MMSE-Optimal Sequential Processing for Cell-Free Massive MIMO With Radio Stripes

Zakir Hussain Shaik, *Student Member, IEEE*, Emil Björnson, *Senior Member, IEEE*, and
Erik G. Larsson, *Fellow, IEEE*

***Abstract*—Cell-free massive multiple-input-multiple-output (mMIMO) is an emerging technology for beyond 5G with its promising features such as higher spectral efficiency and superior spatial diversity as compared to conventional multiple-input-multiple-output (MIMO) technology. The main working principle of cell-free mMIMO is that many distributed access points (APs) cooperate simultaneously to serve all the users within the network without creating cell boundaries. This paper considers the uplink of a cell-free mMIMO system utilizing the radio stripe network architecture with a sequential fronthaul between the APs. A novel uplink sequential processing algorithm is developed, which is proved to be optimal in both the maximum spectral efficiency (SE) and the minimum mean square error (MSE) sense. A detailed quantitative analysis of the fronthaul requirement or signaling of the proposed algorithm and its comparison with competing sub-optimal algorithms is provided. Key conclusions and implications are summarized in the form of corollaries. Based on the analytical and numerical simulation results, we conclude that the proposed scheme can significantly reduce the fronthaul signaling, without compromising the communication performance.**

***Keywords*—**Beyond 5G, radio stripes, cell-free massive MIMO, uplink, spectral efficiency, mean square error, sequential processing.

I. Introduction

Massive multiple-input-multiple-output (mMIMO) networks, with their manifold benefits over conventional multiple-input-multiple-output (MIMO) networks [2]–[4] such as high spatial resolution and very high-spectral efficiency (SE), have garnered intense interest in the past decade, making it a reality in the year 2018 [5], [6]. Nevertheless, mMIMO in its original form suffers from large signal-to-noise ratio (SNR) variations between cell center and cell-edge users. This problem can be tackled to some extent by small-cell networks [7], but again they suffer from high inter-cell interference due to its inherent cell-centric implementation. Hence, there's a need for a paradigm shift of networks from cellular to cell-free. This can be achieved by the more recently evolving idea of a cell-free mMIMO network which is essentially a decentralized implementation of mMIMO [7]–[9].

A cell-free mMIMO network consists of a central processing unit (CPU) connected to a set of access points (APs) which jointly serve all the user equipments (UEs) in the network. An AP consists of antennas and the signal processing units required to operate them locally. The original idea of cell-free mMIMO networks was to have a dedicated fronthaul and power supply to every AP running up to the CPU (e.g., a star topology) [7]. In the uplink, each AP after receiving the pilot and data signals, forwards them to the CPU, where final fusion of data is done, and the information signals are decoded. However, this network topology requires a large fronthaul capacity from the APs to the CPU, which directly impacts the amount of data to be managed and processed at the CPU. For a wired implementation, a long cable is required to be connected between each AP and the CPU, making the cost the main bottleneck for practical implementation. These factors limit its practicability and necessitate the search for more practical architectures that can decentralize this processing and reduce the fronthaul signaling.

In the literature, there have been various techniques and algorithms that are developed for decentralizing the signal processing in mMIMO systems [10]–[16], which can be adapted to use in cell-free mMIMO implementation. The network topologies of interest include sequential (daisy-chain network), centralized (star-like network), tree, and fully connected (mesh network). The choice of a particular topology depends on the application of interest. One promising direction for cell-free networks is radio stripes, which utilize the sequential topology [1], [17]. This architecture is suitable for deployments in dense areas such as sports arenas and railway stations with many APs and UEs per km^2, and large construction elements that the stripes can be attached to. This network comprises sequentially connected APs in a daisy chain topology and shares the same cable for fronthaul and power supply, as illustrated in Fig. 1. A few benefits of a sequential network are: (i) ease of deployment and cable routing in practical applications such as railway stations, museums, factories, etc., (ii) the number of links connecting to each AP (including connections to the CPU) is fixed, and it is two, i.e., one for sending and the other for receiving signals, and (iii) node failures can be mitigated by using routing mechanisms making the network fault-tolerant. In contrast, other topologies have at least one AP or the CPU with more than one transmit or receive links which could add additional complexity in implementation and timing synchronization, also the design of fault-tolerant

Preliminary results on a precursor to proposed algorithm were presented in IEEE ICC 2020 [1]. E. Björnson was supported by the Grant 2019-05068 from the Swedish Research Council. Erik G. Larsson was supported in part by the Swedish Research Council (VR) with Grant 642-2013-7607, ELLIIT and the Knut and Alice Wallenberg (KAW) Foundation The authors are with the Department of Electrical Engineering (ISY), Linköping University, SE-58183 Linköping, Sweden. E. Björnson is also with the Department of Computer Science, KTH Royal Institute of Technology, SE-10044, Stockholm, Sweden. Emails: {zakir.hussain.shaik, emil.bjornson, erik.g.larsson}@liu.se.

routing mechanisms may be challenging depending on the topology, [17], [18]. Moreover, the sequential implementation of cell-free mMIMO has the potential to deliver the benefits of mMIMO with much lower fronthaul requirements when compared to centralized. In this paper, we develop an optimal uplink processing algorithm for radio stripes in the sense of maximum SE and minimum mean square error (MSE).

A. Related Work

The decentralization methods found in the literature can be broadly categorized into two types: fully centralized and fully distributed implementation. In the former category, all the required processing is done at the CPU [7], [9], [11], while in the latter category, all the processing is done locally at the APs [1], [10], [18], [19], except for the final fusion at the CPU, using statistical channel state information (CSI). A centralized implementation has superior performance because of its access to complete information but, the downside is the requirement of a very large fronthaul compared to other topologies such as sequential processing. There are a few other methods that do not strictly fall into either of the above categories. The method proposed in [10] is partially decentralizing the process, i.e., the APs partially process the received signals and forward the Gramian of the channel matrix to the CPU, where most of the signal processing and estimation of the signal is done. The authors have shown analytically that this strategy achieves the same performance as linear methods such as maximum ratio combining (MRC), zero forcing (ZF), and linear minimum mean square error (LMMSE). However, the authors have only presented the analysis for the perfect CSI case.

The relevant works which focused on developing algorithms for a sequential network for mMIMO are [1], [10], [19]. The authors in [19] proposed algorithms which decentralize ZF in a sequential manner but only for the perfect CSI case. Moreover, SE analysis is not investigated in [10] and [19].

B. Contribution

The main contributions of this paper are:

(i) We develop a novel uplink sequential processing algorithm which is proved to be optimal in both the SE and MSE sense.

(ii) We provide closed-form expressions for the SE and MSE of the proposed method and also for any sequential linear processing algorithm.

(iii) We prove that the ordering of the APs has no impact on the performance when using the proposed algorithm.

(iv) We provide update expressions for the SE and MSE when adding additional APs.

(v) We quantify and compare the fronthaul requirements of the proposed algorithm with competing algorithms [1], [19].

(vi) To address the latency issue that appears in long sequential networks, we provide a semi-distributed algorithm without loss in performance.

(vii) We provide numerical and simulation results comparing the proposed algorithm with the existing algorithms for sequential and centralized cell-free networks.

In the conference version [1], we proposed an algorithm that had a trade-off between the SE and fronthaul signaling when compared to a centralized implementation [9] with LMMSE receiver. Ideally, we would like to achieve the performance of a centralized scheme receiver but most of the fully decentralized algorithms fall short of achieving the optimal SE. This paper proposes a new algorithm that achieves the same performance as the optimal centralized implementation, while reducing the fronthaul requirement, thus the trade-off issue is resolved.

C. Paper Outline

The remainder of this paper is organized as follows. Section II presents the system model for uplink cell-free networks with a sequential fronthaul for both payload transmission and channel estimation. Section III presents most of the contributions, including an optimal algorithm for sequential signal processing and closed-form expressions of the SE and MSE achieved by the proposed method. In Section IV, a quantitative analysis of the fronthaul capacity requirements of the proposed method and other competing algorithms is provided, and also latency for a semi-distributed algorithm is analyzed. Numerical results are presented and analyzed in Section V. Finally, the main conclusions of this paper are presented in Section VI. Appendix includes the proof of the main theorem.

Reproducible research: All the simulation results can be reproduced using the Matlab code and data files available at: https://github.com/emilbjornson/radio-stripes

D. Notations

Boldface lowercase letters, $\mathbf{a}$, denote column vectors and boldface uppercase letters, $\mathbf{A}$, denote matrices. The superscripts $(\cdot)^*$, $(\cdot)^T$, and $(\cdot)^H$ denote the conjugate, transpose, and Hermitian transpose, respectively. The notation $\mathbf{I}_N$ represents the $N \times N$ identity matrix. The (m,n)th element of a matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{mn}$. A block-diagonal matrix is represented by $\mathrm{diag}(\mathbf{A}_1, \cdots, \mathbf{A}_N)$ for square matrices $\mathbf{A}_1, \cdots, \mathbf{A}_N$. The absolute value of a scalar and l_2-norm of a vector are denoted by $|\cdot|$ and $\|\cdot\|$, respectively. The real value of a scalar is denoted by $\Re\{\cdot\}$. We denote the expectation and variance by $\mathbb{E}\{\cdot\}$ and $\mathrm{Var}\{\cdot\}$, respectively. We use $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C})$ to denote a multi-variate circularly symmetric complex Gaussian random vector with zero mean and covariance matrix $\mathbf{C}$.

II. System Model And Channel Estimation

We consider a cell-free mMIMO network comprising L APs, each equipped with N antennas. The fronthaul connections are assumed to go from AP 1 to AP 2 $\cdots$ to AP L to the CPU as shown in the Fig. 1. This architecture with a sequential fronthaul is called a radio stripe network. There are K UEs, each with a single antenna, distributed arbitrarily in the network. We consider the standard block fading channel model with coherence block length of τ_c channel uses [20]. The channel between AP l and UE k is denoted by $\mathbf{h}_{kl} \in \mathbb{C}^N$.

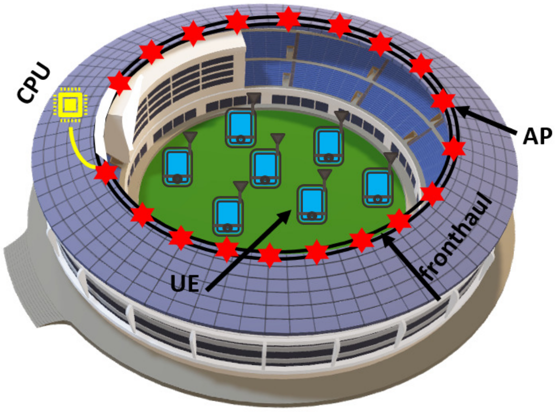


Figure 1: Radio stripes network deployed over a football arena.

