Blind Demixing for Low-Latency Communication

Jialin Dong , Kai Yang, and Yuanming Shi

School of Information Science and Technology, ShanghaiTech University, Shanghai, China
E-mail: {dongjl, yangkai, shiym}@shanghaitech.edu.cn

***Abstract*—In the next generation wireless networks, low-latency communication is critical to support emerging diversified applications, e.g., Tactile Internet and Virtual Reality. In this paper, a novel blind demixing approach is developed to reduce the channel signaling overhead, thereby supporting low-latency communication. Specifically, we develop a low-rank approach to recover the original information only based on a single observed vector without any channel estimation. Unfortunately, this problem turns out to be a highly intractable non-convex optimization problem due to the multiple non-convex rank-one constraints. To address the unique challenges, the quotient manifold geometry of product of complex asymmetric rank-one matrices is exploited by equivalently reformulating original complex asymmetric matrices to the Hermitian positive semidefinite matrices. We further generalize the geometric concepts of the complex product manifolds via element-wise extension of the geometric concepts of the individual manifolds. A scalable Riemannian trust-region algorithm is then developed to solve the blind demixing problem efficiently with fast convergence rates and low iteration cost. Numerical results will demonstrate the algorithmic advantages and admirable performance of the proposed algorithm compared with the state-of-art methods.**

I. Introduction

Recently, various emerging 5G applications such as Tactile Internet [1] and Virtual Reality [2] are unleashing a sense of urgency in providing *low-latency* communications [3], for which innovative new technologies need to be developed. To achieve this goal, various solutions have been investigated, which can be categorized into three main types, i.e., radio access network (RAN), core network and caching [4]. In particular, reducing the packet blocklength, e.g., short packets communication [5], is a promising technique in RAN to support low-latency communication, for which the theoretical analysis on the tradeoffs among the channel coding rate, blocklength and error probability was provided in [6]. Furthermore, by pushing the computation and storage resources to the network edge, Fog-RAN provides a principled way to reduce the latency [7].

However, to design a latency communication system, channel signaling overhead reduction becomes critical. In particular, when packet blocklength is reduced as envisioned in 5G systems, channel signaling overhead dominates the major portion of the packet [4]. To address this issue, numerous research efforts have been paid on channel signaling overhead reduction. The compressed sensing based approach was proposed in [8] to estimate sparse multipath channels, thereby achieving good performance with low energy, latency and bandwidth cost. The recent proposal of topological interference alignment [9] serves a promising way to manage the interference only based on the network connectivity information at the transmitters. Furthermore, by equipping large number of antennas at the base stations, massive MIMO [10] can manage interference without channel estimation at the transmitters. However, all the methods [8], [9], [10] still assume that the channel state information (CSI) is available at the receivers for message decoding.

More recently, a new proposal has emerged, namely, the mixture of blind deconvolution and demixing [11], i.e., *blind demixing* for brief, regarded as a promising solution to support efficient low-latency communication without channel estimation at both transmitters and receivers. In particular, blind deconvolution is a problem of estimating two unknown vectors from their convolution, which can be exploited in the context of channel coding for multipath channel protection [12]. However, the results of blind deconvolution [12] cannot be directly extended to the blind demixing problem since only a single observed vector is available. Demixing refers to the problem of identifying multiple structured signals given the mixture of measurements of these signals, which can be exploited in a secure communications protocol [13]. In particular, the measurement matrices in the demixing problem are normally assumed to be full-rank matrices to assist theoretical analysis [14]. However, the measurement matrices in the blind demixing problem are rank-one matrices [15], which hampers the extension of results developed in [14] to the blind demixing problem.

In this paper, we propose a low-rank approach to recover the original signals for the demixing problem. However, the resulting rank-constrained optimization problem is known to be non-convex and highly intractable. A growing body of literature has proposed marvelous algorithms to deal with low-rank problems. In particular, convex relaxation approach is an effective way to solve this problem with theoretical guarantees [11]. However, it is not scalable to medium- and large-scale problems due to the high iteration cost of the convex program technique. To enable scalability, non-convex algorithms (e.g, regularized gradient descent algorithm [15] and iterative hard thresholding method [16]), endowed with lower iteration cost, have been developed. However, the overall computational complexity of these algorithms is still high due

This work was partly supported by the National Nature Science Foundation of China under Grant No. 61601290, and the Shanghai Sailing Program under Grant No. 16YF1407700.

978-1-5386-1734-2/18/$31.00 ©2018 IEEE

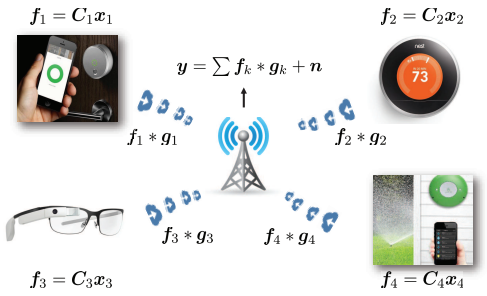

Fig. 1. Blind demixing for multi-user low-latency communication systems without channel estimation.

to the slow convergence rate, i.e., high iteration complexity.

