorter snapshots and the array with a small number of sensors.

The rest of this paper is organized as follows. Section 2 reviews the CPD prerequisites. The tensor signal model is derived in Section 3, and the introduction to traditional subspace-based algorithms is also provided. The CPD-based algorithm is proposed in Section 4, followed by the numerical simulations in Section 5. Finally, the last part presents our concluding remarks. The algebra notations involved in the paper are shown in Table 1.

Table 1: Algebra notations.

a	Scalar a
$\mathbf{a}$	Vector $\mathbf{a}$
$\mathbf{A}$	Matrix $\mathbf{A}$
$\mathscr{A}$	Tensor $\mathscr{A}$
$\mathbf{A}^*$	Conjugate of $\mathbf{A}$
$\mathbf{A}^T$	Transpose of $\mathbf{A}$
$\mathbf{A}^H$	Conjugate transpose of $\mathbf{A}$
$\mathscr{A} \circledcirc \mathscr{B}$	Outer product between $\mathscr{A}$ and $\mathscr{B}$
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product between $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A} \odot \mathbf{B}$	Khatri-Rao product between $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A} \circ \mathbf{B}$	Hadamard product between $\mathbf{A}$ and $\mathbf{B}$
$r_{\mathbf{A}}$	Rank of the matrix $\mathbf{A}$
$\mathbb{R}$	Real number field $\mathbb{R}$
$\mathbb{C}$	Complex number field $\mathbb{C}$

2. Prerequisites for CPD

Let the (i, j, k)-th entry $\mathscr{D}(i, j, k)$ of the third-order tensor $\mathscr{D} \in \mathbb{C}^{I\times J\times K}$ be symbolized by d_{ijk}. Herein, we define $r_{\mathscr{D}} = 1$; i.e., there exist three nonzero vectors $\mathbf{a} \in \mathbb{C}^I$, $\mathbf{b} \in \mathbb{C}^J$, and $\mathbf{c} \in \mathbb{C}^K$ such that $\mathscr{D} = \mathbf{a}\circledcirc\mathbf{b}\circledcirc\mathbf{c}$, which means that $d_{ijk} = a_i b_j c_k$ stands for all possible values of indices.

A third-order tensor $\mathscr{F} \in \mathbb{C}^{I\times J\times K}$ with general-rank can be expressed as a sum of rank-one items:

$$\mathscr{F} = \sum_{r=1}^{R} \mathscr{D}(r) \Longleftrightarrow \mathscr{F} = \sum_{r=1}^{R} \mathbf{a}_r \circledcirc \mathbf{b}_r \circledcirc \mathbf{c}_r, \qquad (1)$$

where $\mathbf{a}_r \in \mathbb{C}^I$, $\mathbf{b}_r \in \mathbb{C}^J$, and $\mathbf{c}_r \in \mathbb{C}^K$. The decomposition shown in (1) is called the polyadic decomposition (PD). Particularly, if the number of rank-one terms R is minimal, then R is the rank of $\mathscr{F}$ (i.e., $r_{\mathscr{F}} = R$), and (1) is defined as the CPD of $\mathscr{F}$.

Additionally, (1) can also be written as $\mathscr{F} = [\mathbf{A}, \mathbf{B}, \mathbf{C}]_R$, where the matrices $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_R] \in \mathbb{C}^{I\times R}$, $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_R] \in \mathbb{C}^{J\times R}$, and $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_R] \in \mathbb{C}^{K\times R}$ are defined as the first, second, and third factor matrix of $\mathscr{F}$, respectively. Obviously, the entries of $\mathscr{F}$ can be written as $f_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$. In addition, here is a potential scenario: the CPD of a general-rank tensor $\mathscr{F}$ is not unique. Nevertheless, we still call that the CPD of $\mathscr{F}$ is unique when it is only subject to the trivial indeterminacies, which include that the rank-one terms $\mathscr{D}(r)$ shown in (1) can be arbitrarily permuted and the vectors belonging to any rank-one term can be arbitrarily scaled. Similarly, the factor matrices for any two CPDs $\mathscr{F} = [\mathbf{A}, \mathbf{B}, \mathbf{C}]_R$ and $\mathscr{F} = [\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{C}}]_R$ coincide up to column permutation and scaling; for instance, $\overline{\mathbf{A}}$ can be obtained by scaling and permuting the columns of $\mathbf{A}$ (see [9] for more details). Herein, for reshaping the tensor $\mathscr{F}$ into a matrix $\mathbf{F}$, we introduce the following definition.

Definition 1. The matrix unfolding that yields $N' \in \{1, \ldots, N\}$ and the permutation $\mathfrak{T}\{1, \ldots, N\} = \{p_1, \ldots, p_N\}$, $p_n \in \{1, \ldots, N\}$ for the tensor $\mathscr{A} \in \mathbb{C}^{M_1\times M_2\times\ldots\times M_N}$ is defined as $\mathbf{A}_{(\mathfrak{T},N')} \in \mathbb{C}^{(M_{p_1} M_{p_2} \ldots M_{p_{N'}})\times(M_{p_{N'+1}} \ldots M_{p_N})}$. The (l_1, l_2)-th entry of $\mathscr{A}$ is given by

$$\mathbf{A}_{(\mathfrak{T},N')}(l_1, l_2) = a_{m_1 m_2 \ldots m_N}, \qquad (2)$$

where $1 \leqslant m_n \leqslant M_n$ for $n \in \{1, 2, \ldots, N\}$, and $l_1 = m_{p_{N'}} + \sum_{n=2}^{N'} (m_{p_{n-1}} - 1) \prod_{j=n}^{N'} M_{p_j}$, $l_2 = m_{p_N} + \sum_{n=N'+2}^{N} (m_{p_{n-1}} - 1) \prod_{j=n}^{N} M_{p_j}$. Assuming that $N' = 2$ and the permutation $\mathfrak{T}_1\{1,2,3\} = \{1,2,3\}$, the corresponding matrix unfolding of (1) can be expressed as

$$\mathbf{F}_{(\mathfrak{T}_1,2)} = \begin{bmatrix} \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_I \end{bmatrix} = \begin{bmatrix} \mathbf{B}\operatorname{diag}\left(\mathbf{a}^1\right)\mathbf{C}^T \\ \vdots \\ \mathbf{B}\operatorname{diag}\left(\mathbf{a}^I\right)\mathbf{C}^T \end{bmatrix} = (\mathbf{A} \odot \mathbf{B})\,\mathbf{C}^T, \qquad (3)$$

where $\mathbf{F}_{(\mathfrak{T}_1,2)} \in \mathbb{C}^{IJ \times K}$, $\mathbf{F}_i = [f_{ijk}]_{j,k=1}^{J,K}$ symbolizes the ith horizontal slice of $\mathcal{F}$, $\mathbf{a}^i$ denotes the ith row of $\mathbf{A}$, and diag$(\cdot)$ is the diagonalization operator. What is more, in the case of $N' = 2$, we can obtain the other matrix unfoldings:

$$\begin{aligned} \mathbf{F}_{(\mathfrak{T}_2,2)} &= (\mathbf{B} \odot \mathbf{C})\,\mathbf{A}^T, \\ \mathbf{F}_{(\mathfrak{T}_3,2)} &= (\mathbf{C} \odot \mathbf{A})\,\mathbf{B}^T, etc, \end{aligned} \qquad (4)$$

where $\mathfrak{T}_2\{1,2,3\} = \{2,3,1\}$ and $\mathfrak{T}_3\{1,2,3\} = \{3,1,2\}$. This paper mainly uses the algebraic algorithms described in [14, 23] to explore the application of CPD in signal parameters estimation. The key of the algebraic algorithms presented in [14, 21] is the following theorem, which has been derived in [24].

Theorem 2. *Define a third-order tensor* $\mathcal{F} = [\mathbf{A}, \mathbf{B}, \mathbf{C}]_R$, *and assume that the factor matrices* $\mathbf{B}$ *and* $\mathbf{C}$ *have full column rank and any two columns selected from the factor matrix* $\mathbf{A}$ *are linearly independent:*

$$\begin{aligned} r_{\mathbf{B}} &= r_{\mathbf{C}} = R, \\ k_{\mathbf{A}} &\geqslant 2, \end{aligned} \qquad (5)$$

where $k_{\mathbf{A}}$ *represents the largest number such that every subset with* $k_{\mathbf{A}}$ *columns of the factor matrix* $\mathbf{A}$ *is linearly independent. Then* $r_{\mathcal{F}} = R$*; the CPD of* $\mathcal{F}$ *can be computed by algebraic algorithms uniquely.*

One algebraic algorithm elaborated in [23] is adopted in this paper. Herein, we consider the particular case of $\mathcal{F} \in \mathbb{C}^{2 \times R \times R}$ that conforms to Theorem 2. By (3), the matrix unfolding of $\mathcal{F}$ can be expressed as

$$\mathbf{F}_{(\mathfrak{T}_1,2)} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}\operatorname{diag}\left(\mathbf{a}^1\right)\mathbf{C}^T \\ \mathbf{B}\operatorname{diag}\left(\mathbf{a}^2\right)\mathbf{C}^T \end{bmatrix}, \qquad (6)$$

which obtains the equations that

$$\begin{aligned} \mathbf{F}_1\mathbf{F}_2^{-1} &= \mathbf{B}\operatorname{diag}\left(\mathbf{a}^1\right)\operatorname{diag}\left(\mathbf{a}^2\right)^{-1}\mathbf{B}^{-1}, \\ \mathbf{F}_1^T\mathbf{F}_2^{-T} &= \mathbf{C}\underbrace{\operatorname{diag}\left(\mathbf{a}^1\right)\operatorname{diag}\left(\mathbf{a}^2\right)^{-1}}_{\mathbf{D}}\mathbf{C}^{-1}. \end{aligned} \qquad (7)$$

Since $k_{\mathbf{A}} \geqslant 2$, the diagonal elements of the diagonal matrix $\mathbf{D}$ are distinct. Therefore, the factor matrices $\mathbf{B}$ and $\mathbf{C}$ can be uniquely recovered by the eigenvalue decomposition (EVD) of (7), and then the matrix $\mathbf{A}$ can be easily identified from (6). However, the factor matrices are not necessarily square matrices in reality. Hence, we need to consider the generalized inverse operation of the matrices to obtain the CPD of the tensor.