In each block, an independent realization is drawn from a correlated Rayleigh fading distribution as

$$\mathbf{h}_{kl} \sim \mathcal{CN}\left(\mathbf{0}, \mathbf{R}_{kl}\right), \tag{1}$$

where $\mathbf{R}_{kl} \in \mathbb{C}^{N\times N}$ is the spatial correlation matrix, which attributes the spatial channel correlation characteristics and large-scale fading. The large-scale fading coefficient describing the shadowing and pathloss is given by $\beta_{kl} \triangleq \text{tr}\left(\mathbf{R}_{kl}\right)/N$. The spatial correlation matrices $\{\mathbf{R}_{kl}\}$ are assumed to be known at all the APs and the CPU.

This paper studies an uplink scenario where each coherence block consists of τ_p channel uses for pilot transmission to estimate the channels and $\tau_c - \tau_p$ channel uses for payload data. Both phases are described in detail below.

A. Channel Estimation

We assume there are τ_p mutually orthogonal τ_p-length pilot signals $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_{\tau_p}$ with $\|\boldsymbol{\phi}_k\|^2 = \tau_p$, which are used for channel estimation. We are mainly interested in the case $K > \tau_p$, where more than one UE is assigned the same pilot causing pilot contamination. We let the pilot assigned to UE k, for $k = 1, \ldots, K$, be denoted by $t_k \in \{1, \ldots, \tau_p\}$ and the set $\mathcal{S}_k = \{i : t_i = t_k\}$ accounts for those UEs that are assigned the same pilot as UE k. The received signal $\mathbf{Y}_l^p \in \mathbb{C}^{N\times\tau_p}$ at AP l is

$$\mathbf{Y}_l^p = \sum_{i=1}^{K} \sqrt{p_i}\mathbf{h}_{il}\boldsymbol{\phi}_{t_i}^T + \mathbf{N}_l, \tag{2}$$

where $p_i \geq 0$ is the transmit power of UE i and $\mathbf{N}_l \in \mathbb{C}^{N\times\tau_p}$ is the noise at the receiver modeled with independent entries distributed as $\mathcal{CN}\left(0, \sigma^2\right)$ with σ^2 being the noise power. The estimation of $\mathbf{h}_{kl}$ at AP l proceeds in two phases: first despreading of the received signal is done and then the MMSE estimator is employed. Accordingly, the MMSE channel estimate $\widehat{\mathbf{h}}_{kl} \in \mathbb{C}^{N\times 1}$ is given by [21]

$$\widehat{\mathbf{h}}_{kl} = \sqrt{p_k\tau_p}\mathbf{R}_{kl}\boldsymbol{\Psi}_{t_kl}^{-1}\mathbf{y}_{t_kl}^p, \tag{3}$$

where

$$\mathbf{y}_{t_kl}^p = \mathbf{Y}_l^p\frac{\boldsymbol{\phi}_{t_k}^*}{\sqrt{\tau_p}} = \sum_{i\in\mathcal{S}_k}\sqrt{p_i\tau_p}\mathbf{h}_{il} + \mathbf{n}_{t_kl}, \tag{4}$$

$$\boldsymbol{\Psi}_{t_kl} = \sum_{i\in\mathcal{S}_k}\tau_p p_i\mathbf{R}_{il} + \sigma^2\mathbf{I}_N \tag{5}$$

are the despreaded signal and its covariance matrix, respectively. Here, $\mathbf{n}_{t_kl} \triangleq \mathbf{N}_l\boldsymbol{\phi}_{t_k}^*/\sqrt{\tau_p} \sim \mathcal{CN}\left(\mathbf{0}, \sigma^2\mathbf{I}_N\right)$ is the effective noise. An important consequence of MMSE estimation is that the estimate $\widehat{\mathbf{h}}_{kl} \sim \mathcal{CN}(\mathbf{0}, \widehat{\mathbf{R}}_{kl})$ and the estimation error $\widetilde{\mathbf{h}}_{kl} = \mathbf{h}_{kl} - \widehat{\mathbf{h}}_{kl} \sim \mathcal{CN}(\mathbf{0}, \widetilde{\mathbf{R}}_{kl})$ are uncorrelated, where $\widehat{\mathbf{R}}_{kl} = p_k\tau_p\mathbf{R}_{kl}\boldsymbol{\Psi}_{t_kl}^{-1}\mathbf{R}_{kl}$ and $\widetilde{\mathbf{R}}_{kl} = \mathbf{R}_{kl} - \widehat{\mathbf{R}}_{kl}$ are the respective covariance matrices. A note on the choice of the pilot signals, that the performance with the proposed algorithms will be the same as the optimal centralized implementation irrespective of the pilots being orthogonal (as presented above) or non-orthogonal and there will not be errors in the processing when compared to the state-of-the-art centralized implementation. Moreover, the paper considers the case $K > \tau_p$ with pilot contamination, thus it is a non-orthogonal scheme both in the channel estimation and data transmission. This standard approach can be refined using power control and a larger set of partially overlapping pilots.

B. Uplink Payload Transmission

During the uplink payload transmission, the received signal $\mathbf{y}_l \in \mathbb{C}^N$ at AP l is given by

$$\mathbf{y}_l = \mathbf{H}_l\mathbf{s} + \mathbf{n}_l, \tag{6}$$

where $\mathbf{H}_l = [\mathbf{h}_{1l}, \mathbf{h}_{2l}, \cdots, \mathbf{h}_{Kl}] \in \mathbb{C}^{N\times K}$ is the channel matrix, $\mathbf{s} = [s_1, s_2, \cdots, s_K]^T \in \mathbb{C}^K$ is the signal vector with $s_k \sim \mathcal{CN}\left(0, p_k\right)$ being the payload signal transmitted by UE k with power p_k and the receiver noise vector $\mathbf{n}_l \sim \mathcal{CN}\left(\mathbf{0}, \sigma^2\mathbf{I}_N\right)$. We assume s_k is independent of s_m for $k \neq m$ and the signal vector is distributed as $\mathbf{s} \sim \mathcal{CN}\left(\mathbf{0}, \mathbf{Q}\right)$ with $\mathbf{Q} = \text{diag}(p_1, \cdots, p_K)$. Let $\mathbf{H}_l = \widehat{\mathbf{H}}_l + \widetilde{\mathbf{H}}_l$ with $\widehat{\mathbf{H}}_l = [\widehat{\mathbf{h}}_{1l}, \widehat{\mathbf{h}}_{2l}, \cdots \widehat{\mathbf{h}}_{Kl}]$ being the matrix with channel estimates and $\widetilde{\mathbf{H}}_l = [\widetilde{\mathbf{h}}_{1l}, \widetilde{\mathbf{h}}_{2l}, \cdots \widetilde{\mathbf{h}}_{Kl}]$ the matrix with estimation errors. Accordingly, (6) is equivalent to

$$\begin{aligned}\mathbf{y}_l &= \widehat{\mathbf{H}}_l\mathbf{s} + \widetilde{\mathbf{H}}_l\mathbf{s} + \mathbf{n}_l \\ &= \widehat{\mathbf{H}}_l\mathbf{s} + \mathbf{w}_l,\end{aligned} \tag{7}$$

where $\mathbf{w}_l = \widetilde{\mathbf{H}}_l\mathbf{s} + \mathbf{n}_l$ can be thought of as a colored noise vector with zero mean and covariance matrix

$$\boldsymbol{\Sigma}_l = \sum_{i=1}^{K} p_i\widetilde{\mathbf{R}}_{il} + \sigma^2\mathbf{I}_N. \tag{8}$$

Note that $\mathbf{s}$ is uncorrelated with $\mathbf{w}_l$ but is statistically dependent. We denote the estimate of $\mathbf{s}$ at AP l as $\widehat{\mathbf{s}}_l = [\widehat{s}_{1l}, \widehat{s}_{2l}, \cdots, \widehat{s}_{Kl}]^T$ with $\widehat{s}_{kl}$ being the estimate of s_k. We will analyze different ways to compute the estimate in the next section.

III. Sequential Uplink Processing

In this section, we derive an optimal sequential receiver algorithm in the sense of simultaneously achieving the maximum SE and the minimum MSE at the CPU. We first briefly describe the optimal receiver for the centralized cell-free mMIMO network [9], [22] which is a baseline for performance. We introduce the following notation which is utilized throughout this paper:

$$\mathbf{z}_l = \widehat{\mathbf{G}}_l \mathbf{s} + \overline{\mathbf{w}}_l \tag{9}$$

where $\mathbf{z}_l = [\mathbf{y}_1^H, \cdots, \mathbf{y}_l^H]^H, \widehat{\mathbf{G}}_l = [\widehat{\mathbf{H}}_1^H, \cdots, \widehat{\mathbf{H}}_l^H]^H, \overline{\mathbf{w}}_l = [\mathbf{w}_1^H, \cdots, \mathbf{w}_l^H]^H$; where $\mathbf{z}_l$ can be thought of as the augmented received signal at AP $l \in \{1, \cdots, L\}$, $\widehat{\mathbf{G}}_l$ contains the augmented channel estimates and $\overline{\mathbf{w}}_l$ being the augmented colored noise vector with zero mean and covariance matrix $\mathbf{K}_l = \text{diag}(\mathbf{\Sigma}_1, \cdots, \mathbf{\Sigma}_l)$.