To address the limitations of the existing algorithms for the blind demixing problem, we propose the Riemannian optimization algorithm over product complex manifolds in order to simultaneously reduce the iteration cost and iteration complexity. Specifically, the quotient manifold geometry of product of complex asymmetric rank-one matrices is exploited by equivalently reformulating the original complex asymmetric matrices as Hermitian positive semidefinite (PSD) matrices. To reduce the iteration complexity, the Riemannian trust-region algorithm is developed to support superlinear convergence rate. By exploiting the benign geometric structure of the blind demixing problem, i.e., symmetric rank-one matrices, the iteration cost can be significantly reduced (the same as the regularized gradient descent algorithm [15]) and is scalable to large problem sizes.

The major contributions to solve the blind demixing problem are summarized as follow:

1) We present a novel blind demixing approach to support low-latency communication, thereby recovering information signals without channel estimation. A low-rank approach is developed to solve the blind demixing problem.
2) To efficiently exploit the quotient manifold geometry of the product of complex asymmetric rank-one matrices, we equivalently reformulate the original complex asymmetric matrices to Hermitian positive semidefinite matrices.
3) To simultaneously reduce the iteration cost and iteration complexity as well as enhance the performance, we develop the scalable Riemannian trust-region algorithm by exploiting the benign geometric structure of symmetric rank-one matrices.

Simulation results demonstrate that the proposed Riemannian trust-region algorithm outperforms the state-of-art algorithms in terms of algorithmic advantages and admirable performance.

II. System Model And Problem Formulation

In this section, we present a blind demixing approach to support low-latency communication by reducing the channel signaling overhead.

A. System Model

We consider a network with one base station and s active mobile users, as shown in Fig. 1. Specifically, let $\boldsymbol{x}_k \in \mathbb{C}^N$ be the original data symbols of length N from the k-th user. Over L time slots, the transmit signal at the k-th user is given by $\boldsymbol{f}_k = \boldsymbol{C}_k\boldsymbol{x}_k$, where $\boldsymbol{C}_k \in \mathbb{C}^{L\times N}$ with $L > N$ is the encoding matrix and known to the base station. Furthermore, the signal $\boldsymbol{f}_k$ convolves with the channel impulse response during transmissions, represented by $\boldsymbol{f}_k * \boldsymbol{g}_k$ [17], where the channel impulse response $\boldsymbol{g}_k \in \mathbb{C}^L$ is assumed to stay constant during transmission. Thus, the received signal is given as

$$\boldsymbol{z} = \sum\nolimits_{k=1}^{s} \boldsymbol{f}_k * \boldsymbol{g}_k + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{n}$ denotes the additive white complex Gaussian noise and the operator $*$ denotes the circular convolution. Our goal is to recover the original information $\{\boldsymbol{x}_k\}_{k=1}^s$ from the single observation $\boldsymbol{z}$ without knowing channel impulse response $\{\boldsymbol{g}_k\}_{k=1}^s$. We call this problem as the *blind demixing* problem.

However, the information recovery problem is highly intractable without any further structural assumptions. In wireless communication, we can design the encoding matrices $\{\boldsymbol{C}_k\}_{k=1}^s$ such that satisfy "local" mutual incoherence conditions [11]. Furthermore, due to the physical properties of wave propagation [18], the impulse response $\boldsymbol{g}_k$ is compactly supported [16]. Here, the size of the compact set of $\boldsymbol{g}_k$, i.e., K where $K < L$, is termed as its maximum delay spread from engineering perspective [16]. The channel impulse response $\boldsymbol{g}_k \in \mathbb{C}^L$ is thus can be rewritten as $\boldsymbol{g}_k = [\boldsymbol{h}_k^T, 0, \cdots, 0]^T$, where $\boldsymbol{h}_k \in \mathbb{C}^K$ is the non-zero impulse response. In this paper, we assume that the impulse response $\boldsymbol{g}_k$ is not known to both receivers and transmitters during the transmissions in order to reduce the channel signaling overhead [15].

B. Demixing of Rank-One Matrices

Define $\boldsymbol{F} \in \mathbb{C}^{L\times L}$ as the normalized discrete Fourier transform (DFT) matrix and let $\boldsymbol{B} \in \mathbb{C}^{L\times K}$ consist of the first K columns of $\boldsymbol{F}$. It is convenient to represent the formulation (1) in the Fourier domain [11]

$$\boldsymbol{y} = \boldsymbol{F}\boldsymbol{z} = (\boldsymbol{F}\boldsymbol{C}_k\boldsymbol{x}_k) \odot \boldsymbol{B}\boldsymbol{h}_k + \boldsymbol{F}\boldsymbol{n}, \tag{2}$$

where $\odot$ denotes the componentwise product. Let $\boldsymbol{M}^{\mathsf{H}}$ and $\boldsymbol{z}^{\mathsf{H}}$ denote the complex conjugate transpose of matrix and vector, respectively. We define $\langle \boldsymbol{M}_1, \boldsymbol{M}_2\rangle = \mathrm{Tr}(\boldsymbol{M}_1^{\mathsf{H}}\boldsymbol{M}_2)$, where $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are two complex matrices. Let $\bar{\boldsymbol{z}}$ denote the complex conjugate of the vector $\boldsymbol{z}$. The first term of (2) can be further rewritten as [16]