Herein, in order to avoid ambiguity in signal modeling, some of the main assumptions are stated as follows.

A1. Completely polarized wave: the horizontal and vertical electric components located in the wavefront of the transverse electromagnetic (TEM) wave are completely correlated.

A2. Far-field assumption: we assume that the distance between the source and the receiving array is much larger than the array aperture. Thus, the source of the signal can be considered point-like. Moreover, the wavefront of the TEM wave can be approximated as a plane at the sensor level.

A3. Narrow-band assumption: the reciprocal of the signal bandwidth is much greater than the maximum propagation time of the signal passing through the array aperture. Typically, the signal received by an array is a varying complex envelope, which is modulated at a carrier with high frequency (HF). In order to facilitate the digital signal processing, the intermediate frequency (IF) or baseband signal is usually wanted. Hence, the analog down-conversion for the received HF signal is a more common practice.

3. Signal Modeling

This section mainly focuses on the three-dimensional feature, i.e., the vector-sensor enjoys in spatial, polarization, and time diversities and then establishes the third-order tensor-based signal model. In addition, conventional matrix-based solutions are introduced briefly. Firstly, the diversities involved in this paper are shown as follows.

Time Diversity. The baseband analog signal obtained by down-conversion is a time-dependent function. Moreover, the discrete-time signal can be obtained by sampling the baseband analog signal in time domain. It is assumed that R signals with the complex amplitudes $\{s_r(k), r = 1, \ldots, R\}$ impinge on the vector-sensor array. Thus, the time diversity can be formulized as

$$f^{(1)}(k) = \sum_{r=1}^{R} s_r(k), \qquad (8)$$

where $k = 1, \ldots, K$, represents the snapshot at the instant t_k.

Spatial Diversity. The spatial sampling of the array reflects the spatial diversity. Consider a vector-sensor array composed of L EM vector-sensors located at $\mathbf{b}_l \in \mathbb{R}^3$ for $l = 1, \ldots, L$. Combined with the time diversity, the signal received by the array can be expressed as

$$f^{(2)}(k, l) = \sum_{r=1}^{R} e^{-i2\pi(\mathbf{b}_l^T \boldsymbol{\epsilon}_r/\lambda)} s_r(k), \qquad (9)$$

where the symbol λ denotes the wave length of the signal impinged on the vector-sensor array. Moreover, $\boldsymbol{\epsilon}_r = -\left[\sin\phi_r \cos\theta_r \;\; \sin\phi_r \sin\theta_r \;\; \cos\phi_r\right]^T$ denotes the incident direction of the rth signal source associated with the azimuth angle $\theta_r \in [0, 2\pi)$ and the elevation angle $\phi_r \in [0, \pi]$.

Polarization Diversity. The EM vector-sensor is usually composed of multiple magnetic and electric short dipoles in a common spatial point. Hence, the EM vector-sensor is capable of measuring multiple electromagnetic components with the number of not less than one. For example, a three-dipole sensor can measure the three components of the electric field in three perpendicular directions. Furthermore, the complete EM vector-sensor has the ability to obtain all electromagnetic components (i.e., three electric components and three magnetic components). In particular, if the vector-sensor array is composed of a monopole, the orientations of the monopoles cannot be the same; otherwise, the array will degrade into a scalar-array. We formulate the polarization diversity combined with the time diversity,

$$f^{(3)}(k, j) = \sum_{r=1}^{R} \mathbf{a}_j^p(\Psi_r)\, s_r(k), \tag{10}$$

where $j = 1, \ldots, J$, J is the number of component measured by an EM vector-sensor, $J = 3$ for three-dipoles, and $J = 6$ for complete EM vector-sensor. Moreover, $\mathbf{a}_j^p(\Psi_r)$ represents the jth component of the polarization steering vector $\mathbf{a}^p(\Psi_r)$, and the symbol $\Psi_r = (\theta_r, \phi_r, \gamma_r, \eta_r)$ denotes the spatial-polarization parameter of the rth signal source with respect to the polarization auxiliary angle $\gamma_r \in [0, \pi/2]$ and the polarization phase difference $\eta_r \in [-\pi, \pi)$.

3.1. Tensor-Based Signal Model. Consider the diversities mentioned above, and assume that all the signals are cofrequency and incoherent. The spatial steering vector of the signal with DOA (θ, ϕ) is expressed as

$$\mathbf{a}^s(\theta, \phi) = \left[e^{-i2\pi(\mathbf{b}_1^T \boldsymbol{\epsilon}/\lambda)}, \ldots, e^{-i2\pi(\mathbf{b}_L^T \boldsymbol{\epsilon}/\lambda)}\right]^T. \tag{11}$$

The polarization steering vector of the signal with spatial-polarization parameter $\Psi = (\theta, \phi, \gamma, \eta)$ is given by

$$\begin{aligned}\mathbf{a}^p(\Psi) &= \mathfrak{B}\Xi_{\theta,\phi}\boldsymbol{h}_{\gamma,\eta}, \Xi_{\theta,\phi} \\ &= \begin{bmatrix} -\sin\theta & \cos\phi\cos\theta \\ \cos\theta & \cos\phi\sin\theta \\ 0 & -\sin\phi \\ \cos\phi\cos\theta & \sin\theta \\ \cos\phi\sin\theta & -\cos\theta \\ -\sin\phi & 0 \end{bmatrix},\end{aligned} \tag{12}$$

where $\boldsymbol{h}_{\gamma,\eta} = [\cos\gamma, \sin\gamma e^{i\eta}]^T$ and $\mathfrak{B} \in \mathbb{R}^{J\times 6}$ represents the polarization selection matrix of the EM vector-sensor. Particularly, $\mathfrak{B} = \mathbf{I}_{6\times 6}$ represents the complete EM vector-sensor, and $\mathfrak{B} = [\mathbf{I}_{3\times 3}, \mathbf{0}_{3\times 3}]$ symbolizes the three-dipole sensor, where $\mathbf{I}$ denotes the identity matrix. Additionally, the first three rows of the matrix $\Xi_{\theta,\phi}$ represent the projections of the electric field component of the incident signal with DOA (θ, ϕ) in the three axial directions of the coordinate system, and the last three rows of the matrix $\Xi_{\theta,\phi}$ represent the projections of the magnetic field component in the three axial directions. The spatial-polarization steering vector can be further defined as

$$\mathbf{a}(\Psi) = \mathbf{a}^s(\theta, \phi) \otimes \mathbf{a}^p(\Psi). \tag{13}$$

Hereby, let $\mathbf{n}(k)$ denote the spatial white noise at the instant t_k, which is additive and Gaussian complex circular. The vector-output of the array at the instant t_k can be expressed as

$$\boldsymbol{y}(k) = \sum_{r=1}^{R} \mathbf{a}(\Psi_r)\, s_r(k) + \mathbf{n}(k). \tag{14}$$

Considering that the multidimensional nature of the tensor just caters to the multidomain diversities of the signal, the following spatial-polarization steering matrix can be obtained by performing the outer product operations of (12) and (11):

$$\widetilde{\mathbf{A}}(\Psi) = \mathbf{a}^p(\Psi) \circledcirc \mathbf{a}^s(\theta, \phi). \tag{15}$$

where $\widetilde{\mathbf{A}}(\Psi) \in \mathbb{C}^{J\times L}$. Let $\mathbf{s} = [s(1), \ldots, s(K)]$ denote the consecutive K snapshots of the signal; the tensor-output of K snapshots can be modeled as

$$\begin{aligned}\mathscr{Y} &= \sum_{r=1}^{R} \widetilde{\mathbf{A}}(\Psi_r) \circledcirc \mathbf{s}_r + \mathscr{N} \\ &= \sum_{r=1}^{R} \mathbf{a}^p(\Psi_r) \circledcirc \mathbf{a}^s(\theta_r, \phi_r) \circledcirc \mathbf{s}_r + \mathscr{N},\end{aligned} \tag{16}$$

where $\mathscr{Y} \in \mathbb{C}^{J\times L\times K}$. The tensor noise is expressed as $\mathscr{N} \in \mathbb{C}^{J\times L\times K}$, which is yielded by the tensorization with respect to the noise matrix $\mathbf{N} = [\mathbf{n}(1), \ldots, \mathbf{n}(k)]$.