A. Optimal Centralized Implementation

In a centralized setup, all the L APs send the received pilot and payload signals to the CPU which after computing channel estimates forms the following augmented received signal:

$$\mathbf{z}_L = \widehat{\mathbf{G}}_L \mathbf{s} + \overline{\mathbf{w}}_L. \tag{10}$$

The CPU employs the LMMSE receiver which is optimal in both the maximum SE and the minimum MSE sense to compute the data signal estimate as [9]

$$\widehat{\mathbf{s}}_L^c = \mathbf{V}_L^c \mathbf{z}_L, \tag{11}$$

where $\widehat{\mathbf{s}}_L^c$ is the signal estimate, $\mathbf{V}_L^c$ is the receive combining matrix and "c" indicates that it is a centralized scheme. The LMMSE receiver matrix is given as [21]

$$\mathbf{V}_L^c = \mathbf{Q}\widehat{\mathbf{G}}_L^H \mathbf{\Lambda}_L^{-1}, \tag{12}$$

where

$$\mathbf{\Lambda}_L = \left(\mathbf{K}_L + \widehat{\mathbf{G}}_L \mathbf{Q} \widehat{\mathbf{G}}_L^H\right). \tag{13}$$

The achievable SE of UE k using the LMMSE receiver is given by [9]

$$\text{SE}_k^c = \left(1 - \frac{\tau_p}{\tau_c}\right) \mathbb{E}\left\{\log_2\left(1 + \Gamma_k^c\right)\right\}, \tag{14}$$

where the instantaneous effective signal-to-interference-and-noise ratio (SINR) is

$$\Gamma_k^c = p_k \widehat{\mathbf{h}}_k^H \left(\sum_{i=1, i\neq k}^{K} p_i \widehat{\mathbf{h}}_i \widehat{\mathbf{h}}_i^H + \mathbf{K}_L\right)^{-1} \widehat{\mathbf{h}}_k. \tag{15}$$

The minimum MSE at the CPU is

$$\begin{aligned} \mathbf{P}_L^c &= \mathbb{E}\left\{\left(\mathbf{s} - \widehat{\mathbf{s}}_L^c\right)\left(\mathbf{s} - \widehat{\mathbf{s}}_L^c\right)^H \mid \widehat{\mathbf{G}}_L\right\} \\ &= \mathbf{Q} - \mathbf{V}_L^c \widehat{\mathbf{G}}_L \mathbf{Q}. \end{aligned} \tag{16}$$

Now we will briefly discuss about other common linear receivers such as MRC and ZF: The MRC receiver has low complexity than the LMMSE receiver, and ZF has comparable complexity as the LMMSE receiver, however, the achievable SE of any linear receiver such as MRC or ZF will be suboptimal to the LMMSE receiver [9]. Thus, there is a tradeoff between complexity and performance when the LMMSE receiver is used over the MRC receiver. In this work, we aim to develop an algorithm with maximum achievable SE, and hence we focus on the LMMSE receiver.

B. Sequential Linear Processing

In this subsection, we present a generic class of linear processing algorithms suitable for sequential cell-free mMIMO networks. In general, a sequential processing algorithm starts with AP 1 computing the soft estimate $\widehat{\mathbf{s}}_1$ of the signal $\mathbf{s}$ and forwards the computed estimate and also some useful side information to AP 2 which would help it in making a better estimate. Now, AP $l \in \{2, \cdots, L\}$ upon receiving the information from AP $(l-1)$ computes the soft estimate $\widehat{\mathbf{s}}_l$ of the signal $\mathbf{s}$ and then it forwards the computed estimate along with some useful side information (side information is algorithm dependent which can be a function of channel estimates, channel and signal statistics) to AP $(l+1)$. This sequential process continues till AP L which forwards the final estimate $\widehat{\mathbf{s}}_L$ of the signal to the CPU. The CPU might also be co-located with AP L.

Beginning with AP 1, the soft estimate $\widehat{\mathbf{s}}_1$ of $\mathbf{s}$ computed by AP 1 is given by

$$\widehat{\mathbf{s}}_1 = \mathbf{B}_1 \mathbf{y}_1, \tag{17}$$

where $\mathbf{B}_1 \in \mathbb{C}^{K\times N}$ is any receiver combining matrix that can be selected based on the information available at AP 1 (e.g., channel estimate matrix, channel and noise statistics). AP 1 then forwards the estimate (17) along with other useful side information to AP 2. Then, AP $l \in \{2, \ldots, L\}$ upon receiving the estimate $\widehat{\mathbf{s}}_{(l-1)}$ and other side information from AP $(l-1)$ computes its estimate as follows:

$$\widehat{\mathbf{s}}_l = \mathbf{A}_l \widehat{\mathbf{s}}_{(l-1)} + \mathbf{B}_l \mathbf{y}_l, \quad l \in \{2, \cdots, L\} \tag{18}$$

where $\mathbf{A}_l \in \mathbb{C}^{K\times K}$ and $\mathbf{B}_l \in \mathbb{C}^{K\times N}$ are some receiver combining matrices which depend on the information available at AP l. Considering the initial estimate $\widehat{\mathbf{s}}_0$ as the zero vector, the computation in (18) can be generalized to any AP $l \in \{1, \ldots, L\}$. We now present the achievable SE and the MSE at the CPU for any generic sequential linear processing of the form (18). First note that, (18) can be equivalently written as

$$\widehat{\mathbf{s}}_l = \overline{\mathbf{B}}_l \mathbf{z}_l \tag{19}$$

where,

$$\overline{\mathbf{B}}_l = \begin{cases} \begin{bmatrix} \mathbf{A}_l \overline{\mathbf{B}}_{(l-1)} & \mathbf{B}_l \end{bmatrix}, & l > 1 \\ \mathbf{B}_1, & l = 1. \end{cases} \tag{20}$$

The following two propositions provide closed-form expressions for the achievable SE and the MSE at the CPU, respectively. Let the augmented receiver matrix in (20) at the AP L be

$$\overline{\mathbf{B}}_L = [\mathbf{b}_1, \cdots, \mathbf{b}_K]^H \tag{21}$$

with $\mathbf{b}_k$ being the combining vector for kth UE and $\widehat{\mathbf{h}}_k$ be the kth column of channel estimation matrix $\widehat{\mathbf{G}}_L$ or equivalently

$\widehat{\mathbf{h}}_k = [\widehat{\mathbf{h}}_{k1}^H, \cdots, \widehat{\mathbf{h}}_{kL}^H]^H$ at the final AP L. Thus, note the kth UE estimate $\widehat{s}_{kL}$ at the CPU is a function of $\mathbf{b}_k$ i.e., $\widehat{s}_{kL}(\mathbf{b}_k)$.

Proposition 1 *The achievable SE of UE k for any receiver algorithm of the form* (18) *is*

$$\mathrm{SE}_k(\mathbf{b}_k) = \left(1 - \frac{\tau_p}{\tau_c}\right) \mathbb{E}\left\{\log_2\left(1 + \Gamma_k^{'}(\mathbf{b}_k)\right)\right\}, \qquad (22)$$

where the instantaneous effective SINR is

$$\Gamma_k^{'}(\mathbf{b}_k) = \frac{p_k |\mathbf{b}_k^H \widehat{\mathbf{h}}_k|^2}{\sum_{i=1, i\neq k}^{K} p_i |\mathbf{b}_k^H \widehat{\mathbf{h}}_i|^2 + \mathbf{b}_k^H \mathbf{K}_L \mathbf{b}_k}. \qquad (23)$$

Proof: The proof follows from Proposition 1 in [9]. ■

Proposition 2 *The MSE of UE k at the CPU for any receiver algorithm of the form* (18) *for the given side information including channel estimates is*

$$\begin{aligned} e_k^{'}(\mathbf{b}_k) &= \min_{\mathbf{b}_k} \mathbb{E}\{|s_k - \widehat{s}_{kL}(\mathbf{b}_k)|^2 | \text{side information}\} \\ &= p_k - 2\Re\{\mathbf{b}_k^H \widehat{\mathbf{h}}_k\} p_k + \mathbf{b}_k^H \mathbf{\Lambda}_L \mathbf{b}_k. \end{aligned} \qquad (24)$$

One special case of the sequential process in (18) is MR, which is obtained when the receiver matrix $\mathbf{A}_l$ is an identity matrix and

$$\mathbf{B}_l = [\widehat{\mathbf{h}}_{1l}, \cdots, \widehat{\mathbf{h}}_{Kl}]^H. \qquad (25)$$

One obtains the following estimate at AP L:

$$\widehat{s}_{kL} = \sum_{l=1}^{L} \widehat{\mathbf{h}}_{kl}^H \mathbf{y}_l = \widehat{\mathbf{h}}_k^H \mathbf{z}_L. \qquad (26)$$

C. Optimal Sequential Linear Processing (OSLP)

In this subsection, we provide the choice of receiver combining matrices $\{\mathbf{A}_l\}$ and $\{\mathbf{B}_l\}$ among the class of generic sequential receivers that jointly maximize (22) and minimize (24) i.e., we present a particular sequential algorithm of the form (18) which we will show that it achieves the same optimal performance as centralized LMMSE but with lower fronthaul requirements.

We begin by providing a geometric motivation for the choice of matrices $\{\mathbf{A}_l, \mathbf{B}_l\}$ with two APs to achieve the performance of centralized LMMSE receiver, i.e., achieve minimum MSE. Then, we provide the general expressions for $\{\mathbf{A}_l, \mathbf{B}_l\}$ for $l \in \{1, \ldots, L\}$. Beginning with AP 1, with the available side information $\{\mathbf{y}_l, \widehat{\mathbf{H}}_1, \mathbf{Q}\}$, the receiver which achieves the minimum MSE is the standard LMMSE receiver, and the corresponding estimate is given by

$$\widehat{\mathbf{s}}_1 = \mathbf{R}_{\mathbf{s}\mathbf{y}_1} \mathbf{R}_{\mathbf{y}_1\mathbf{y}_1}^{-1} \mathbf{y}_1, \qquad (27)$$