$$[(\boldsymbol{F}\boldsymbol{C}_k\boldsymbol{x}_k) \odot \boldsymbol{B}\boldsymbol{h}_k]_i = (\boldsymbol{c}_{ki}^{\mathsf{H}}\boldsymbol{x}_k)(\boldsymbol{b}_i^{\mathsf{H}}\boldsymbol{h}_k) = \langle \boldsymbol{c}_{ki}\bar{\boldsymbol{b}}_i^{\mathsf{H}}, \boldsymbol{X}_k\rangle, \tag{3}$$

where $\boldsymbol{c}_{ki}^{\mathsf{H}}$ denotes the i-th row of $\boldsymbol{F}\boldsymbol{C}_k$, $\boldsymbol{b}_i^{\mathsf{H}}$ represents the i-th row of $\boldsymbol{B}$ and $\boldsymbol{X}_k = \boldsymbol{x}_k\bar{\boldsymbol{h}}_k^{\mathsf{H}} \in \mathbb{C}^{N\times K}$ is a rank-one matrix. Hence, the received signal at the base station can be represented in the Fourier domain as

$$\boldsymbol{y} = \sum\nolimits_{k=1}^{s} \mathcal{A}_k(\boldsymbol{X}_k) + \boldsymbol{e}, \tag{4}$$

where the vector $\boldsymbol{e} = \boldsymbol{F}\boldsymbol{n}$ and the linear operator $\mathcal{A}_k : \mathbb{C}^{N\times K} \to \mathbb{C}^L$ is given as [15]

$$\mathcal{A}_k(\boldsymbol{X}_k) := \{\langle \boldsymbol{c}_{ki}\bar{\boldsymbol{b}}_i^{\mathsf{H}}, \boldsymbol{X}_k\rangle\}_{i=1}^L = \{\langle \boldsymbol{A}_{ki}, \boldsymbol{X}_k\rangle\}_{i=1}^L, \quad (5)$$

with $\boldsymbol{A}_{ki} = \boldsymbol{c}_{ki}\bar{\boldsymbol{b}}_i^{\mathsf{H}}$. We thus formulate the blind demixing problem as the following low-rank optimization problem:

$$\mathscr{P}: \underset{\boldsymbol{W}_k, k=1,\cdots,s}{\text{minimize}} \quad \Big\| \sum\nolimits_{k=1}^s \mathcal{A}_k(\boldsymbol{W}_k) - \boldsymbol{y}\Big\|^2 \\ \text{subject to} \quad \text{rank}(\boldsymbol{W}_k) = 1,\ k = 1,\cdots,s, \quad (6)$$

where $\{\mathcal{A}_k\}_{k=1}^s$ are known and $\|\boldsymbol{z}\|$ denotes the ℓ_2-norm of the vector $\boldsymbol{z}$. However, problem $\mathscr{P}$ turns out to be highly intractable due to non-convexity of rank-one constraints. In this paper, we shall propose efficient algorithms with low iteration cost, fast convergence rate and good performance to solve the highly intractable problem $\mathscr{P}$, thereby recovering the original information $\{\boldsymbol{x}_k\}_{k=1}^s$ without channel estimation.

C. Problem Analysis

To address the algorithmic challenge of problem $\mathscr{P}$, enormous progress has been recently made to develop convex methods [11] and non-convex methods [15], [16].

1) Convex Relaxation Approach: A line of literature [11] adopted the nuclear norm minimization (NNM) method to reformulate the problem $\mathscr{P}$ as

$$\underset{\boldsymbol{W}_k, k=1,\cdots,s}{\text{minimize}} \quad \sum\nolimits_{k=1}^s \|\boldsymbol{W}_k\|_* \\ \text{subject to} \quad \Big\| \sum\nolimits_{k=1}^s \mathcal{A}_k(\boldsymbol{W}_k) - \boldsymbol{y}\Big\| \le \varepsilon, \quad (7)$$

where $\boldsymbol{W}_k \in \mathbb{C}^{N\times K}$, the operator $\|\cdot\|_*$ represents the nuclear norm, the parameter ε is an upper bound of $\|\boldsymbol{e}\|$ in (4) and assumed known a *prior*. While the blind demixing problem can be solved by convex technique provably and robustly under certain situations, the convex relaxation approach is computationally infeasible to medium-scale or large-scale problems due to the limitations of high iteration cost. This motivates the development of efficient non-convex approaches with lower iteration cost.

2) Non-convex Optimization Paradigms: A line of recent work [15], [16] has developed non-convex algorithms which reduces the iteration cost. In particular, work [16] solved the problem $\mathscr{P}$ via hard thresholding technique. Specifically, the t-th iterate with respect to the k-th variable is given by

$$\boldsymbol{W}_k^{[t+1]} = \mathcal{P}_r\big(\boldsymbol{W}_k^{[t]} + \alpha_k^{[t]}\mathcal{P}_{T_{k,t}}(\boldsymbol{G}_k^{[t]})\big), \quad (8)$$

where the hard thresholding operator $\mathcal{P}_r$ returns the best rank-r approximation of a matrix, $\mathcal{P}_{T_{k,t}}(\boldsymbol{G}_k^{[t]})$ represents the projected gradient descent direction, with $T_{k,t}$ as the tangent space of the rank-r matrix manifold at the current iterate $\boldsymbol{W}_k^{[t]}$ and the stepsizes α_k's are provided in [16].