3.2. Traditional Subspace-Based Algorithms. Subspace decomposition methods with respect to EVD and SVD are the mainstream of MUSIC, ESPRIT, and their derivative algorithms. Herein, we introduce the matrix-based MUSIC algorithm and the tensor-based MUSIC algorithm based on HOSVD. The covariance matrix for K snapshots corresponding to (14) can be given by

$$\widehat{\mathbf{R}}_{\boldsymbol{yy}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}(k)\, \boldsymbol{y}^H(k), \tag{17}$$

where $\widehat{\mathbf{R}}_{\boldsymbol{yy}} \in \mathbb{C}^{LJ\times LJ}$. Then the eigenvector corresponding to the small eigenvalues of $\widehat{\mathbf{R}}_{\boldsymbol{yy}}$ can be written in a matrix form $\widehat{\mathbf{U}}_n$, where $\widehat{\mathbf{U}}_n \in \mathbb{C}^{LJ\times(LJ-R)}$. Moreover, according to (11) and (12), we need to construct a search matrix:

$$\mathbf{D}_{\theta,\phi} = \mathbf{a}^s(\theta, \phi) \otimes \left(\mathfrak{B}\Xi_{\theta,\phi}\right). \tag{18}$$

The solution of the matrix-based MUSIC algorithm can be defined as

$$\{\theta_r, \phi_r\} = \arg\max_{\theta,\phi} \det{}^{-1}\left\{\mathbf{D}_{\theta,\phi}^H \widehat{\mathbf{U}}_n \widehat{\mathbf{U}}_n^H \mathbf{D}_{\theta,\phi}\right\}, \tag{19}$$

where $r = 1, \ldots, R$. The estimation of the polarization parameter is detailed in Appendix A. For the tensor-based MUSIC algorithm based on HOSVD [11], the following signal model needs to be constructed:

$$\mathbf{Y}(k) = \sum_{r=1}^{R} \widetilde{\mathbf{A}}(\Psi_r)\,\mathbf{s}_r + \mathbf{N}, \tag{20}$$

where $\mathbf{Y}(k) \in \mathbb{C}^{J \times L}$. The covariance tensor for K snapshots is expressed as

$$\widehat{\mathscr{R}}_{\mathbf{YY}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{Y}(k) \circledcirc \mathbf{Y}^{*}(k), \tag{21}$$

where $\widehat{\mathscr{R}}_{\mathbf{YY}} \in \mathbb{C}^{J \times L \times J \times L}$. Then, $\widehat{\mathscr{R}}_{\mathbf{YY}}$ is decomposed using the HOSVD procedure as

$$\widehat{\mathscr{R}}_{\mathbf{YY}} = \widehat{\mathscr{K}} \times_1 \widehat{\mathbf{U}}^{(1)} \times_2 \widehat{\mathbf{U}}^{(2)} \times_3 \widehat{\mathbf{U}}^{(3)} \times_4 \widehat{\mathbf{U}}^{(4)}, \tag{22}$$

where $\times_n$ represents n-mode product, $\widehat{\mathscr{K}}$ is the core tensor, and $\widehat{\mathbf{U}}^{(n)}$ can be obtained by SVD of n-mode matrix unfolding of tensor $\widehat{\mathscr{R}}_{\mathbf{YY}}$. Finally the parameters of the sources are obtained by maximizing the following criterion:

$$\begin{aligned}\{\theta_r, \phi_r\, \gamma_r, \eta_r\} \\ = \arg\max_{\Psi} \left\| \widetilde{\mathbf{A}}(\Psi) \times_1 \widehat{\mathbf{U}}_0^{(1)} \widehat{\mathbf{U}}_0^{(1)H} \times_2 \widehat{\mathbf{U}}_0^{(2)} \widehat{\mathbf{U}}_0^{(2)H} \right\|_F^{-1},\end{aligned} \tag{23}$$

where $\widehat{\mathbf{U}}_0^{(n)}$ is a truncated form of $\widehat{\mathbf{U}}^{(n)}$ (see [11] for specific truncation methods). In addition, the HOSVD-ESPRIT algorithm depending on translation invariant property of the array is elaborated in Appendix B.

4. CPD-Based Algorithm

We hereby execute the CPD of the tensor-based signal model to obtain the factor matrices, which are then used to extract the parameters of the signals. Let $\mathscr{Y} = [\mathbf{A}_p, \mathbf{A}_s, \mathbf{S}]_R$, where $\mathbf{A}_p = [\mathbf{a}^p(\Psi_1), \ldots, \mathbf{a}^p(\Psi_R)] \in \mathbb{C}^{J \times R}$, $\mathbf{A}_s = [\mathbf{a}^s(\theta_1, \phi_1), \ldots, \mathbf{a}^s(\theta_R, \phi_R)] \in \mathbb{C}^{L \times R}$, and $\mathbf{S} = [\mathbf{s_1}, \ldots, \mathbf{s_R}] \in \mathbb{C}^{K \times R}$. Assume that at least two factor matrices of $\mathscr{Y}$ have full column rank and the k-rank of the rest factor matrix is not less than 2, which are consistent with the Theorem 2. Moreover, there are conditions in which J, L, and K are not less than R, respectively, and this cannot be ignored here to ensure the uniqueness of the factor matrices.

4.1. Computation of CPD. As is well known, CPD of a third-order tensor can be achieved by the classical Alternating Least Squares (ALS) algorithm, which is detailed in Appendix C. Differently, we hereby propose an EVD-based algebraic method to achieve CPD of tensor. The tensor-based signal model (16) can be unfolded as

$$\mathbf{Y}_{(\mathfrak{X}_1,2)} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_J \end{bmatrix} = \begin{bmatrix} \mathbf{A}_s \operatorname{diag}\left(\mathbf{a}_p^1\right) \mathbf{S}^T \\ \vdots \\ \mathbf{A}_s \operatorname{diag}\left(\mathbf{a}_p^J\right) \mathbf{S}^T \end{bmatrix}, \tag{24}$$

where $\mathbf{a}_p^j$ denotes the jth row of the $\mathbf{A}_p$, $j = 1, \ldots, J$. According to (24), the following relationship can be easily derived:

$$\begin{aligned} \mathbf{Y}_{j_1} \mathbf{Y}_{j_2}^{\dagger} &= \mathbf{A}_s \operatorname{diag}\left(\mathbf{a}_p^{j_1}\right) \operatorname{diag}\left(\mathbf{a}_p^{j_2}\right)^{-1} \mathbf{A}_s^{\dagger}, \\ \mathbf{Y}_{j_1}^T \left(\mathbf{Y}_{j_2}^T\right)^{\dagger} &= \mathbf{S} \operatorname{diag}\left(\mathbf{a}_p^{j_1}\right) \operatorname{diag}\left(\mathbf{a}_p^{j_2}\right)^{-1} \mathbf{S}^{\dagger}, \end{aligned} \tag{25}$$

where $j_1 \neq j_2$, $j_1, j_2 = 1, \ldots, J$, and $(\cdot)^{\dagger}$ denotes the Moore-Penrose inverse. From (24), we have $\mathbf{Y}_j = \mathbf{A}_s \operatorname{diag}(\mathbf{a}_p^j)\mathbf{S}^T$, and $j = 1, \ldots, J$. Hence, for any j, $\operatorname{diag}(\mathbf{a}_p^j) = \mathbf{A}_s^{\dagger}\mathbf{Y}_j(\mathbf{S}^T)^{\dagger}$. The features of the CPD are demonstrated as follows:

(1) The columns of the factor matrix $\mathbf{A}_s$ are the eigenvectors of $\mathbf{Y}_{j_1}\mathbf{Y}_{j_2}^{\dagger}$. The corresponding eigenvalues are the first R larger eigenvalues, which are arranged in a descending order.

(2) The eigenvectors corresponding to the R-large eigenvalues of $\mathbf{Y}_{j_1}^T(\mathbf{Y}_{j_2}^T)^{\dagger}$ constitute the factor matrix $\mathbf{S}$.

(3) The rows of factor matrix $\mathbf{A}_p$ can be recovered thanks to the relationship $\operatorname{diag}(\mathbf{a}_p^j) = \mathbf{A}_s^{\dagger}\mathbf{Y}_j(\mathbf{S}^T)^{\dagger}$.