where $\mathbf{R}_{\mathbf{xy}} = \mathbb{E}\{\mathbf{x}\mathbf{y}^H\}$ is the correlation matrix between two arbitrary random vectors $\mathbf{x}$ and $\mathbf{y}$. From this, we observe that optimal $\mathbf{A}_1 = \mathbf{0}$ and $\mathbf{B}_1 = \mathbf{Q}\widehat{\mathbf{H}}^H \left(\mathbf{\Sigma}_l + \widehat{\mathbf{H}}\mathbf{Q}\widehat{\mathbf{H}}^H\right)^{-1}$. Note that from the orthogonality property of LMMSE estimator, the LMMSE estimation error is orthogonal to the data vector and we use this property to obtain optimal $\mathbf{A}_2$ and $\mathbf{B}_2$ based on $\mathbf{y}_2$. To this end, it is required to remove the extra information that is contained in $\mathbf{y}_2$ about $\widehat{\mathbf{s}}_1$, i.e., extract only that component of vector $\mathbf{y}_2$ that is orthogonal to $\mathbf{y}_1$ (as is required with LMMSE receiver with $(\mathbf{y}_1, \mathbf{y}_2)$ as the received signals). This is obtained in two steps: first obtaining the LMMSE estimate of $\mathbf{y}_2$, denoted by $\widehat{\mathbf{y}}_2$, given $\mathbf{y}_1$ and then it follows that the estimation error $\widetilde{\mathbf{y}}_2 = \mathbf{y}_2 - \widehat{\mathbf{y}}_2$ is orthogonal to $\mathbf{y}_1$; and then obtain the LMMSE estimate of $\mathbf{s}$ with $\widetilde{\mathbf{y}}_2$ as the available data. It is worth noting that $\widehat{\mathbf{y}}_2 = \mathbf{R}_{\mathbf{y}_2\mathbf{y}_1} \mathbf{R}_{\mathbf{y}_1\mathbf{y}_1}^{-1} \mathbf{y}_1 = \widehat{\mathbf{H}}_2 \mathbf{R}_{\mathbf{s}\mathbf{y}_1} \mathbf{R}_{\mathbf{y}_1\mathbf{y}_1}^{-1} \mathbf{y}_1 = \widehat{\mathbf{H}}_2 \widehat{\mathbf{s}}_1$. Now computing the LMMSE receiver using $\widetilde{\mathbf{y}}_2$ and doing some mathematical manipulations gives the receiver at AP 2 as $\mathbf{T}_2 = \mathbf{P}_1 \widehat{\mathbf{H}}_2^H \left(\mathbf{\Sigma}_2 + \widehat{\mathbf{H}}_2 \mathbf{P}_1 \widehat{\mathbf{H}}_2^H\right)^{-1}$ where $\mathbf{P}_1$ is the MSE matrix at AP 1 with its estimate $\widehat{\mathbf{s}}_1$. Thus, the matrices at AP 2 are $\mathbf{A}_2 = \mathbf{I}_K - \mathbf{T}_2 \widehat{\mathbf{H}}_2$ and $\mathbf{B}_2 = \mathbf{T}_2$. This is illustrated in Fig. 2, where $\mathbf{T}_1 = \mathbf{B}_1$ and $\mathbf{e} = \mathbf{s} - \widehat{\mathbf{s}}_2$. From the geometrical interpretation, it is clear that the minimum side information required at AP 2 from AP 1 is $\mathbf{P}_1$. This idea can be generalized to L APs.

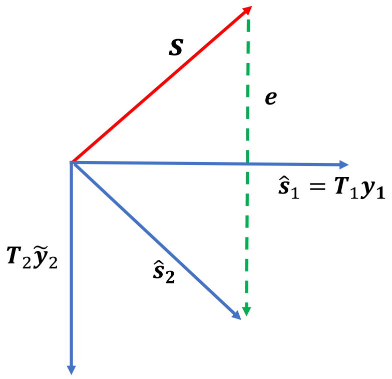

$\boldsymbol{s}$
$\boldsymbol{e}$
$\widehat{\boldsymbol{s}}_1 = \boldsymbol{T}_1 \boldsymbol{y}_1$
$\boldsymbol{T}_2 \widetilde{\boldsymbol{y}}_2$
$\widehat{\boldsymbol{s}}_2$

Figure 2: Geometric interpretation of the OSLP algorithm.

By generalizing the above framework, in the proposed algorithm, the receiver combining matrices are chosen as follows:

$$\mathbf{A}_l = \mathbf{I}_K - \mathbf{T}_l \widehat{\mathbf{H}}_l \qquad (28)$$

$$\mathbf{B}_l = \mathbf{T}_l, \qquad (29)$$

where

$$\mathbf{T}_l = \mathbf{P}_{(l-1)} \widehat{\mathbf{H}}_l^H \left(\mathbf{\Sigma}_l + \widehat{\mathbf{H}}_l \mathbf{P}_{(l-1)} \widehat{\mathbf{H}}_l^H\right)^{-1} \qquad (30)$$

and

$$\mathbf{P}_{(l-1)} = \left(\mathbf{I}_K - \mathbf{T}_{(l-1)} \widehat{\mathbf{H}}_{(l-1)}\right) \mathbf{P}_{l-2}, \text{ with } \mathbf{P}_0 = \mathbf{Q}. \qquad (31)$$

Thus, substituting (28) and (29) in (18) gives soft estimate at AP l as

$$\widehat{\mathbf{s}}_l = \widehat{\mathbf{s}}_{(l-1)} + \mathbf{T}_l \left(\mathbf{y}_l - \widehat{\mathbf{H}}_l \widehat{\mathbf{s}}_{(l-1)}\right). \qquad (32)$$

We will prove later that $\mathbf{T}_l$ is an optimal local LMMSE receiver i.e., it minimizes

$$\mathbb{E}\left\{\|\mathbf{s} - \widehat{\mathbf{s}}_l\|^2 \mid \widehat{\mathbf{H}}_l, \ \widehat{\mathbf{s}}_{(l-1)}, \mathbf{P}_{(l-1)}\right\} \qquad (33)$$

and $\mathbf{P}_l$ is the error covariance matrix at AP l i.e.,

$$\mathbf{P}_l = \mathbb{E}\left\{(\mathbf{s} - \widehat{\mathbf{s}}_l)(\mathbf{s} - \widehat{\mathbf{s}}_l)^H \mid \widehat{\mathbf{H}}_l, \ \widehat{\mathbf{s}}_{(l-1)}, \mathbf{P}_{(l-1)}\right\}. \qquad (34)$$

Algorithm 1 Optimal Sequential Linear Processing (OSLP) for Radio Stripe

```
1. Initialize: $\widehat{\mathbf{s}}_0 = \mathbf{0}$;
2. Compute (once in every coherence block): $\mathbf{T}_{l'}$, $\forall\, l' = 1, \cdots, L$ according to (30);
3. for $l = 1 : L$
       $\widehat{\mathbf{s}}_l = \widehat{\mathbf{s}}_{(l-1)} + \mathbf{T}_l(\mathbf{y}_l - \widehat{\mathbf{H}}_l\widehat{\mathbf{s}}_{(l-1)})$
   end
4. Output: $\widehat{\mathbf{s}}_L$
```

After AP L computes the final estimate, it forwards the estimate $\widehat{\mathbf{s}}_L$ to the CPU where the final decoding of the signal is done. Apart from the signal estimate, the side information forwarded in the proposed OSLP algorithm from AP l to AP $(l+1)$ is the error covariance matrix $\mathbf{P}_l$. The above described algorithm is presented in the form of a pseudo-code in Algorithm 1. Observe that, inverse terms in the proposed algorithm are computed only once in every coherence block and each AP $l \in \{1, \cdots, L\}$ can simultaneously compute $\mathbf{T}_l\widehat{\mathbf{H}}_l$ and $\left(\mathbf{I}_K - \mathbf{T}_l\widehat{\mathbf{H}}_l\right)$ once for $\tau_c - \tau_p$ channel uses. So in each $\tau_c - \tau_p$ channel use, every AP just has to perform multiplication and addition of signals and forward to the consecutive AP.

We will now show that the proposed algorithm is optimal in the sense of minimizing the MSE (i.e., showing that $\mathbf{P}_L = \mathbf{P}_L^{\mathrm{c}}$) and simultaneously maximizing the SE (i.e., it has the same spectral efficiency as in (14)). Note that (32) can be re-written as in (19)

$$\begin{aligned}\widehat{\mathbf{s}}_l &= \left[\left(\mathbf{I} - \mathbf{T}_l\widehat{\mathbf{H}}_l\right)\ \mathbf{T}_l\right]\begin{bmatrix}\widehat{\mathbf{s}}_{(l-1)}\\ \mathbf{y}_l\end{bmatrix}\\ &= \left[(\overline{\mathbf{V}}_{(l-1)} - \mathbf{T}_l\widehat{\mathbf{H}}_l\overline{\mathbf{V}}_{(l-1)})\ \ \mathbf{T}_l\right]\mathbf{z}_l\\ &= \overline{\mathbf{V}}_l\mathbf{z}_l,\end{aligned} \tag{35}$$

where

$$\overline{\mathbf{V}}_l = \begin{cases}\left[(\overline{\mathbf{V}}_{(l-1)} - \mathbf{T}_l\widehat{\mathbf{H}}_l\overline{\mathbf{V}}_{(l-1)})\ \ \mathbf{T}_l\right], & l > 1\\ \mathbf{T}_1, & l = 1.\end{cases} \tag{36}$$

Equation (35) is important since it establishes the relationship between the proposed sequential processing estimate and the centralized processing estimate in (11). We prove the proposed algorithm is optimal with the help of the following theorem.

Theorem 1 *In the sequential processing with Algorithm 1, the estimate obtained at AP L is equivalent to that obtained by centralized processing with LMMSE receiver i.e.,*

$$\widehat{\mathbf{s}}_L = \widehat{\mathbf{s}}_L^{\mathrm{c}}, \tag{37}$$

Proof: The proof is given in Appendix A. ■

In a nutshell, Theorem 1 shows that it is possible to decentralize the LMMSE receiver to a sequential implementation using the Algorithm 1 and we will call this algorithm optimal sequential linear processing (OSLP) since it is optimal in the generic class of sequential linear processing in the sense of the SE. Since the signal's estimate at the CPU with the proposed OSLP algorithm is exactly equal to that of a centralized LMMSE receiver, any performance metric (eg., the MSE or the BER) computed based on the receiver's output will also be identical. We introduce the following notations to use in the corollaries that follow:

$$\mathbf{A}_l^o = \mathbf{I}_K - \mathbf{T}_l\widehat{\mathbf{H}}_l \tag{38}$$

$$\mathbf{B}_l^o = \mathbf{T}_l, \tag{39}$$

be the OSLP receiver matrices at AP L from (28) and (29). The important consequences of Theorem 1 are presented as corollaries below:

Corollary 1.1 *Comparing* (35) *and* (19) *and from Theorem 1, the following relation holds:*

$$\mathrm{SE}_k\left(\{\mathbf{A}_l, \mathbf{B}_l\}\right) \leq \mathrm{SE}_k(\mathbf{V}_L^{\mathrm{c}}),\ l = 1, \ldots, L \tag{40}$$

where $\mathrm{SE}_k(\cdot)$ *is the achievable SE of k UE,* $\{\mathbf{A}_l, \mathbf{B}_l\}$ *are receiver matrices for any generic sequential linear processing given in* (18) *and* $\mathbf{V}_L^{\mathrm{c}}$ *is the centralized LMMSE receiver given in* (12). *Equality is achieved with the proposed OSLP algorithm i.e., when* $\{\mathbf{A}_l, \mathbf{B}_l\} = \{\mathbf{A}_l^o, \mathbf{B}_l^o\}$. *A rigorous lower bound on the capacity using the OSLP can be obtained in a closed form by plugging* $\overline{\mathbf{B}}_L = \overline{\mathbf{V}}_L$ *in* (20) *and using Proposition 1. The maximum achievable SE of UE k is given by*

$$\mathrm{SE}_k = \left(1 - \frac{\tau_p}{\tau_c}\right)\mathbb{E}\left\{\log_2(1 + \Gamma_k^{max})\right\}, \tag{41}$$

where Γ_k^{max} *is the maximum instantaneous effective SINR given by*

$$\Gamma_k^{max} = p_k\widehat{\mathbf{h}}_k^H\left(\sum_{i=1, i\neq k}^{K} p_i\widehat{\mathbf{h}}_i\widehat{\mathbf{h}}_i^H + \mathbf{K}_L\right)^{-1}\widehat{\mathbf{h}}_k. \tag{42}$$

Corollary 1.2 *Comparing* (35) *and* (19) *and from Theorem 1, the following relation holds:*

$$e_k^{'}(\mathbf{V}_L^{\mathrm{c}}) \leq e_k^{'}\left(\{\mathbf{A}_l, \mathbf{B}_l\}\right),\ l = 1, \ldots, L \tag{43}$$

where $e_k^{'}(\cdot)$ *is the MSE of the kth UE at the CPU. Equality is achieved with the proposed OSLP algorithm i.e., when* $\{\mathbf{A}_l, \mathbf{B}_l\} = \{\mathbf{A}_l^o, \mathbf{B}_l^o\}$. *The proposed OLSP algorithm achieves the minimum MSE* e_k^{min} *given the side information for UE k at the CPU which is computed in closed form by taking the kth diagonal entry of the error covariance matrix* $\mathbf{P}_L^{\mathrm{c}}$ *(since* $\mathbf{P}_L = \mathbf{P}_L^{\mathrm{c}}$ *) in* (16) *and is given by*

$$e_k^{min} = p_k - p_k^2\widehat{\mathbf{h}}_k^H\mathbf{\Lambda}_L^{-1}\widehat{\mathbf{h}}_k. \tag{44}$$

Corollary 1.3 *The achievable SINR* Γ_k *of UE k increases monotonically with an increase in the number of APs L. On other hand, the MSE at the CPU,* e_k *being inversely related to* Γ_k *as*

$$e_k^{min} = \frac{p_k}{1 + \Gamma_k^{max}}, \tag{45}$$

decreases monotonically with the increase in L.

Proof: For an optimal centralized scheme, with an increase in the number of APs, say AP L to AP $(L+1)$, the available information at the CPU increases from $\{\mathbf{y}_1, \cdots, \mathbf{y}_L\}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3100619, IEEE Transactions on Communications

to $\{\mathbf{y}_1, \cdots, \mathbf{y}_{L+1}\}$. As there is no loss in the original information i.e., $\{\mathbf{y}_1, \cdots, \mathbf{y}_L\}$, it implies that the SINR (or the MSE) also monotonically increases (or decreases). From Theorem 1, the same result follows for the proposed OSLP algorithm with an increase in the number of APs. ■

Corollary 1.4 *The performance of the OSLP algorithm is invariant to the order in which the sequential processing is implemented i.e. the ordering of the APs.*

Proof: In a centralized processing with LMMSE receiver matrix, the estimate of the signal is obtained by utilizing the complete information received from all the APs in a cell-free setup. Hence, the performance will have no effect if the order in which the signals are received at the CPU is changed. Since the OSLP performance is equal to the centralized LMMSE processing from Theorem 1, the required result follows. ■

It is worth noting that the Corollary 1.4 is not true in general for any generic sequential linear processing in (18). For instance, it does not hold for the algorithms in [1] and [19].

There are two important practical benefits of the OSLP algorithm. The first benefit is that it makes use of local processing capabilities at the APs instead of requiring a CPU with a fast processor. The second benefit is that the achievable SE (or the MSE at the CPU) monotonically increases (or MSE decreases) with an increase in the number of APs while maintaining the same fronthaul capacity requirements in each link between the APs. We provide mathematical update equations for the SE and the MSE in the following proposition:

Proposition 3 *Let e_{kl} and Γ_{kl} denote the MSE and SINR of UE k achieved at AP l when using the OSLP algorithm for given channel estimates. The MSEs achieved at the adjacent APs can be computed using*

$$e_{kl} = e_{k(l-1)} - \alpha_{kl}, \tag{46}$$

where

$$\alpha_{kl} = \left[\mathbf{T}_l \widehat{\mathbf{H}}_l \mathbf{P}_{(l-1)}\right]_{kk}. \tag{47}$$

Note that the matrix $\mathbf{T}_l \widehat{\mathbf{H}}_l \mathbf{P}_{(l-1)}$ is non-negative definite and hence $\alpha_{kl} \geq 0$. Using (45)*, the update equation for the achievable SINR, Γ_{kl} of UE k at AP l is*

$$\Gamma_{kl} = \Gamma_{k(l-1)} + \gamma_{kl}, \tag{48}$$

where

$$\gamma_{kl} = \frac{\alpha_{kl}\left(\Gamma_{k(l-1)} + 1\right)^2}{p_k - \alpha_{kl}\left(\Gamma_{k(l-1)} + 1\right)}. \tag{49}$$

Finally, using the logarithmic equality, $\log_2(a+b) = \log_2(a) + \log_2(1 + \frac{b}{a})$ *and letting $a = 1 + \Gamma_{k(l-1)}, b = \gamma_{kl}$, we obtain the update equation of SE of UE k at AP l as*

$$\mathrm{SE}_{kl} = \mathrm{SE}_{k(l-1)} + \zeta_{kl} \tag{50}$$

where $\zeta_{kl} = \mathbb{E}\left\{\log_2(1 + \frac{\gamma_{kl}}{1+\Gamma_{k(l-1)}})\right\}$.

Algorithm 2 Sequential N-LMMSE Processing from [1]

```
1. Compute local LMMSE receiver, $\mathbf{V}_1 = [\mathbf{v}_{11}, \cdots, \mathbf{v}_{1K}]^H$
   given $\widehat{\mathbf{H}}_1$ and $\boldsymbol{\Sigma}_1$
2. Compute $\widehat{\mathbf{s}}_1 = \mathbf{V}_1 \mathbf{y}_1$ and initialize: $\widehat{\mathbf{H}}^{'}_{k1} = \widehat{\mathbf{H}}_1$,
   $\boldsymbol{\Sigma}^{'}_{kl} = \boldsymbol{\Sigma}_1, \ \forall k \in \{1, \cdots, K\}$
4. **for** $l = 2 : L$
      **for** $k = 1 : K$
      (a) Compute LMMSE receiver $\mathbf{v}_{kl}$ for $k$th UE, given $\widehat{\mathbf{H}}^{'}_{kl}$ and
          $\boldsymbol{\Sigma}^{'}_{kl}$ where, $\widehat{\mathbf{H}}^{'}_{kl} = [\widehat{\mathbf{H}}^{'H}_{k(l-1)} \mathbf{v}_{k(l-1)}, \widehat{\mathbf{H}}^H_l]^H$ is
          augmented channel estimate and
          $\boldsymbol{\Sigma}^{'}_{kl} = \mathrm{diag}(\mathbf{v}^H_{k(l-1)} \boldsymbol{\Sigma}^{'}_{k(l-1)} \mathbf{v}_{k(l-1)}, \boldsymbol{\Sigma}_l)$ is augmented
          noise covariance matrix
      (b) Compute $\widehat{s}_{kl} = \mathbf{v}^H_{kl} [\widehat{s}^{\,*}_{kl}, \mathbf{y}^H_l]^H$
      **end**
   **end**
5. **Output**: $\widehat{\mathbf{s}}_L$
```

D. Normalized LMMSE based sequential processing

In the conference version of this paper [1], a sequential processing algorithm using normalized LMMSE (N-LMMSE) scheme was presented which is given as a pseudo-code in Algorithm 2. The main difference between the algorithm presented in Algorithm 2 and the OSLP algorithm is that the former computes the kth UE's estimate $\widehat{s}_{kl}$ using only $\widehat{s}_{k(l-1)}$, whereas OSLP uses all the signal estimates in $\widehat{\mathbf{s}}_{(l-1)}$. Hence, the computation of the signal estimate in the OSLP algorithm makes use of more available information than Algorithm 2, thus the OSLP algorithm always performs superior or equal to Algorithm 2 in the sense of minimum MSE and also maximum SE.

For Algorithm 2, besides being suboptimal to the OSLP algorithm, the fronthaul signaling in each link between the APs is actually higher. This is quantitatively explained in Section IV.

IV. Fronthaul Signalling and Latency

In this section, we analyze two practical aspects crucial for radio stripes implementation, namely the fronthaul signaling capacity in each link connecting the APs and the latency in the network. We study these factors for the different algorithms described in Section III. We define the fronthaul signaling quantitatively as the total number of real symbols that AP L shares with the CPU in an arbitrary coherence block. One can observe that the amount of data required to be managed and processed by the CPU is directly proportional to the fronthaul signaling.

A. Centralized LMMSE Implementation

In the fully centralized processing scheme the signaling increases along the fronthaul since the signals from every AP must reach the CPU without being merged with the signals from other APs. Hence, we consider the fronthaul signaling to be the total number of real symbols that are being transmitted in all the links between the APs and the CPU, which is the capacity that is required between AP L and the CPU. We assume that the CPU has the information of channel spatial statistics i.e., $\mathbf{R}_{kl}$. Each AP forwards $\tau_c N$ complex symbols corresponding to the received pilot and data signals to the CPU. This amounts to $\tau_c NL$ complex symbols or equivalently $2\tau_c NL$ real symbols being transmitted from the APs to the CPU.

B. The OSLP Algorithm

In each link between the APs, the following amount of information is being shared:

(i) Each AP forwards the computed signal estimate $\widehat{\mathbf{s}}_l$ of the signal $\mathbf{s}$ to the successive AP i.e., from AP l to $(l+1)$, $l = 1, \cdots L-1$. This corresponds to $2K(\tau_c - \tau_p)$ real symbols.