Moreover, work [15] developed an algorithm based on matrix factorization and regularized gradient descent methods. Specifically, problem $\mathscr{P}$ can be rewritten as

$$\underset{\boldsymbol{u}_k, \boldsymbol{v}_k, k=1,\cdots,s}{\text{minimize}} \ F(\boldsymbol{u}_k, \boldsymbol{v}_k) := g(\boldsymbol{u}_k, \boldsymbol{v}_k) + \lambda R(\boldsymbol{u}_k, \boldsymbol{v}_k), \quad (9)$$

where $g(\boldsymbol{u}_k, \boldsymbol{v}_k) := \|\sum_{k=1}^s \mathcal{A}_k(\boldsymbol{u}_k\boldsymbol{v}_k^{\mathsf{H}}) - \boldsymbol{y}\|^2$. Please refer to [15] for the details about the regularized parameter λ and the regularizer $R(\boldsymbol{u}_k, \boldsymbol{v}_k)$. The algorithm starts from a good initial point and updates the following iterates simultaneously:

$$\boldsymbol{u}_k^{[t+1]} = \boldsymbol{u}_k^{[t]} - \eta\nabla F_{\boldsymbol{u}_k}(\boldsymbol{u}_k^{[t]}, \boldsymbol{v}_k^{[t]}), \\ \boldsymbol{v}_k^{[t+1]} = \boldsymbol{v}_k^{[t]} - \eta\nabla F_{\boldsymbol{v}_k}(\boldsymbol{u}_k^{[t]}, \boldsymbol{v}_k^{[t]}), \quad (10)$$

where $\nabla F_{\boldsymbol{u}_k}$ denotes the derivative of the objective function (9) with respect to $\boldsymbol{u}_k$. Although the above non-convex algorithms have low iteration cost, the overall computational complexity is still high due to the slow convergence rate, i.e., high iteration complexity.

D. Riemannian Optimization Approach

In this paper, we develop a Riemannian trust-region algorithm based on the Riemannian optimization framework [19, Section 7] to solve problem $\mathscr{P}$, thereby addressing the limitations of the existing algorithms (e.g., regularized gradient descent algorithm [15], nuclear norm minimization method [12] and fast iterative hard thresholding algorithm [16]) by

- Exploiting the Riemannian quotient geometry of the product of *complex asymmetric rank-one* matrices to reduce the iteration cost.
- Developing scalable Riemannian trust-region based on the Riemannian optimization framework to reduce the iteration complexity.

The Riemannian optimization technique has been applied in a wide range of area to solve rank-constrained problem and achieves excellent performance, e.g., the low-rank matrix completion problem [20] and topological interference management (TIM) problem [9]. However, all of the existing Riemannian optimization algorithms are mainly for real Riemannian manifold with single asymmetric variable. Therefore, to leverage the algorithmic advantages of the Riemannian optimization framework, challenges arise due to *multiple complex asymmetric rank-one matrices* in problem $\mathscr{P}$. In this paper, we thus mainly address the following coupled challenges:

- Constructing product Riemannian manifolds for the multiple complex asymmetric rank-one matrices.
- Developing the Riemannian optimization algorithm on the complex product manifolds.

III. Riemannian Optimization over Product Manifolds

In this section, to exploit the Riemannian quotient geometry of the product of complex asymmetric rank-one matrices [21], we reformulate the original optimization problem on complex asymmetric matrices to the one on Hermitian PSD matrices. The Riemannian optimization algorithm on product manifolds is then developed.

A. Handling Complex Asymmetric Matrices

Define the linear map $\mathcal{J}_k : \mathbb{C}^{n\times n} \to \mathbb{C}^L$ where $n = N + K$ and a Hermitian positive semidefinite (PSD) matrix $\boldsymbol{J}_{ki}$ such that $[\mathcal{J}_k(\boldsymbol{Y}_k)]_i = \langle \boldsymbol{J}_{ki}, \boldsymbol{Y}_k \rangle$ with $\boldsymbol{Y}_k \in \mathbb{C}^{n\times n}$ and $\boldsymbol{J}_{ki}$ as

$$\boldsymbol{J}_{ki} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A}_{ki} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \in \mathbb{C}^{n\times n}. \tag{11}$$