Obviously, when $J > 2$, there are various combinations (j_1, j_2) that can be used for EVD. In order to make full use of the statistical properties of (24), we select C_J^2 possible combinations for EVD to obtain $\mathbf{A}_s^{j'}$, $\mathbf{S}^{j'}$, where $j' = 1, \ldots, C_J^2$. We follow the convention that the (j_1, j_2)-tuples are ordered lexicographically: the j_1'th tuple (j_1, j_2) is preceding the j_2'th tuple (j_3, j_4) if and only if $j_1 \leqslant j_3$, $j_2 \leqslant j_4$. According to the trivial indeterminacies aforementioned for CPD uniqueness, we know that $\mathbf{A}_s^{j_1'}$ and $\mathbf{A}_s^{j_2'}$, $j_1' \neq j_2'$, coincide up to column permutation and scaling. In order to achieve the addition of multiple sets of results, we need to match the multiple sets of results. That is to say, the eigenvectors in each factor matrix obtained by the EVD need to be arranged in a descending order according to the corresponding eigenvalues, and all the eigenvectors need to be normalized. After the matching process, $\widetilde{\mathbf{A}}_s^{j'}$ and $\widetilde{\mathbf{S}}^{j'}$ can be obtained, and then the following statistical operations are performed:

$$\begin{aligned} \mathbf{A}_s &= \frac{1}{C_J^2} \sum_{j'=1}^{C_L^2} \widetilde{\mathbf{A}}_s^{j'}, \\ \mathbf{S} &= \frac{1}{C_J^2} \sum_{j'=1}^{C_L^2} \widetilde{\mathbf{S}}^{j'}. \end{aligned} \tag{26}$$

Next, the factor matrix $\mathbf{A}_p$ can be recovered according to the results of (26).

4.2. Parameters Estimation from the Factor Matrices. From the aforementioned tensor-based signal model, we can see that the spatial-polarization parameters of signals can be reflected by the factor matrices $\mathbf{A}_p$ and the factor matrix $\mathbf{A}_s$ that only contains the spatial parameters of signals. Hence, once the factor matrices are obtained, we first need to extract the spatial parameters from $\mathbf{A}_s$.

4.2.1. Estimation of Spatial Parameters. Spatial DOA (θ_r, ϕ_r) of the rth signal impinging on the array can be calculated directly by solving the following equations:

$$\begin{aligned} \left[\mathbf{A}_s\right]_{1r} &= e^{-i2\pi(\mathbf{b}_1^T \boldsymbol{\epsilon}_r/\lambda)} \\ &\vdots \\ \left[\mathbf{A}_s\right]_{Lr} &= e^{-i2\pi(\mathbf{b}_L^T \boldsymbol{\epsilon}_r/\lambda)} \end{aligned} \tag{27}$$

Obviously, any two equations belonging to (27) can be used to derive the DOA of the rth signal. Therefore, we can get C_L^2 sets of solutions from (27). Improve the accuracy of the estimated parameters by averaging multiple sets of solutions:

$$(\theta_r, \phi_r) = \frac{1}{C_L^2} \sum_{m=1}^{C_L^2} (\theta_r, \phi_r)_m, \tag{28}$$

where (θ_r, ϕ_r) represents the solution of the mth subequations $\{[\mathbf{A}_s]_{l_1 r}, [\mathbf{A}_s]_{l_2 r}\}$. We follow the convention that the (l_1, l_2)-tuples are ordered lexicographically: the mth tuple (l_1, l_2) is preceding the nth tuple (l_1', l_2') if and only if $l_1 \leqslant l_1'$, $l_2 \leqslant l_2'$. Furthermore, we can construct the optimization problem as follows to estimate the DOAs of the signals:

$$\{\theta_r, \phi_r\} = \arg\min_{\theta,\phi} \left\|\boldsymbol{\zeta}_s^r \circ \mathbf{a}^s(\theta, \phi) - \mathbf{a}_s^{:,r}\right\|^2, \tag{29}$$

where $\mathbf{a}_s^{:,r}$ represents the rth column of the factor matrix $\mathbf{A}_s$ obtained by CPD, $\boldsymbol{\zeta}_s^r$ represents the amplitude vector of $\mathbf{a}_s^{:,r}$, and the operator $\|\cdot\|$ denotes the 2-norm. The solution method to (29) and its performance can be used as a further work, and this paper discusses only the performance based on the solution of (28).

4.2.2. Estimation of Polarization Parameters. Once the estimation of DOA is obtained, the polarization parameters (γ_r, η_r) of the rth signal can be extracted from

$$\mathbf{a}_p^{:,r} = \mathfrak{B}\Xi_{\theta_r,\phi_r}\left[\cos\gamma_r, \sin\gamma_r e^{i\eta_r}\right]^T, \tag{30}$$

where $\mathbf{a}_p^{:,r}$ represents the rth column of the factor matrix $\mathbf{A}_p$ obtained by CPD. Hereby, the estimated value of $\mathbf{h}_{\gamma_r,\eta_r}$ can be expressed as $\widehat{\mathbf{h}}_{\gamma_r,\eta_r} = \Xi_{\theta_r,\phi_r}^{-1}\mathfrak{B}^{-1}\mathbf{a}_p^{:,r}$, and the estimated values of the polarization parameters can be obtained by

$$\begin{aligned} \gamma_r &= \arctan\left(\left|\frac{\left[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}\right]_2}{\left[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}\right]_1}\right|\right), \\ \eta_r &= \arg\left(\left(\frac{\left[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}\right]_2}{\left[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}\right]_1}\right)\right), \end{aligned} \tag{31}$$

where $\arctan(\cdot)$ is the arctangent operator and the operator $\arg(\cdot)$ denotes phase acquisition.

Obviously, the estimation accuracy of the DOA only depends on the factor matrix $\mathbf{A}_s$ obtained by CPD, and the estimation accuracy of the polarization parameters is affected by both the estimation accuracy of the DOA and the matrix $\mathbf{A}_p$ obtained by CPD. It is well known that, in order to satisfy the validity of the parameters estimation, the algorithm must follow the premise of $L > R$. According to [14], there exists Kruskal bound ensuring the CPD uniqueness $R < (J + L + K - 2)/2$. With the noise present, the accuracy of the CPD is severely impaired, when R is very close to the bound. However, as $(J+L+K-2)/2$ increases, the dependence of the CPD accuracy on $(J+L+K-2)/2$ will rapidly weaken. In addition, as can be seen from (26) and (28), the proposed algorithm can obtain multiple sets of parameters estimation results in each CPD and perform statistical operations on these results. However, what is different is that the subspace-based methods requires sufficient statistical information, which is usually obtained by increasing the number of sensors or snapshots, to ensure the accuracy of the parameters estimation, even if the number of the parameters to be estimated is very small. We, therefore, conclude that the proposed algorithm can achieve relatively desirable performance by using a smaller number of snapshots as well as a smaller number of sensors compared with the subspace-based methods. The subsequent simulation results also justify the above theoretical analysis.

4.3. Computational Complexity. The parameters estimation algorithm using EVD-based CPD of tensors is summarized in Algorithm 1, which is easy to implement by row-by-row access. Note that the computational complexity scales linearly to the number of loops, since the procedure of the algorithm runs serially.