(ii) The error covariance matrix $\mathbf{P}_k$, which corresponds to K^2 real symbols.

With the proposed OSLP algorithm, the data that AP L forwards to the CPU captures all the useful information content from all the other APs, however, the number of symbols shared by AP L to the CPU will remain the same and there is no redundancy in the data.

C. Algorithm 2 [1]

As described briefly in Section III-D, the amount of data that is being shared in each link between the APs is:

(i) Each AP forwards the computed signal estimate $\widehat{\mathbf{s}}_l$ of the signal $\mathbf{s}$ to the next AP, thus $2K(\tau_c - \tau_p)$ real symbols.

(ii) Effective channel estimates $\widehat{\mathbf{H}}'^{H}_{(l-1)}\mathbf{v}_{k(l-1)}$ $\forall k$ that is represented by $2K^2$ real symbols.

(iii) Effective channel matrix estimation statistics $\mathbf{v}^H_{k(l-1)}\boldsymbol{\Sigma}'_{(l-1)}\mathbf{v}_{k(l-1)}$ $\forall k$ which amounts to K real symbols.

The total fronthaul signaling required for Algorithm 2 is $2K^2 + K + 2K(\tau_c - \tau_p)$ real symbols. The fronthaul mentioned in [1] is $3K^2 + 2K(\tau_c - \tau_p)$ real symbols. The extra $K^2 - K$ fronthaul signaling is due to the way the channel estimation error statistics are shared between the APs as per algorithm described therein. The algorithm presented in Algorithm 2 is more efficient way of implementing the algorithm described in [1].

D. The RLS Algorithm [19]

One of the competing algorithms is RLS which is a recursive implementation of ZF algorithm. This algorithm is also a special case of (18) and hence can be implemented in a sequential cell-free network. We will compare our proposed algorithm with the RLS algorithm in the simulation section. The fronthaul analysis of the RLS algorithm is as follows:

(i) Each AP forwards the computed signal estimate $\widehat{\mathbf{s}}_l$ of the signal $\mathbf{s}$ to the successive AP i.e., from AP l to $l+1$, $l = 1, \cdots L-1$. This corresponds to $2K(\tau_c - \tau_p)$ real symbols.

(ii) A side information hermitian-matrix of size $K \times K$, which corresponds to K^2 real symbols.

Table I presents the fronthaul requirements for all the algorithms under two categories, one being the data estimates which are shared for every channel use and the statistical parameters (conditioned on available side information) or channel estimates which are shared once in every coherence block. The data estimates shared are the same for all the sequential algorithms described above and they amount to $2K(\tau_c - \tau_p)$ real symbols. It can be observed that the fronthaul requirement for a centralized processing increases linearly with an increase in L. From this, we conclude that the proposed OSLP algorithm has a lower fronthaul requirement than the centralized processing implementation (for large L) and also over Algorithm 2. On the other hand, it has an equivalent fronthaul requirement as the RLS algorithm.

E. Latency

We define the latency as the total time required from the signal's reception at AP 1 to the computation of the signal's estimate at the CPU. The data signals transmitted per channel use would take L time blocks (each time block corresponds to the processing time of each AP) to reach the CPU. However, all APs need not wait to process the next received signal i.e., in sequential processing, when AP $l \in \{2, \cdots, L\}$ is computing the estimate of the received signal at any arbitrary time block t_n, then AP 1 to AP $(l-1)$ can process the next payload data received in time blocks t_{n+1} to t_{n+l-1} as depicted in Fig. 3. Thus, if t_u channel uses are allocated for transmission of data, there will be $\tau_u + L - 1$ rows in the Fig. 3 which amounts to delay of $t_u + L$ time blocks as opposed to $t_u L$ time blocks. The idea here is that the APs can process on next consecutive data as soon as they are available. Thus, the AP l, stores $(l-1)$ received signals over time and processes them once its corresponding processed signal is received from AP $(l-1)$ (practically, the number of APs in a single radio stripe may range from 50-100, and therefore, memory storage will not form a bottleneck). When $t_u \gg L$, which is practically the case, the delay is approximately t_u and the extra delay of L time blocks is small. Thus, for t_u channel uses the delay is approximately t_c time blocks and hence sequential processing practically have less effect on overall latency. The latency can be further reduced by an alternative OSLP algorithm as described below, which is a semi-distributed implementation and then we present the computation of latency quantitatively.

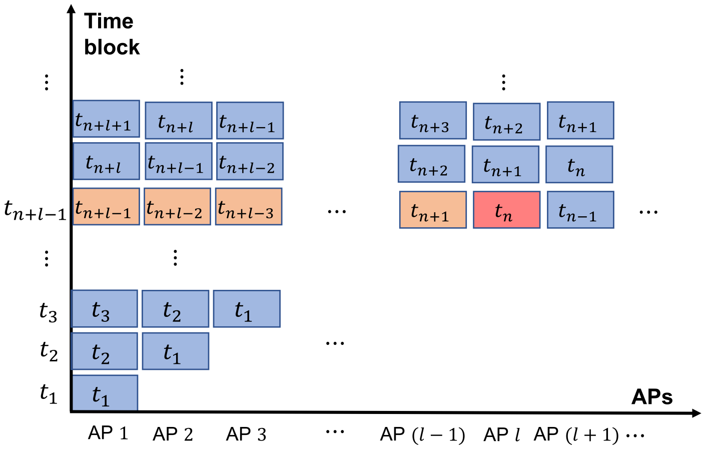

Time block
⋮
t_{n+l+1} t_{n+l} t_{n+l-1}
t_{n+3} t_{n+2} t_{n+1}
t_{n+l} t_{n+l-1} t_{n+l-2}
t_{n+2} t_{n+1} t_n
t_{n+l-1}
t_{n+l-1} t_{n+l-2} t_{n+l-3} …
t_{n+1} t_n t_{n-1} …
t_3 t_3 t_2 t_1
t_2 t_2 t_1
t_1 t_1
APs
AP 1
AP 2
AP 3
⋯
AP $(l-1)$
AP l
AP $(l+1)$ ⋯

Figure 3: Depiction of processing of data by each AP in each time block.

Fronthaul requirement in each coherence block		
Methods/Algorithms	Data Estimates	Statistical Parameters/Channel Estimates
Centralized Processing	$2\tau_c NL$	0
OSLP	$2K(\tau_c - \tau_p)$	K^2
Algorithm 2	$2K(\tau_c - \tau_p)$	$2K^2 + K$
S-MR	$2K(\tau_c - \tau_p)$	K
RLS [19]	$2K(\tau_c - \tau_p)$	K^2

Table I: Summary of fronthaul signaling for various algorithms

To describe the alternative OSLP algorithm (same as the OSLP but different implementation), recall that the estimate of the signal at the CPU with the centralized LMMSE receiver from (11) as

$$\widehat{\mathbf{s}}^{\text{c}}_L = \mathbf{Q}\widehat{\mathbf{G}}_L^H \left(\mathbf{K}_L + \widehat{\mathbf{G}}_L\mathbf{Q}\widehat{\mathbf{G}}_L^H\right)^{-1} \mathbf{z}_L. \tag{51}$$

This estimate can be written in an equivalent form as

$$\begin{aligned}\widehat{\mathbf{s}}^{\text{c}}_L &\overset{(a)}{=} \left(\mathbf{Q}^{-1} + \widehat{\mathbf{G}}_L^H\mathbf{K}_L^{-1}\widehat{\mathbf{G}}_L\right)^{-1}\widehat{\mathbf{G}}_L^H\mathbf{K}_L^{-1}\mathbf{z}_L \\ &= \left(\mathbf{Q}^{-1} + \sum_{l=1}^{L}\widehat{\mathbf{H}}_l^H\boldsymbol{\Sigma}_l^{-1}\widehat{\mathbf{H}}_l\right)^{-1}\left(\sum_{l=1}^{L}\widehat{\mathbf{H}}_l^H\boldsymbol{\Sigma}_l^{-1}\mathbf{y}_l\right),\end{aligned} \tag{52}$$

where (a) is an alternative form of the LMMSE receiver matrix [21]. This approach is similar to the partial decentralization method presented in [10] but the authors therein analyzed it for the case of perfect CSI. It is worth noting that the estimate in (51) (or equivalently the OSLP estimate) is also the maximum a posteriori (MAP) estimate if the information signal vector is Gaussian distributed. It is also interesting to note that this algorithm can be extended to a tree network [18].

In this alternative form of the OSLP algorithm, after the APs have estimated the channels, they simultaneously compute their quadratic forms $\widehat{\mathbf{H}}_l^H\boldsymbol{\Sigma}_l^{-1}\widehat{\mathbf{H}}_l$ and then AP $l \in \{1, \cdots, L\}$ forwards the following cumulative sum of quadratic form to AP $(l+1)$

$$\mathbf{M}_l = \mathbf{M}_{(l-1)} + \widehat{\mathbf{H}}_l^H\boldsymbol{\Sigma}_l^{-1}\widehat{\mathbf{H}}_l, \tag{53}$$

where $\mathbf{M}_0$ is a $K \times K$ matrix with only zeros. When the CPU receives $\mathbf{M}_L$, it computes the inverse matrix in (51) (computed once in every coherence block). Next, in each of $\tau_c - \tau_p$ channel uses, all APs simultaneously compute the local weighted MR estimate $\widehat{\mathbf{H}}_l^H\boldsymbol{\Sigma}_l^{-1}\mathbf{y}_l$ using their corresponding received signal. Then, AP $l \in \{1, \cdots, L\}$ forwards the following cumulative sum of weighted MR estimated signals to the CPU:

$$\widetilde{\mathbf{s}}_l = \widetilde{\mathbf{s}}_{(l-1)} + \widehat{\mathbf{H}}_l^H\boldsymbol{\Sigma}_l^{-1}\mathbf{y}_l, \tag{54}$$

where $\widetilde{\mathbf{s}}_l = [\widetilde{s}_{1l}, \cdots, \widetilde{s}_{Kl}]^T$ is the local weighted MR estimate of the UEs payload with $\widetilde{s}_{kl}$ being the kth UE local MR estimate and $\widetilde{\mathbf{s}}_0 = \mathbf{0}$. Upon receiving $\widetilde{\mathbf{s}}_L$, the CPU computes the estimate $\widehat{\mathbf{s}}^c_L$ as

$$\widehat{\mathbf{s}}^c_L = \left(\mathbf{Q}^{-1} + \mathbf{M}_L\right)^{-1}\widetilde{\mathbf{s}}_L. \tag{55}$$

With the alternative OSLP algorithm, only (54) needs to be computed for every channel use and (53) is computed once in every coherence block, thus helping in reducing the overall latency as it involves only add and forwards mechanism for every channel use.