It is easy to check that $[\mathcal{J}_k(\boldsymbol{M}_k)]_i = \langle \boldsymbol{A}_{ki}, \boldsymbol{X}_k \rangle$ where $\boldsymbol{A}_{ki}$ is given in (5) and $\boldsymbol{X}_k \in \mathbb{C}^{N\times K}$. Problem $\mathscr{P}$ can be equivalently reformulated as the following optimization problem on the set of Hermitian positive semidefinite matrices

$$\begin{aligned} \underset{\boldsymbol{M}_k, k=1,\cdots,s}{\text{minimize}} \quad & \Big\| \sum\nolimits_{k=1}^{s} \mathcal{J}_k(\boldsymbol{M}_k) - \boldsymbol{y} \Big\|^2 \\ \text{subject to} \quad & \text{rank}(\boldsymbol{M}_k) = 1,\ k = 1, \cdots, s, \end{aligned} \tag{12}$$

where $\boldsymbol{M}_k \in \mathbb{S}_+^T$ with $\mathbb{S}_+^T$ denoting the space of Hermitian positive semidefinite matrices and $T = (N + K) \times (N + K)$. Furthermore, we define $\boldsymbol{V} = \{\boldsymbol{M}_k\}_{k=1}^s \in \mathcal{M}^s$, where $\mathcal{M}$ denotes the non-compact manifold encoded with the rank-one matrices and $\mathcal{M}^s$ represents the product of s manifolds $\mathcal{M}$. By exploiting the Riemannian manifold geometry, the rank-constrained optimization problem (12) can be transformed into the following unconstrained optimization problem over the search space of the product manifolds $\mathcal{M}^s$ [20]:

$$\underset{\boldsymbol{V}=\{\boldsymbol{M}_k\}_{k=1}^s}{\text{minimize}} \quad f(\boldsymbol{V}) := \Big\| \sum\nolimits_{k=1}^{s} \mathcal{J}_k(\boldsymbol{M}_k) - \boldsymbol{y} \Big\|^2. \tag{13}$$

We shall demonstrate the algorithmic advantages of the symmetric transformation in the next subsection. Note that the theoretical advantages of the symmetric transformation have been recently revealed in [22] for the low-rank problems in machine learning.

B. Riemannian Optimization on Product Manifolds

To solve the rank-one optimization problem (13) efficiently, we factorize the rank-one matrix $\boldsymbol{M}_k \in \mathbb{S}_+^T$ as $\boldsymbol{M}_k = \boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}}$. Problem (13) thus can be reformulated as

$$\underset{\boldsymbol{v}=\{\boldsymbol{w}_k\}_{k=1}^s}{\text{minimize}} \quad f(\boldsymbol{v}) := \Big\| \sum\nolimits_{k=1}^{s} \mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}}) - \boldsymbol{y} \Big\|^2, \tag{14}$$

where $\boldsymbol{w}_k \in \mathbb{C}^{N+K}$ with $k = 1, \cdots, s$. To simplify the notations, we abuse the notion of $\boldsymbol{v} \in \mathcal{M}^s$ and consider the optimization problem (14) with respect to $\boldsymbol{v}$ in the following. To remove the indeterminacy, i.e, $\boldsymbol{M}_k = \boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}} = (a\boldsymbol{w}_k)(a\boldsymbol{w}_k)^{\mathsf{H}}$, where $a \in \mathbb{C}$ and $aa^* = 1$ with a^* as the conjugate of the a, we exploit the geometry of quotient manifold [19]. Since the quotient manifold is abstract, the geometric concepts in the quotient space call for the matrix representations in the computational space $\mathcal{M}^s$. Specifically, to develop the Riemannian optimization algorithm over the product manifolds, several geometric concepts such as the notion of length (i.e. Riemannian metric $g_{\boldsymbol{w}_k}$), set of directional derivatives (i.e., horizontal space $\mathcal{H}_{\boldsymbol{w}_k}$) and motion along geodesics (i.e., retraction $\mathcal{R}_{\boldsymbol{w}_k}$) need to be derived [19]. The concrete optimization-related ingredients are shown in Table I. Based on these ingredients, we develop the Riemannian algorithm to solve the blind demixing problem $\mathscr{P}$.

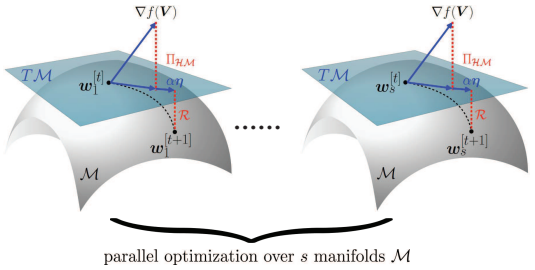

Fig. 2. Graphical representation of the concept of Riemannian optimization on the product manifolds.

Based on the geometry of the product manifold, the Riemannian optimization developed on the product manifolds $\mathcal{M}^s$ can be elementwisely processed over the individual manifolds $\mathcal{M}$. Specifically, for $k = 1, 2, \cdots, s$, we parallelly search the descent direction $\boldsymbol{\eta}$ on the horizontal space $\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$ and parallelly retract it on the individual manifold $\mathcal{M}$ via the retraction mapping $\mathcal{R}_{\boldsymbol{w}_k}$. The schematic viewpoint of Algorithm 1 is illustrated in Fig. 2.

Algorithm 1: Riemannian Optimization on Product Manifolds

Given: Riemannian manifold $\mathcal{M}^s$ with Riemannian metric $g_{\boldsymbol{v}}$, retraction mapping $\mathcal{R}_{\boldsymbol{v}} = \{\mathcal{R}_{\boldsymbol{w}_k}\}_{k=1}^s$, objective function f and the stepsize α.