In fact, the EVDs of (25) follow from the generalized EVDs (GEVDs) of the matrix pencils $(\mathbf{Y}_{j_1}, \mathbf{Y}_{j_2})$ and $(\mathbf{Y}_{j_1}^T, \mathbf{Y}_{j_2}^T)$. In order to evaluate the complexity of the proposed algorithm, we first need to know the number of multiplications required by GEVD. An effective GEVD solution is given by the method of **QZ** iteration [25, pp.413-414], in which $pLKR$ multiplications are required for two $L \times K$ matrices with rank-R, where the constant p depends on the selected iterative strategy, e.g., for Hessenberg-Triangular Reduction, $p = 8$. The Moore-Penrose inverse of the $M \times N$ matrix or $N \times M$ matrix involves $4MN^2 + 8N^3$ multiplications, where $M > N$. Additional $J \times (RLK + R^2K)$ multiplications are also required to recover the factor matrix $\mathbf{A}_p$. Therefore, the computational complexity of the EVD-based CPD does not exceed $(pC_J^2 + 8 + J)K \cdot [O(L^2)]$, since the premise of $L, J, K > R$ has been established. Furthermore, in order to compare the computational complexity for the proposed algorithm, we analyze the required number of multiplications for subspace-based methods. For the aforementioned matrix-based method, the computational complexity of obtaining subspaces is approximately $(L+K)\cdot[O(L^2)]$. Considering the operation process of HOSVD based on orthogonal iterations [25, pp.454-455], the computational complexity of subspace for tensor model is about $(KJ^2+8J^2+pJ)\cdot[O(L^2)]$. Obviously, the conclusion that the computational complexity of the CPD algorithm is lower than that of the tensor-based HOSVD methods can be obtained through the above analysis. Additionally, for the extraction process of the parameters, the

```
Require: $\mathscr{Y} \in \mathbb{C}^{J\times L\times K}$, $L, J, K, R$.
Reshape the matrcies $\mathbf{Y}_1, \ldots, \mathbf{Y}_J \in \mathbb{C}^{L\times K}$ by unfolding the tensor $\mathscr{Y}$;
For $j' = 1$ to $C_J^2$ do
    Calculate $\mathbf{Y}_{j_1}\mathbf{Y}_{j_2}^{\dagger}, \mathbf{Y}_{j_1}^{T}(\mathbf{Y}_{j_2}^{T})^{\dagger}$;
    Obtain $\mathbf{A}_s^{j'}$ by executing the EVD of $\mathbf{Y}_{j_1}\mathbf{Y}_{j_2}^{\dagger}$;
    Obtain $\mathbf{S}^{j'}$ by executing the EVD of $\mathbf{Y}_{j_1}^{T}(\mathbf{Y}_{j_2}^{T})^{\dagger}$;
    Obtain $\widetilde{\mathbf{A}}_s^{j'}$ and $\widetilde{\mathbf{S}}^{j'}$ by matching processing.
End for
$\mathbf{A}_s = \frac{1}{C_J^2}\sum_{j'=1}^{C_L^2}\widetilde{\mathbf{A}}_s^{j'}, \quad \mathbf{S} = \frac{1}{C_J^2}\sum_{j'=1}^{C_L^2}\widetilde{\mathbf{S}}^{j'}$;
Recover $\mathbf{A}_p$ according to $\mathrm{diag}(\mathbf{a}_p^{j}) = \mathbf{A}_s^{\dagger}\mathbf{Y}_j(\mathbf{S}^{T})^{\dagger}$;
For $r = 1$ to $R$ do
    For $m = 1$ to $C_L^2$ do
        Calculate $(\theta_r, \phi_r)_m$ by solving equations $\{[\mathbf{A}_s]_{l_1 r}, [\mathbf{A}_s]_{l_2 r}\}$;
    End for
    $(\theta_r, \phi_r) = \frac{1}{C_L^2}\sum_{m=1}^{C_L^2}(\theta_r, \phi_r)_m$;
    $\widehat{\mathbf{h}}_{\gamma_r,\eta_r} = \Xi_{\theta_r,\phi_r}^{-1}\mathfrak{B}^{-1}\mathbf{a}_p^{:,r}$;
    $\gamma_r = \arctan\left(\left|\frac{[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}]_2}{[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}]_1}\right|\right), \ \eta_r = \arg\left(\left(\frac{[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}]_2}{[\widehat{\mathbf{h}}_{\gamma_r,\eta_r}]_1}\right)\right)$.
End for
Ensure: $\Psi_r = (\theta_r, \phi_r, \gamma_r, \eta_r)$, $r = 1, \ldots, R$.
```

Algorithm 1: EVD-based CPD for parameters estimation.

computational complexity of the search strategies is generally much higher than that of the equation-solving methods, under the premise of ensuring the estimation accuracy. Therefore, the computational complexity of the proposed algorithm is lower than that of the search-based methods.

Additionally, according to the procedure of the ALS-based CPD described in Appendix C, the computational complexity of the factor matrix **S** is not less than $(5 + L + K)\cdot[O(R^3)]$, since the premise of $L, J, K > R$ has been established. Similarly, the computational complexity of the factor matrices $\mathbf{A}_p$ and $\mathbf{A}_s$ is not less than $(5 + 2K)\cdot[O(R^3)]$. Therefore, the computational complexity of the ALS-based CPD is about $(15 + L + 5K)G\cdot[O(R^3)]$, where G represents the number of iterations. We use the Matlab 2016a to implement the after-mentioned simulations on a computer with Intel Core i5 CPU6200U 2.3GHz and 8GB memory running Windows 10. From the perspective of computing time, the time required for ALS-based CPD is about twice that of EVD-based CPD to obtain the subsequent simulation results.

5. Simulation Results

The superiorities of the proposed algorithm will be demonstrated by the results of numerical simulations in this section. We consider the following scenario with a uniform linear array (ULA) separated by half wavelength. The array consists of complete EM vector-sensors, and two signals with the polarization parameters $(\gamma_1, \eta_1) = (10^\circ, 90^\circ)$ and $(\gamma_2, \eta_2) = (30^\circ, 90^\circ)$ impinge on the array from $(\theta_1, \phi_1) = (50^\circ, 90^\circ)$ and $(\theta_2, \phi_2) = (120^\circ, 90^\circ)$, respectively. Both signals satisfy the aforementioned assumptions **A1** to **A3**. The modulation modes of the signals are BPSK and QBPSK, respectively, and the symbol transmission rate of both signals is 320*B*. Additionally, modulation frequency and sampling frequency are 16kHz and 32kHz, respectively. The noise component is assumed to be zero-mean additive white Gaussian noise. Performance of the EVD-based CPD algorithm is compared with three other subspace-based algorithms, MUSIC, HOSVD-MUSIC, HOSVD-ESPRIT, and the ALS-based CPD algorithm. The performances of the algorithms are evaluated by the Root Mean Square Error (RMSE):

$$\mathrm{RMSE}(\mu) = \sqrt{\frac{1}{NR}\sum_{n=1}^{N}\sum_{r=1}^{R}(\widehat{\mu}_{rn} - \mu_r)^2}, \tag{32}$$

where μ_r denotes one of the parameters $\{\theta_r, \phi_r, \gamma_r, \eta_r\}$, and $\widehat{\mu}_{rn}$ is the estimation of μ_r in the nth trial. Throughout all simulations, results are averaged by 500 Monte-Carlo trails and compared with the Cramér-Rao lower bound (CRB) benchmark, which is described in Appendix D.

Example 3. Assume that the number of complete EM vector-sensors is 4, and the signal-to-noise ratio (SNR) is 30dB. The proposed EVD-based CPD algorithm is compared with the subspace-based algorithms and the ALS-based CPD algorithm in terms of the DOA RMSE versus the different number of snapshots K. Figure 1 illustrates a superior performance of the proposed EVD-based CPD algorithm with respect to the other algorithms when the number of snapshots is larger than 50. It is worth noting that, when the number of snapshots is

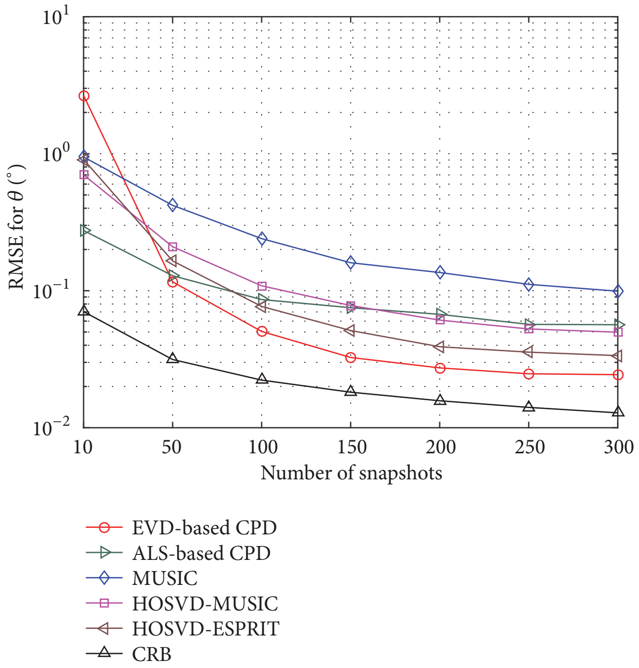


Figure 1: DOA RMSE versus K for $L = 4$ and SNR=30dB.

greater than 100, the performance change of the ALS-based CPD algorithm with the increase of snapshots is negligible. In the case where the number of snapshots is extremely small, the main reason for the poor performance of the EVD-based CPD algorithm is that the number of the signals to be estimated is close to the uniqueness condition for CPD.

Example 4. Scenarios with various numbers of sensors ($L \in \{4, 6, 8\}$) are considered in this example. Each of these scenarios is tested versus SNR, and the number of snapshots is fixed to $K = 200$. The following conclusions can be obtained by observing Figure 2.

(1) For all algorithms in each case, RMSEs decrease gradually with the increase of SNR. Note that the EVD-based CPD algorithm performs better than the other algorithms from case to case when SNR is greater than 10dB.

(2) In the case of low SNR, i.e., SNR is less than 10dB, the problem of the poor estimation accuracy for the proposed EVD-based CPD algorithm is effectively improved with the increase of the number of sensors.

(3) The accuracy of the DOA estimation for MUSIC algorithm changes better obviously, as the number of sensors increases. This is due to the fact that the MUSIC algorithm determines the DOA by executing an exhaustive search [21].

Example 5. Once the estimation of DOA is obtained in Example 4, the polarization parameters can be estimated. Figure 3 shows the estimation accuracy of the parameter γ by the EVD-based CPD and the other algorithms in the scenarios in Example 4. The differences in performance between algorithms are similar to the tendencies emerged in Figure 2. Similarly, the performance of the ALS-based CPD is significantly improved as the number of sensors increases. However, the RMSEs of the algorithms for γ are far away from the CRB boundary, especially for $L = 4$ and $L = 6$. This loss can be explained by the 2-step method of the estimation process, which consists of estimating DOA parameters firstly, and then corresponding polarization parameters in a second step. In other words, the accumulation of errors in the 2-step calculation leads to this loss.