We will now compare the latency quantitatively with an increase in L in terms of the time complexity, $\mathcal{O}(\cdot)$, involved in computing the estimate in the proposed alternative OSLP algorithm and the centralized implementation. For time complexity computation, we consider only those terms which are computed in every channel use. For the proposed alternative OSLP algorithm in (54), the terms $\widehat{\mathbf{H}}_l^H\boldsymbol{\Sigma}_l^{-1}\mathbf{y}_l$, $\forall l$ can be computed in parallel, and then sequentially summing up of those $K \times 1$ terms constitutes the delay in the order of $\mathcal{O}(L)$. Hence, with an increase in L and fixed other parameters, the proposed OSLP algorithm's time complexity increases linearly. On the other hand, a centralized network can implement the LMMSE estimate according to (52) or using (11). With (52), the time complexity order for computing the summation term is approximately $\mathcal{O}(L/\eta_{\text{CPU}} + \log\eta_{\text{CPU}})$, where η_{CPU} is the number of parallel processors available at the CPU [23]. This is because, with η_{CPU} local processors, the local reduction of L tasks at each processor would take $\mathcal{O}(L/\eta_{\text{CPU}})$, and then computing reduction of η_{CPU} local results requires $\mathcal{O}(\log\eta_{\text{CPU}})$. If $\eta_{\text{CPU}} = L$ (e.g., tree-like architecture), then the time complexity increases logarithmically with L, and when $\eta_{\text{CPU}} = 1$ (e.g., daisy-chain architecture), it grows linearly. However, with a fixed number of parallel processors, the time complexity increases linearly with L but at a slower rate than the proposed sequential network. On the other hand, if the CPU utilizes the estimator in (11), then the order of time complexity is approximately $\mathcal{O}(KNL/\eta_{\text{CPU}})$ which depends on N besides K and L, which grows faster than (52). Hence, it is preferable for a centralized implementation to utilize (52). It is interesting to note that for the scalable tree network proposed in [18], the time complexity grows in the order of $\mathcal{O}(\log L)$. So from the discussion, we observe that the latency of the proposed sequential network increases linearly with an increase in L, forming a limiting factor in latency-critical applications. However, as presented in the introduction, a few of the attractive features of radio stripes, such as its ease of deployment and low fronthaul requirements (this has the implication that the amount of data to be processed at the CPU will remain unchanged with an increase in the number of APs) favors the sequential network in many practical applications. Therefore, there is a trade-off between the latency and ease of practical implementation with the proposed radio stripes network.

With the alternative OSLP algorithm, the CPU collects sufficient information from all the APs to decode the data at the CPU, and the performance would be equivalent to a centralized implementation with any performance metric, for instance, bit error rate (BER). All the results established earlier for the OSLP algorithm hold equally true for the alternative OSLP algorithm. The fronthaul requirement for the alternative OSLP is the same as that of the original OSLP algorithm, and they differ only in the way processing is done. In the former case, the final fusion of data is done at the CPU where a $K \times K$ matrix inversion is required to obtain the final estimate, while in the latter case, all the processing is done at APs where $N \times N$ matrix inversion is computed, and only symbol decoding is done at the CPU. But it should be noted that, when the APs are capable of implementing the original OSLP algorithm then it is advantageous over alternative OSLP algorithm mainly because the alternative OSLP algorithm depends on the number of UEs and with the increase in the number of UEs the computation of quadratic form (53) at the APs and especially inverting the $K \times K$ matrix in (55) at the CPU would be computationally expensive. The time complexity of the proposed OSLP algorithm grows as $\mathcal{O}(L)$ but at faster rate than alternative OSLP algorithm.

V. Numerical Results and Discussions

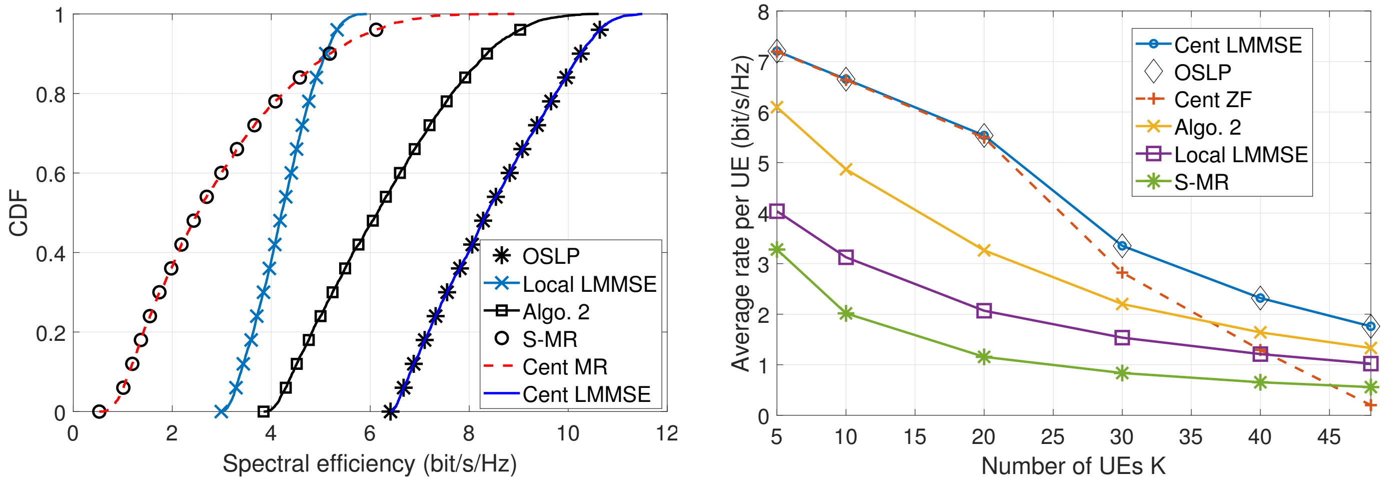
In this section, we evaluate the performance of the proposed OSLP algorithm through numerical results by considering a simulation setup in an area of 125 m × 125 m. We consider the achievable SE as the performance metrics. We assume that the APs are placed equidistant on a radio stripe of length 500 m which is wrapped around the square perimeter of the simulation area. The propagation model considered for analysis is 3GPP Urban Microcell model [24, Table B.1.2.1-1] with 2 GHz carrier frequency and the large-scale fading coefficient as

$$\beta_{kl} = -30.5 - 36.7\log_{10}\left(\frac{d_{kl}}{1\text{m}}\right), \tag{56}$$

where d_{kl} is the distance between AP l and UE k (this includes a vertical height difference of 5 m between the APs and the UEs). We assume further that the UEs are uniformly distributed within the concentric square of 100 m × 100 m. Each UE transmits with 50 mW power unless otherwise mentioned. The noise power σ^2 is -85 dBm with noise figure of 9 dB, the bandwidth is 100 MHz, the coherence block length $\tau_c = 2000$ channel uses, and the number of orthogonal pilot sequences $\tau_p = \min(K, 20)$. The pilot assignment is done as per the algorithm in [20] but without clustering. The total number of APs is $L = 24$ and each has $N = 4$ antennas unless otherwise stated. The spatial correlation is modeled using the local scattering model [25, Sec 2.6]. We consider a uniform linear array for each AP with half wavelength antennas spacing and the multipath components are Gaussian distributed in the angular domain with a 15 degree standard deviation around the nominal angle to the user. Note that the results presented do not include the alternative OSLP algorithm because it has the same performance as the original OSLP algorithm in Algorithm 1 and all the results presented equivalently hold for both implementations.

In Fig. 4a, we plot the cumulative distribution functions (CDFs) of the SE for users at random locations for the proposed OSLP algorithm and compare with other competing algorithms including centralized scheme and in Fig. 4b, the average rate per UE is plotted to understand how many UEs the proposed algorithm supports for fixed other specifications. The results demonstrate and validate the claim for the equivalence of the proposed OSLP algorithm with the centralized LMMSE implementation (labelled as "Cent LMMSE" in plots). Fig. 4a and 4b also demonstrate the performance of S-MR (sequential MR given in (25)). We compare the results with distributed LMMSE processing in [20], described as level 2 processing in [9] (labelled "Local LMMSE" in plots) which is a particular case of generic sequential processing with $\mathbf{A}_l = \mathbf{I}_K, \mathbf{B}_l = (1/L)\mathbf{Q}\widehat{\mathbf{H}}_l^H(\mathbf{\Sigma}_l + \widehat{\mathbf{H}}_l\mathbf{Q}\widehat{\mathbf{H}}_l^H)^{-1}$ and will be sub-optimal to the OSLP algorithm. We also compare these results with Algorithm 2 (labelled "Algo. 2"), and we observe that it has superior performance over MR and Local LMMSE. This is because Algorithm 2 not only makes use of prior knowledge of the channel and the noise statistics but also takes the advantage of APs cooperation in a sequential setup with side information from other APs which helps in suppressing the interference more efficiently than MR and Local LMMSE. From Fig. (4b), it can be observed that the proposed algorithm supports more UEs with a minimum quality of service (minimum rate) than other competing algorithms. Also, we observe that the performance of centralized ZF is very poor when the number of UEs is comparable to the total number of receiver antennas of all APs. This also reassures our motivation to decentralize LMMSE (optimal in all the regions of system specifications) over ZF.