Output: $\boldsymbol{v} = \{\boldsymbol{w}_k\}_{k=1}^s$

1: **Initialize:** initial point $\boldsymbol{v}^{[0]} = \{\boldsymbol{w}_k^{[0]}\}_{k=1}^s$, $t = 0$
2: **while** not converged **do**
3: **for all** $k = 1, \cdots, s$ **do in parallel**
4: Compute a descent direction $\boldsymbol{\eta}$. (e.g., via implementing trust-region method)
5: Update $\boldsymbol{w}_k^{[t+1]} = \mathcal{R}_{\boldsymbol{w}_k^{[t]}}(\alpha\boldsymbol{\eta})$
6: $t = t + 1$.
7: **end for**
8: **end while**

C. Riemannian Trust-region Algorithm

In this paper, we search the descent direction via developing a scalable Riemannian trust-region algorithm [19, Section 7] which have superlinear convergence rate. We first consider the circumstance of searching the direction $\boldsymbol{\eta}_{\boldsymbol{w}_k}$ on the horizontal space $\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$, which paves the way to parallelly search the descent direction on the horizontal space $\mathcal{H}_{\boldsymbol{v}}\mathcal{M}^s$. At each iteration, let $\boldsymbol{w}_k \in \mathcal{M}$, we solve the trust-region sub-problem [19]:

$$\begin{aligned} \underset{\boldsymbol{\eta}_{\boldsymbol{w}_k}}{\text{minimize}} \quad & m(\boldsymbol{\eta}_{\boldsymbol{w}_k}) \\ \text{subject to} \quad & g_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}, \boldsymbol{\eta}_{\boldsymbol{w}_k}) \le \delta^2, \end{aligned} \tag{15}$$

where $\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$, δ denotes the trust-region radius and the cost function is given by $m(\boldsymbol{\eta}_{\boldsymbol{w}_k}) =$

TABLE I
ELEMENT-WISE OPTIMIZATION-RELATED INGREDIENTS FOR PROBLEM $\mathscr{P}$

	minimize $_{\boldsymbol{w}_k\in\mathcal{M}} \| \sum_{k=1}^{s} \mathcal{J}_k(\boldsymbol{w}_k\boldsymbol{w}_k^{\mathsf{H}}) - \boldsymbol{y}\|^2$
Computational space $\mathcal{M}$	$\mathbb{C}_*^{N+K}$
Quotient space $\mathcal{M}/\sim$	$\mathbb{C}_*^{N+K}/\mathrm{SU}(1)$
Riemannian metric $g_{\boldsymbol{w}_k}$	$g_{\boldsymbol{w}_k}(\boldsymbol{\zeta}_{\boldsymbol{w}k},\boldsymbol{\eta}_{\boldsymbol{w}k}) = \mathrm{Tr}(\boldsymbol{\zeta}_{\boldsymbol{w}k}^{\mathsf{H}}\boldsymbol{\eta}_{\boldsymbol{w}k} + \boldsymbol{\eta}_{\boldsymbol{w}k}^{\mathsf{H}}\boldsymbol{\zeta}_{\boldsymbol{w}k})$
Horizontal space $\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$	$\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathbb{C}^{N+K} : \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}}\boldsymbol{w}_k = \boldsymbol{w}_k^{\mathsf{H}}\boldsymbol{\eta}_{\boldsymbol{w}_k}$
Horizontal space projection	$\Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\boldsymbol{\eta}_{\boldsymbol{w}_k}) = \boldsymbol{\eta}_{\boldsymbol{w}_k} - a\boldsymbol{w}_k,\ a = (\boldsymbol{w}_k^{\mathsf{H}}\boldsymbol{\eta}_{\boldsymbol{w}_k} - \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}}\boldsymbol{w}_k)/2\boldsymbol{w}^{\mathsf{H}}\boldsymbol{w}$
Riemannian gradient $\mathrm{grad}_{\boldsymbol{w}_k} f$	$\mathrm{grad}_{\boldsymbol{w}} f = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\frac{1}{2}\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v}))$
Riemannian Hessian $\mathrm{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$	$\mathrm{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\frac{1}{2}\nabla^2_{\boldsymbol{w}_k} f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}])$
Retraction $\mathcal{R}_{\boldsymbol{w}_k} : T_{\boldsymbol{w}_k}\mathcal{M} \to \mathcal{M}$	$\mathcal{R}_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}) = \boldsymbol{w}_k + \boldsymbol{\eta}_{\boldsymbol{w}_k}$

$g_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}, \mathrm{grad}_{\boldsymbol{w}_k} f) + \frac{1}{2} g_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}, \mathrm{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}])$, with $\mathrm{grad}_{\boldsymbol{w}_k} f$ and $\mathrm{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$ as the matrix representations of Riemannian gradient and Riemannian Hessian in the quotient space, respectively. Moreover, the potential iterate being accepted or rejected depends on whether the decrease of the objective function is sufficient or not [19, Section 7]. If the iterate is accepted, the new iterate is represented as [21] $\mathcal{R}_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}) = \boldsymbol{w}_k + \boldsymbol{\eta}_{\boldsymbol{w}_k}$. Based on the above framework, the Riemannian trust-region algorithm is parallelly executed on individual manifold to solve the optimization problem (14). The overview of the Riemannian optimization can be found in [19]. The details about the derivations of Riemannian gradient and Riemannian Hessian are illustrated in the following.