6. Conclusion

In this paper, we proposed an EVD-based CPD approach for tensors to tackle the problem of signal parameters estimation. The procedure of the EVD-based CPD was described, followed by expatiations and simulations of the EVD-based CPD algorithm applied to the signal parameters estimation. The traditional subspace-based algorithms, MUSIC, HOSVD-MUSIC HOSVD-ESPRIT, and the classical ALS-based CPD algorithm, were introduced to verify the efficacy of the proposed EVD-based CPD algorithm.

The strength of the proposed algorithm lies in the estimation of the factor matrices from the tensor model of the signal, which is then used to extract the signal parameters. The signal parameters can be estimated by virtue of the diversities of the spatial and polarization belonging to the factor matrixes. Furthermore, when the number of sensors is small in the array, the proposed algorithm achieves a lower RMSE.

Appendix

A. Polarization Parameters Estimation of MUSIC

Through (18) and (19), we can construct a new matrix:

$$\mathbf{H}_{\hat{\theta}_r,\hat{\phi}_r} = \mathbf{D}^H_{\hat{\theta}_r,\hat{\phi}_r}\widehat{\mathbf{U}}_n\widehat{\mathbf{U}}^H_n\mathbf{D}_{\hat{\theta}_r,\hat{\phi}_r}. \tag{A.1}$$

The polarization parameters estimation of the signals can be expressed as

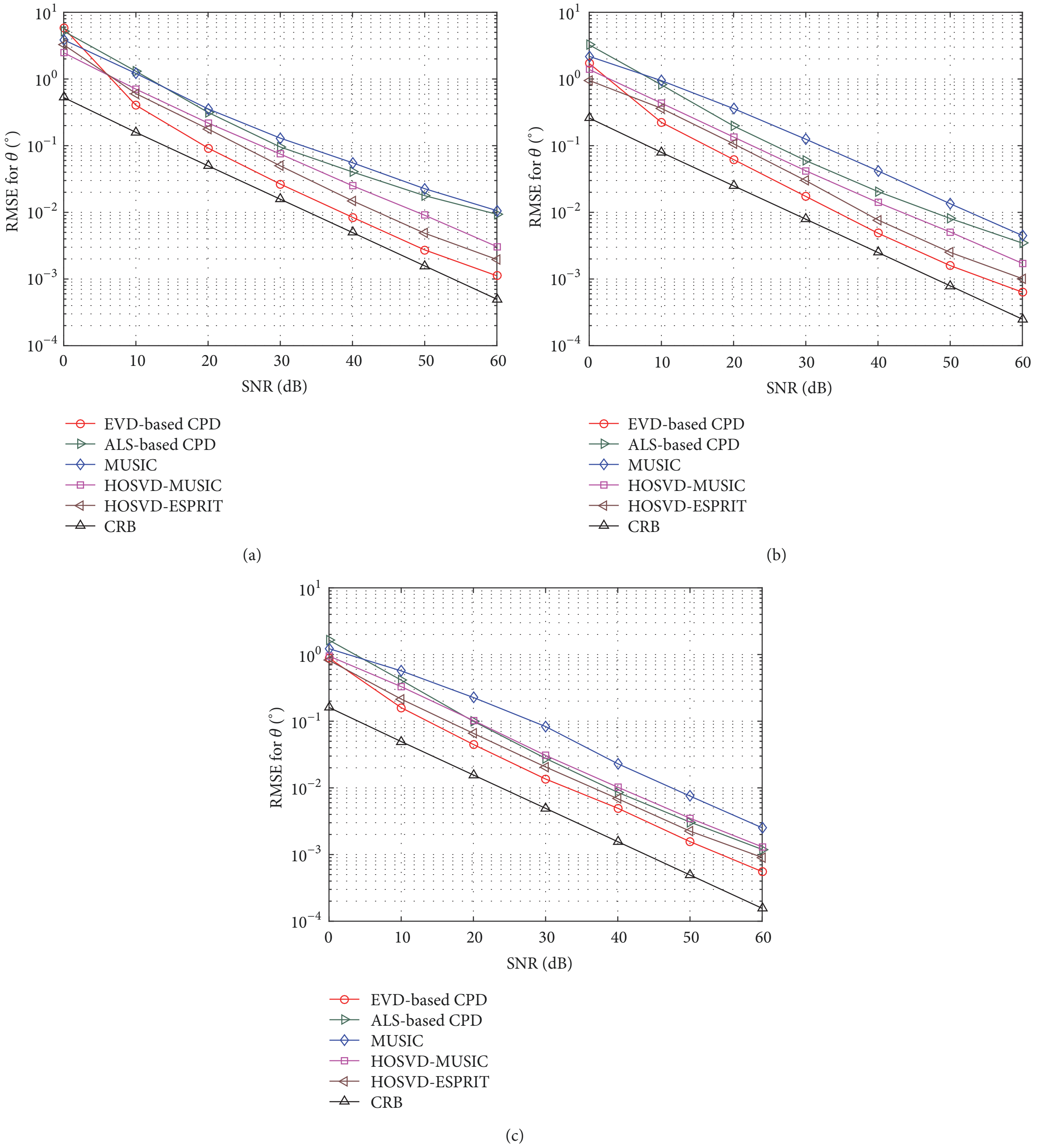
$$\{\hat{\gamma}_r\} = \arctan\left\{\left|\frac{\hat{\boldsymbol{h}}_r(2)}{\hat{\boldsymbol{h}}_r(1)}\right|\right\}, \tag{A.2}$$

$$\{\hat{\eta}_r\} = \arg\left\{\frac{\hat{\boldsymbol{h}}_r(2)}{\hat{\boldsymbol{h}}_r(1)}\right\}, \tag{A.3}$$

where $\hat{\boldsymbol{h}}_r = \mathscr{G}_{\min}\{\mathbf{H}_{\hat{\theta}_r,\hat{\phi}_r}, \mathbf{D}^H_{\hat{\theta}_r,\hat{\phi}_r}\mathbf{D}_{\hat{\theta}_r,\hat{\phi}_r}\}$, $r \in [1,\ldots,R]$, and the operator $\mathscr{G}_{\min}(\cdot)$ symbolizes the generalized eigenvector corresponding to the least generalized eigenvalue.

B. HOSVD-ESPRIT Algorithm

The HOSVD-ESPRIT algorithm based on tensor model can be adopted, when the array has translation invariant property; i.e., the array consists of multiple spatial matching subarrays. This section refers to the idea in [10] to derive the HOSVD-ESPRIT model based on the vector-sensor arrays.



(a)

(b)

(c)

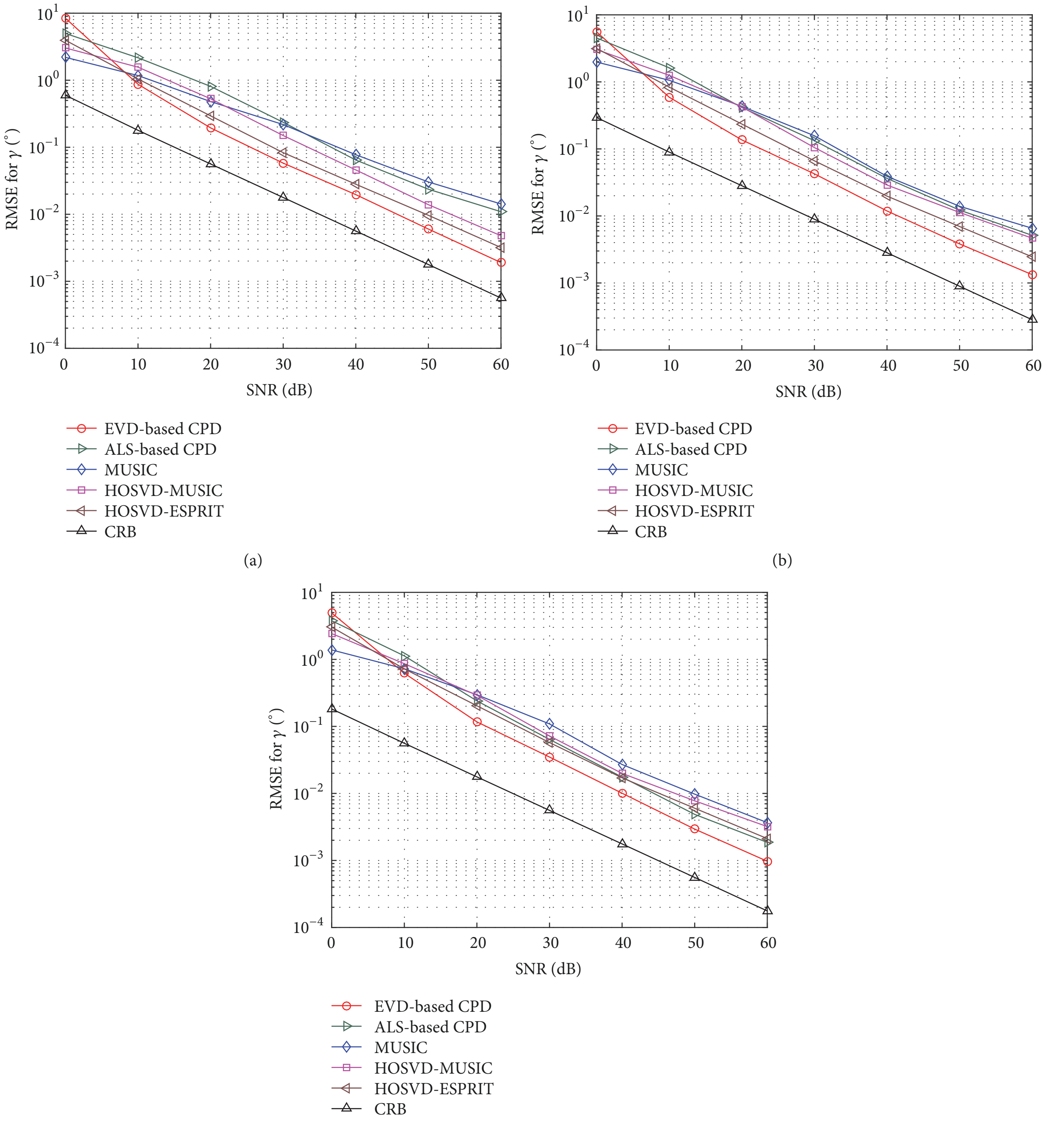
Figure 2: DOA RMSE versus SNR for $K = 200$ and various numbers of sensors. (a) Array of $L = 4$, (b) array of $L = 6$, and (c) array of $L = 8$.