The Riemannian gradient $\mathrm{grad}_{\boldsymbol{w}_k} f$ can be induced from the Euclidean gradient of $f(\boldsymbol{v})$ with respect to $\boldsymbol{w}_k$ and the relationship between them is given by [19, Section 3.6] $g_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}, \mathrm{grad}_{\boldsymbol{w}_k} f) = \nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$, where the directional vector is $\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$. Thus the Riemannian gradient is given as

$$\mathrm{grad}_{\boldsymbol{w}_k} f = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\frac{1}{2}\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})), \tag{16}$$

where $\Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\cdot)$ denotes the horizontal space projector operator. In particular, the Euclidean gradient of $f(\boldsymbol{v})$ with respect to $\boldsymbol{w}_k$ is given as

$$\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v}) = 2 \cdot \sum\nolimits_{i=1}^{L} (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k, \tag{17}$$

where $c_i = \sum_{k=1}^{s} [\mathcal{J}_k(\boldsymbol{w}_k\boldsymbol{w}_k^{\mathsf{H}})]_i - y_i$ and c_i^* denotes the conjugate of the complex scaler c_i.

Furthermore, the Riemannian Hessian $\mathrm{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$ is given by [19, Section 5.5]

$$\mathrm{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\nabla^2_{\boldsymbol{w}_k} f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}]), \tag{18}$$

where the directional derivative of Euclidean gradient $\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})$ (17) in the direction of $\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$, given by

$$\begin{aligned}\nabla^2_{\boldsymbol{w}_k} f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}] =& 2\sum\nolimits_{i=1}^{L} (b_i \boldsymbol{J}_{ki} + b_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k \\ &+ (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{\eta}_{\boldsymbol{w}_k},\end{aligned} \tag{19}$$

where $b_i = \sum_{k=1}^{s} \langle \boldsymbol{J}_{ki}, \boldsymbol{\eta}_{\boldsymbol{w}_k}\boldsymbol{w}_k^{\mathsf{H}} + \boldsymbol{w}_k\boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}} \rangle$.

Due to the structure of symmetric rank-one matrices $\boldsymbol{M}_k = \boldsymbol{w}_k\boldsymbol{w}_k^{\mathsf{H}}$ in problem (13), the cost of computing Riemannian Hessian and Riemannian gradient (16)-(19), i.e., $O(sdL)$, and the computational cost of operations on Riemannian manifold such as projection and retraction, i.e., $O(n)$, are both extremely low. Thus the Riemannian trust-region algorithm is scalable to large-scale problem. It also enjoys fast converge rate with the trust-region method.

IV. SIMULATION RESULTS

In this section, we simulate the propose Riemannian optimization algorithm for blind demixing problem to demonstrate its algorithmic advantages and good performance.

The simulation settings are given follows:

- Ground truth rank-one matrices $\{\hat{\boldsymbol{X}}_k\}_{k=1}^s$: We generate standard complex Gaussian vectors $\boldsymbol{x}_k \in \mathbb{C}^N$ and $\boldsymbol{h}_k \in \mathbb{C}^K$ for $k = 1, \cdots, s$, whose entries are drawn i.i.d from the standard normal distribution. The matrices $\{\hat{\boldsymbol{X}}_k\}_{k=1}^s$ are thus generated as $\{\hat{\boldsymbol{X}}_k\}_{k=1}^s = \{\boldsymbol{x}_k\bar{\boldsymbol{h}}_k^{\mathsf{H}}\}_{k=1}^s$ [15].
- Measurement matrices $\{\{\boldsymbol{J}_{ik}\}_{i=1}^L\}_{k=1}^s$: We generate the complex Gaussian matrix $\boldsymbol{C}_k \in \mathbb{C}^{L\times N}$ for $k = 1, \cdots, s$ and the normalized discrete Fourier transform (DFT) matrix $\boldsymbol{F} \in \mathbb{C}^{L\times L}$ to construct the measurement matrices according to (5) and (11).
- Performance metric: The relative construction error is adopted to evaluate the performance of the algorithms, given as [15] $\sqrt{\sum_{k=1}^s \|\boldsymbol{X}_k - \hat{\boldsymbol{X}}_k\|_F^2} / \sqrt{\sum_{k=1}^s \|\hat{\boldsymbol{X}}_k\|_F^2}$, where $\{\boldsymbol{X}_k\}_{k=1}^s$ are estimated matrices and $\{\hat{\boldsymbol{X}}_k\}$ are ground truth matrices.

The following four algorithms are compared:

- **Proposed Riemannian trust-region algorithm (PRTR)**: We use the manifold optimization toolbox *Manopt* [23] to implement the proposed Riemannian trust-region algorithm (PRTR).
- **Nuclear norm minimization (NNM)**: The algorithm [12] is implemented with the toolbox CVX [24] to solve the convex problem (7) with the parameter $\varepsilon = 10^{-9}$ in noiseless scenario and $\varepsilon = 10^{-2}$ in noisy scenario.
- **Regularized gradient descent (RGD)**: This algorithm [15] is implemented to solve the regularized problem (9).
- **Fast Iterative Hard Thresholding (FIHT)**: The algorithm [16] utilizes hard thresholding to solve the rank-constraint problem $\mathscr{P}$ directly.