Assume that the array is obtained by G linear translations of the reference subarray, which is obtained by selecting a single vector sensor. Construct the spatial steering vector as follows:

$$\mathbf{a}^{s}(\theta,\phi) = \mathbf{a}'_{G}(\theta,\phi)\otimes,\ldots,\otimes\mathbf{a}'_{1}(\theta,\phi), \qquad \text{(B.1)}$$

where $\mathbf{a}'_{g}(\theta,\phi) \in \mathbb{C}^{L_g \times 1}$ represents the gth spatial translation vector, L_g is the number of the gth spatial subarray, $L = \Pi_{g=1}^{G} L_g$, and $1 \leqslant g \leqslant G$. Therefore, the steering tensor can be constructed as follows:

$$\mathscr{A}(\Psi) = \mathbf{a}^{p}(\Psi)\circledcirc\mathbf{a}'_{1}(\theta,\phi)\circledcirc,\ldots,\circledcirc\mathbf{a}'_{G}(\theta,\phi). \qquad \text{(B.2)}$$



(a)

(b)

(c)

FIGURE 3: Polarization parameter's RMSE versus SNR for $K = 200$ and various numbers of sensors. (a) Array of $L = 4$, (b) array of $L = 6$, and (c) array of $L = 8$.

Furthermore, the tensor-output of K snapshots can be denoted as

$$\mathcal{Y} = \sum_{r=1}^{R} \mathcal{A}\left(\Psi_r\right) \circledcirc \mathbf{s}_r + \mathcal{N}, \tag{B.3}$$

where $\mathcal{Y} \in \mathbb{C}^{L_G \times \ldots \times L_1 \times J \times K}$. According to (22), the HOSVD expression of the tensor $\mathcal{Y}$ can be obtained:

$$\mathcal{Y} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2, \ldots, \times_{G+2} \mathbf{U}^{(G+2)}, \tag{B.4}$$

where the core tensor $\mathcal{S}$ of same size as $\mathcal{Y}$. Use $\mathcal{S}^{[s]} \in \mathbb{C}^{L_G \times \ldots \times L_1 \times J \times R}$ to represent the truncated tensor of $\mathcal{S}$ [10]. The low-rank approximation of $\mathcal{Y}$ can be expressed as

$$\left[\mathcal{H}^{[s]}\right]_{G+2}^{T} = \Big(\mathbf{U}^{(1)} \otimes, \ldots, \otimes \mathbf{U}^{(G+2)} \cdot \left[\mathcal{S}^{[s]}\right]_{G+2}^{T}, \tag{B.5}$$

Let $\mathbf{A}_{g+1} = [\mathbf{a}'_g(\theta_1, \phi_1), \ldots, \mathbf{a}'_g(\theta_R, \phi_R)]$ and $\mathbf{A}_1 = [\mathbf{a}^p(\Psi_1), \ldots, \mathbf{a}^p(\Psi_R)]$; $\mathbf{\Phi}_{g'}$ denotes the second row of matrix $\mathbf{A}_{g'}$, where $1 \leqslant g' \leqslant G+1$. Additionally, if $\mathscr{A} = \mathscr{H}^{[s]} \times_{G+2} \mathbf{T}$, there is $\mathbf{\Psi}_{g'} = \mathbf{T}^{-1} \mathbf{\Phi}_{g'} \mathbf{T}$. Furthermore, the parameters estimation of the g'th mode can be expressed as

$$\mathbf{\Psi}_{g'}^{T} = \left(\widetilde{\mathbf{J}}_1^{(g')} \cdot \left[\mathscr{H}^{[s]} \right]_{G+2}^{T} \right)^{\dagger} \cdot \widetilde{\mathbf{J}}_2^{(g')} \cdot \left[\mathscr{H}^{[s]} \right]_{G+2}^{T}, \qquad \text{(B.6)}$$

where $\widetilde{\mathbf{J}}_i^{(g')} = \mathbf{I}_{\Gamma_1^{(g')}} \otimes \mathbf{J}_i^{(g')} \otimes \mathbf{I}_{\Gamma_2^{(g')}}, i = 1, 2, \Gamma_1^{(g')} = \Pi_{q=1}^{g'-1} L_q, \Gamma_2^{(g')} = \Pi_{q=g'+1}^{R+1} L_q$, and $\mathbf{J}_i^{(g')}$ represents the selection matrix for g'th mode. If the array is uniformly spaced in the g'th mode, the maximum overlap case can be constructed as follows:

$$\begin{aligned} \mathbf{J}_1^{(g')} &= \left[\mathbf{I}_{L_{g'}-1}, \mathbf{0}_{(L_{g'}-1)\times 1} \right], \\ \mathbf{J}_2^{(g')} &= \left[\mathbf{0}_{(L_{g'}-1)\times 1}, \mathbf{I}_{L_{g'}-1} \right]. \end{aligned} \qquad \text{(B.7)}$$

C. ALS-Based CPD for the Third-Order Tensor

The approximate factor matrices of the third-order tensor $\mathscr{Y}$ can be obtained by a solution of the optimization problem:

$$\min \left\| \mathscr{Y} - \left[\mathbf{A}_p, \mathbf{A}_s, \mathbf{S} \right]_R \right\|. \qquad \text{(C.1)}$$

According to (3), optimization problem (C.1) can be rewrite as a quadratic form:

$$\min \left\| \mathbf{Y}_{(\mathfrak{F}_1,2)}^{T} - \mathbf{S} \left(\mathbf{A}_s \odot \mathbf{A}_p \right)^{T} \right\|_F. \qquad \text{(C.2)}$$

Furthermore, we can obtain the factor matrix $\mathbf{S}$ by the following approximate form:

$$\mathbf{S} = \mathbf{Y}_{(\mathfrak{F}_1,2)}^{T} \left(\mathbf{A}_s \odot \mathbf{A}_p \right) \left(\left(\mathbf{A}_s^T \mathbf{A}_s \right) \circ \left(\mathbf{A}_p^T \mathbf{A}_p \right) \right)^{-1}. \qquad \text{(C.3)}$$

In a similar way, the update methods of the factor matrices $\mathbf{A}_p$ and $\mathbf{A}_s$ can be derived as

$$\begin{aligned} \mathbf{A}_p &= \mathbf{Y}_{(\mathfrak{F}_2,2)}^{T} \left(\mathbf{S} \odot \mathbf{A}_s \right) \left(\left(\mathbf{S}^T \mathbf{S} \right) \circ \left(\mathbf{A}_s^T \mathbf{A}_s \right) \right)^{-1}, \\ \mathbf{A}_s &= \mathbf{Y}_{(\mathfrak{F}_3,2)}^{T} \left(\mathbf{A}_p \odot \mathbf{S} \right) \left(\left(\mathbf{A}_p^T \mathbf{A}_p \right) \circ \left(\mathbf{S}^T \mathbf{S} \right) \right)^{-1}. \end{aligned} \qquad \text{(C.4)}$$

In order to obtain the optimal solution of the problem (C.1), it is necessary to update the matrices $\mathbf{A}_p$, $\mathbf{A}_s$, and $\mathbf{S}$ through multiple iterations until (C.1) converges to the desired state.

D. Cramér-Rao Lower Bound for the Vector-Sensor Array

Consider the situation described by (14) and establish its matrix form:

$$\boldsymbol{x}(k) = \mathbf{A} \boldsymbol{s}(k) + \mathbf{n}(k), \qquad \text{(D.1)}$$

where $\mathbf{A} = [\mathbf{a}(\Psi_1) \ldots \mathbf{a}(\Psi_R)] \in \mathbb{C}^{LJ\times R}$ and $\boldsymbol{s}(k) = [s_1(k) \ldots s_R(k)]^T \in \mathbb{C}^{R\times 1}$. The unknown parameters vector can be denoted as

$$\mathbf{\Psi} = \left[\Psi_1 \ldots \Psi_R \right]^T, \qquad \text{(D.2)}$$

where Ψ_r represents the unknown parameters vector of the rth source, $r \in [1, \ldots, R]$. Assume that matrix $\mathbf{A}$ in (D.1) is column full rank matrix, and the Jacobian $\partial \mathbf{A} / \partial \mathbf{\Psi}$ is also full rank. Further, set

$$\begin{aligned} \widetilde{\mathbf{A}} &= \left[\widetilde{\boldsymbol{a}}_1^{(1)} \ldots \widetilde{\boldsymbol{a}}_{q_1}^{(1)} \ldots \widetilde{\boldsymbol{a}}_1^{(R)} \ldots \widetilde{\boldsymbol{a}}_{q_R}^{(R)} \right], \\ \widetilde{\boldsymbol{a}}_m^{(n)} &= \frac{\partial \mathbf{a}(\Psi_n)}{\partial \Psi_n(m)}, \end{aligned} \qquad \text{(D.3)}$$

where q_r denotes the number of elements in vector Ψ_r. The main concern of this paper is to investigate the performance of estimating $\mathbf{\Psi}$ in (D.1) from $\boldsymbol{x}(1), \ldots \boldsymbol{x}(K)$.

To simplify the after-mentioned expression of the Cramér-Rao lower bound, two intermediate matrices are constructed as follows:

$$\begin{aligned} \mathbf{U} &= \mathbf{R}_{\boldsymbol{ss}} \left(\mathbf{A}^H \mathbf{A} \mathbf{R}_{\boldsymbol{ss}} + \sigma^2 \mathbf{I} \right)^{-1} \mathbf{A}^H \mathbf{A} \mathbf{R}_{\boldsymbol{ss}}, \\ \mathbf{P} &= \mathbf{I} - \mathbf{A} \left(\mathbf{A}^H \mathbf{A} \right)^{-1} \mathbf{A}^H, \end{aligned} \qquad \text{(D.4)}$$

where $\mathbf{R}_{\boldsymbol{ss}}$ denotes the covariance of the signal matrix $\mathbf{S} = [\boldsymbol{s}(1) \ldots \boldsymbol{s}(K)]$, σ^2 is the noise variance, and $\mathbf{I}$ symbolizes a $R\times R$ unit matrix. The Cramér-Rao lower bound of the unbiased estimation of $\mathbf{\Psi}$ is

$$\begin{aligned} &\mathbf{CRB}(\mathbf{\Psi}) \\ &\quad = \frac{\sigma^2}{2K} \left\{ \operatorname{Re} \left[\operatorname{btr} \left((1 \boxtimes \mathbf{U}) \boxdot \left(\widetilde{\mathbf{A}}^H \mathbf{P} \widetilde{\mathbf{A}} \right)^{bT} \right) \right] \right\}^{-1}, \end{aligned} \qquad \text{(D.5)}$$

where 1 symbolizes a $\overline{q}\times\overline{q}$ matrix with all entries equal to one, $\overline{q} = \sum_{r=1}^{R} q_r$. Assume that $\mathbf{Q}_{<ij>}$ with dimension $p_i \times p_j$ is the (i, j)-th block entry of the matrix $\mathbf{Q}$; the block trace operator $\operatorname{btr}(\cdot)$, the block transpose operator bT, the block Kronecker product $\boxtimes$, and the block Hadamard product $\boxdot$ are defined as follows.

Definition D.1. Block trace operator is

$$[\operatorname{btr}(\mathbf{Q})]_{ij} = \operatorname{tr}\left(\mathbf{Q}_{<ij>} \right). \qquad \text{(D.6)}$$

Definition D.2. Block transpose is

$$\left(\mathbf{Q}^{bT} \right)_{<ij>} = \mathbf{Q}_{<ji>}. \qquad \text{(D.7)}$$

Definition D.3. Block Kronecker product is

$$\left(\mathbf{Q}_1 \boxtimes \mathbf{Q}_2 \right)_{<ij>} = \left[\mathbf{Q}_1 \right]_{<ij>} \otimes \left[\mathbf{Q}_2 \right]_{<ij>}. \qquad \text{(D.8)}$$

Definition D.4. Block Hadamard product is

$$\left(\mathbf{Q}_1 \boxdot \mathbf{Q}_2 \right)_{<ij>} = \left[\mathbf{Q}_1 \right]_{<ij>} \left[\mathbf{Q}_2 \right]_{<ij>}. \qquad \text{(D.9)}$$

Data Availability

The simulation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

We declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by The National Key Research and Development Program of China under Grant no. 2018YFB0505104. This work is also supported by Shenzhen Science and Technology Innovation Committee of Basic Research Projects under Grant no. JCYJ20170306154016149.

References

[1] R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[2] R. Roy and T. Kailath, “ESPRIT-estimation of signal parameters via rotational invariance techniques,” *IEEE Transactions on Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[3] A. J. Barabell, “Improving the resolution performance of eigenstructure-based direction-finding algorithms,” in *Proceedings of the ICASSP 83, IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 336–339, Boston, Mass, USA, 1983.

[4] G. Liu, H. Chen, X. Sun, and R. C. Qiu, “Modified MUSIC Algorithm for DOA Estimation With Nyström Approximation,” *IEEE Sensors Journal*, vol. 16, no. 12, pp. 4673-4674, 2016.

[5] C. Wen, Y. Zhou, M. Tao, J. Wu, and J. Peng, “Beam-Doppler Unitary ESPRIT for Multitarget DOA Estimation,” *International Journal of Antennas and Propagation*, vol. 2018, Article ID 3568286, 10 pages, 2018.

[6] W. Si, X. Qu, L. Liu, and Z. Qu, “Two-Dimensional DOA Estimation in Compressed Sensing with Compressive-Reduced Dimension- $_{lp}$-MUSIC,” *International Journal of Antennas and Propagation*, vol. 2015, Article ID 792181, 9 pages, 2015.

[7] W. Liao, “MUSIC for multidimensional spectral estimation: stability and super-resolution,” *IEEE Transactions on Signal Processing*, vol. 63, no. 23, pp. 6395–6406, 2015.

[8] W. Suleiman, M. Pesavento, and A. Zoubir, “Performance analysis of the decentralized eigendecomposition and ESPRIT algorithm,” *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2375–2386, 2016.

[9] S. Li, X. Chen, and R. He, “Robust cyclic MUSIC algorithm for finding directions in impulsive noise environment,” *International Journal of Antennas and Propagation*, vol. 2017, Article ID 9038341, 9 pages, 2017.

[10] M. Haardt, F. Roemer, and G. Del Galdo, “Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3198–3213, 2008.

[11] M. Boizard, G. Ginolhac, F. Pascal, S. Miron, and P. Forster, “Numerical performance of a tensor MUSIC algorithm based on HOSVD for a mixture of polarized sources,” in *Proceedings of the 21st European Signal Processing Conference, EUSIPCO 2013*, pp. 1–5, Morocco, September 2013.

[12] K. Han and A. Nehorai, “Nested vector-sensor array processing via tensor modeling,” *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2542–2553, 2014.

[13] Y. E. G. Guzman, R. C. de Lamare, and M. Haardt, “Randomized multiple candidate iterative hard thresholding algorithm for direction of arrival estimation,” in *Proceeding of the 22nd International ITG Workshop on Smart Antennas*, vol. 4, pp. 1–4, 2018.

[14] I. Domanov and L. De Lathauwer, “Canonical polyadic decomposition of third-order tensors: relaxed uniqueness conditions and algebraic algorithm,” *Linear Algebra and its Applications*, vol. 513, pp. 342–375, 2017.

[15] H. Xi and A. L. F. de Almeida, “Multiuser receiver for joint symbol/channel estimation in dual-hop relaying systems,” *Wireless Personal Communications*, vol. 83, no. 1, pp. 17–33, 2015.

[16] A. Cichocki, D. Mandic, L. De Lathauwer et al., “Tensor decompositions for signal processing applications: from two-way to multiway component analysis,” *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[17] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Separation*, vol. 4, Academic Press, Oxford, UK, 2010.

[18] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[19] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[20] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, “Parallel factor analysis in sensor array processing,” *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.

[21] F. Raimondi, P. Comon, O. Michel, S. Sahnoun, and A. Helmstetter, “Tensor decomposition exploiting diversity of propagation velocities: Application to localization of icequake events,” *Signal Processing*, vol. 118, pp. 75–88, 2016.

[22] L. Liu, L. Wang, and Z. L. Zhang, “Vector Sensor Based Signal Parameters Estimation by Exploiting CPD of Tensors,” *IEEE Sensors Letters*, vol. 2, no. 3, pp. 1–4, 2018.

[23] I. Domanov and L. De Lathauwer, “Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition,” *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 2, pp. 636–660, 2014.

[24] S. E. Leurgans, R. T. Ross, and R. B. Abel, “A decomposition for three-way arrays,” *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 4, pp. 1064–1083, 1993.

[25] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, Md, USA, 4th edition, 2013.