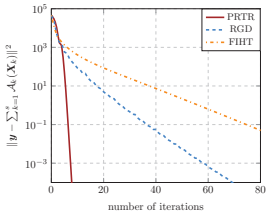

Fig. 3. Convergence rate of different algorithms with respect to the number of iterations.

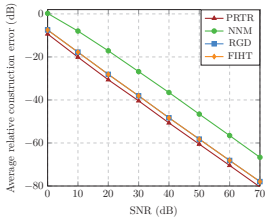

Fig. 4. Average relative construction error vs. SNR (dB).

We adopt the initialization strategy in [16] for all the non-convex optimization algorithms (i.e., PRTR, RGD and FIHT). The PRTR algorithm stops when the norm of Riemannian gradient falls below 10^{-8} or the number of iterations exceeds 500. The stopping criteria of RGD and FIHT are based on [15] and [16], respectively.

Three non-convex algorithms are compared under the noiseless scenario . We set $N = K = 50$, $L = 1250$ and $s = 5$. Fig. 3 illustrates the convergence rate of different non-convex algorithms. In the noisy scenario, we assume the additive noise in formulation (4) satisfies [16] $\boldsymbol{e} = \sigma \cdot \|\boldsymbol{y}\| \cdot \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$, where $\boldsymbol{\omega} \in \mathbb{C}^L$ is a standard complex Gaussian vector. Four algorithms with different signal to noise ratio (SNR) σ are compared. We set the number of measurements $L = 1500$, $N = K = 50$ and the number of users $s = 2$. For every circumstance, 10 independent trails are simulated. The average relative construction error in dB against the signal to noise ratio (SNR) is illustrated in Fig. 4. All simulation results demonstrate the proposed Riemannian trust-region algorithm significantly outperforms the state-of-art algorithms in terms of algorithmic advantages and admirable performance.

V. Conclusion

In this paper, we presented the blind demixing approach to support low-latency communication without any channel estimation, for which a low-rank modeling approach is further developed. To address the unique challenge of *multiple asymmetric complex* variables as well as develop efficient algorithm, we exploited the Riemannian quotient geometry of product of complex asymmetric rank-one matrices via reformulating problems on complex asymmetric matrices to problems on Hermitian PSD matrices. By exploiting the admirable structure of symmetric rank-one matrices, we developed a scaled Riemannian trust-region algorithm with superlinear converging rate and low iteration cost. Simulation results demonstrated that the proposed algorithm is robust to the additive noise and outperforms the state-of-art algorithms in terms of algorithmic advantages and performance.

References

[1] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled Tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Feb. 2016.

[2] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected Virtual Reality: Opportunities, challenges, and enablers," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?," *IEEE J. Sel. Area. Comm.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[4] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *arXiv preprint arXiv:1708.02562*, 2017.

[5] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[6] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[7] Y. Shi, J. Zhang, K. B. Letaief, B. Bai, and W. Chen, "Large-scale convex optimization for ultra-dense cloud-RAN," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 84–91, Jun. 2015.

[8] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.

[9] Y. Shi, J. Zhang, and K. B. Letaief, "Low-rank matrix completion for topological interference management by Riemannian pursuit," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4703–4717, Jul. 2016.

[10] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[11] S. Ling and T. Strohmer, "Blind deconvolution meets blind demixing: Algorithms and performance bounds," *IEEE Trans. Inf. Theory*, May 2017.

[12] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1711–1732, Mar. 2014.

[13] M. B. McCoy and J. A. Tropp, "Sharp recovery bounds for convex demixing, with applications," *Found. Comput. Math.*, vol. 14, no. 3, pp. 503–567, Jun. 2014.

[14] M. B. McCoy and J. A. Tropp, "The achievable performance of convex demixing," *ACM Report 2017-02, Caltech*, Feb. 2017.

[15] S. Ling and T. Strohmer, "Regularized gradient descent: A nonconvex recipe for fast joint blind deconvolution and demixing," *arXiv preprint arXiv:1703.08642*, 2017.

[16] T. Strohmer and K. Wei, "Painless breakups–efficient demixing of low rank matrices," *arXiv preprint arXiv:1703.09848*, 2017.

[17] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, May 2005.

[18] A. Goldsmith, *Wireless communications*. Cambridge University Press, Aug. 2005.

[19] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, Apr. 2009.

[20] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM J. Optimiz.*, vol. 23, no. 2, pp. 1214–1236, Jun. 2013.

[21] S. Yatawatta, "Radio interferometric calibration using a Riemannian manifold," in *Proc. IEEE Int. Conf. Acoustics Speech Sig. Process. (ICASSP), 2013*, pp. 3866–3870, May 2013.

[22] R. Ge, C. Jin, and Y. Zheng, "No spurious local minima in nonconvex low rank problems: A unified geometric analysis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 70, pp. 1233–1242, Aug. 2017.

[23] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, Jan. 2014.

[24] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008.