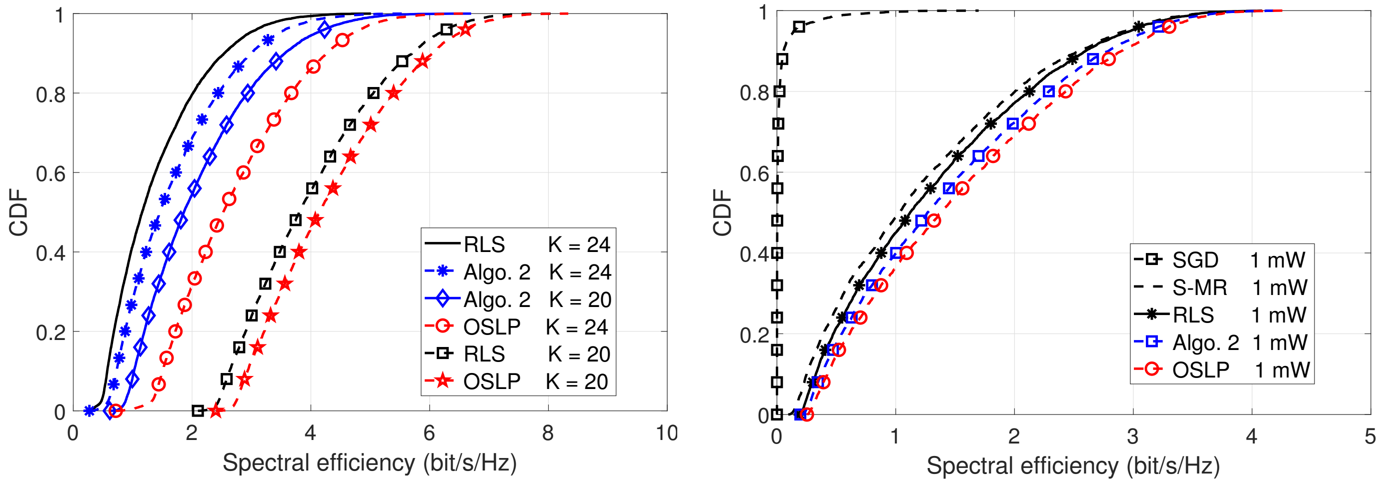
In Fig. 5a, the proposed OSLP algorithm is compared with the RLS [19] algorithm (discussed in the subsection of fronthaul analysis) with the CDF of the SE of the users. Since, [19] considered $N = 1$, we make the same assumption in Fig. 4 to achieve a fair comparison. The results have been plotted for the system model of this paper with imperfect CSI for $K = 20$ and $K = 24$. It is observed that as the number of UEs increases the performance gain of the OSLP algorithm over the RLS algorithm increases. For instance, with $K = 24$, the proposed OSLP algorithm gains 1.24 bit/sec/Hz as compared to the RLS algorithm with 50% probability. Interestingly, for $K = 24$, with Algorithm 2, the UEs gain 0.3 bit/s/Hz over the RLS algorithm with 50% probability. It has to be noted that for the case of $K = 24$, pilot contamination effect is taken into consideration. This shows that the RLS performance is poor when the system is heavily loaded and pilot contamination limited. Since, the performance of the centralized ZF is very poor (refer 4b) when the number of UEs is comparable to the total number of receiver antennas, it follows that the RLS will also under perform in such scenarios. In Fig. 5b, the CDF of the SE of the UEs is analyzed at low transmit power i.e., 1 mW for each UE. The number of UEs considered for the results in Fig. 5b is $K = 10$. Besides RLS, another recursive algorithm called stochastic gradient descent (SGD) [19] with step size 0.02 is considered. The SGD algorithm does not seem to adopt with imperfections in CSI and we



(a) CDF of the SE with $L = 24$ and $K = 10$.

(b) Average rate per UE with $L = 12$ and varying K.

Figure 4: Comparison of the proposed OSLP algorithm with centralized LMMSE and other competing algorithms.



(a) Varying K and fixed $L = 24, \ N = 1$

(b) At low SNR (with $p_k = 1\text{mW} \ \forall k$)

Figure 5: Comparison of the proposed OSLP algorithm with other competing algorithms using the CDF of the SE with fixed $L = 24, \ N = 1$.

observe it has inferior performance. With the proposed OSLP algorithm, the SE of the UEs with 50% probability have 0.24 bits/sec/Hz gain over the RLS algorithm. While at high SNR, the RLS method has comparable performance with OSLP (not shown in Fig. 5b). Since practical systems operate mostly at low SNR regimes, these results illustrate that the RLS and the SGD algorithms have poor performance as compared to the proposed OSLP algorithm.The RLS algorithm [19] is a recursive implementation of centralized ZF algorithm and does not take the side information of the channel and noise statistics into consideration. Hence, it will have inferior performance when compared to the proposed OSLP algorithm in general and especially at low SNR. Next we analyze the performance of the proposed OSLP algorithm in terms of normalized MSE of the signal estimates versus the number of APs. We define the normalized MSE as

$$\frac{\mathbb{E}\{\|\mathbf{s} - \widehat{\mathbf{s}}\|^2\}}{\mathbb{E}\{\|\mathbf{s}\|^2\}}. \tag{57}$$

In Fig. 6, we analyze the MSE performance of the algorithms with imperfect CSI with $K = 10$ UEs. We observe that Algorithm 2 has better performance over RLS when the number of APs is less than the total number of receiver antennas of all APs i.e., NL but eventually under-performs with further increase in L. In our simulation, we observed that the SGD does not scale with the imperfections in CSI, i.e., the normalized MSE is very high and hence is excluded in Fig. 6. From these results, it is clear that the OSLP algorithm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3100619, IEEE Transactions on Communications

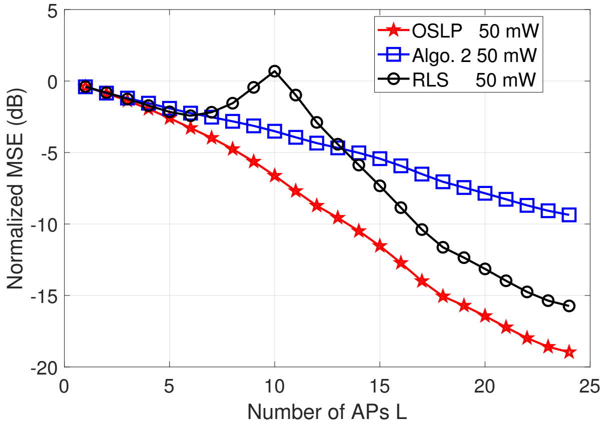


Figure 6: Comparison of MSE performance of the proposed OSLP algorithm when $K = 10$ and $N = 1$.

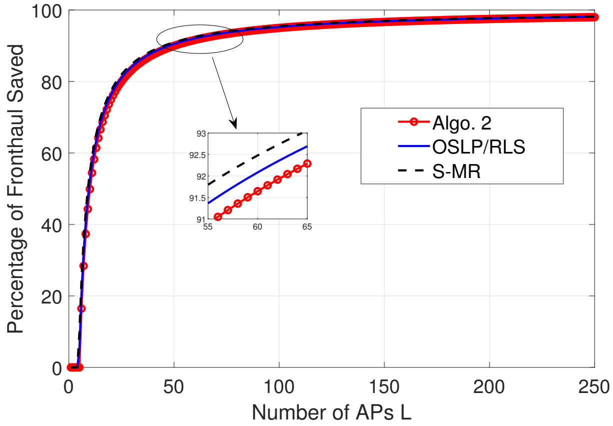


Figure 7: Percentage of fronthaul saved for the proposed OSLP algorithm over centralized implementation.

is an ideal choice for a radio stripe network. These results demonstrate that the prior knowledge about channel and noise plays a key role in its consistent performance even in extreme cases (e.g., with the numbers of UEs being comparable to the number of APs and operating in the low SNR regime).

Finally, in Fig. 7, we illustrate the percentage of fronthaul signaling saved i.e., the number of real symbols saved compared with the centralized processing with a fixed number of UEs and increasing number of APs (since in cell-free mMIMO $L \gg K$). The results in Fig. 7 are generated by using Table I for $K = 20$. As an example with $L = 60$, for the proposed OSLP algorithm, the fronthaul signaling reduces by 90% as compared to that of the centralized LMMSE algorithm. The RLS algorithm has the same fronthaul signaling requirement as for OSLP. This analysis concludes that the proposed OSLP algorithm has lower fronthaul requirements than centralized implementation besides being optimal.

VI. Conclusion

This paper proposes a sequential uplink processing framework that is an optimal choice for any sequential implementation of cell-free mMIMO networks, for instance, radio stripes. We have shown analytically that the proposed OSLP algorithm is optimal in both the maximum SE and the minimum MSE sense. The proposed OSLP algorithm forms the benchmark to analyze the loss of performance of other competing sequential linear algorithms in the sense of SE. We have provided closed-form expressions for the achievable maximum SE and the minimum MSE for the proposed OSLP algorithm, and elaborated on the implications. We also briefly presented an alternative implementation of the same algorithm that is semi-distributed. The main benefit of the OSLP algorithm is that it achieves the same performance as the optimal centralized scheme, but requires much lower fronthaul signaling and makes use of the distributed processors located at the APs.

Appendix A: Proof of Theorem 1

We will prove Theorem 1 using mathematical induction. To establish that the estimate at AP L using the OSLP algorithm and the estimate obtained using centralized LMMSE receiver are the same, we make use of the alternative expression for the signal estimate obtained by the OSLP algorithm given in (35). From (35) and (37), it is sufficient to show that the receive filters are equal: $\overline{\mathbf{V}}_L = \mathbf{V}_L^c$. Besides showing that both receivers are the same, we also show simultaneously that the MSE matrix $\mathbf{P}_L$ in the OSLP algorithm is equal to the MSE of the centralized scheme i.e., $\mathbf{P}_L = \mathbf{P}_L^c$. It then follows that the same SE and MSE are achieved.

Recall that the LMMSE receiver for centralized scheme with L APs, $\mathbf{V}_L^c$, and its corresponding error covariance matrix, $\mathbf{P}_L^c$ for (10) are given [21], respectively, as

$$\mathbf{V}_L^{\mathrm{c}} = \mathbf{Q}\widehat{\mathbf{G}}_L^H \left(\mathbf{K}_L + \widehat{\mathbf{G}}_L \mathbf{Q} \widehat{\mathbf{G}}_L^H\right)^{-1} \tag{58}$$

$$\mathbf{P}_L^{\mathrm{c}} = \mathbf{Q} - \mathbf{V}_L^{\mathrm{c}} \widehat{\mathbf{G}}_l \mathbf{Q}. \tag{59}$$

Now we prove the claim with mathematical induction using two cases: Case (i): $L = 1$. In the case where there is a single AP i.e., only AP 1, then the LMMSE receive matrix at the CPU for centralized scheme is

$$\begin{aligned} \mathbf{V}_1^{\mathrm{c}} &= \mathbf{Q}\widehat{\mathbf{G}}_1^H \left(\mathbf{K}_1 + \widehat{\mathbf{G}}_1 \mathbf{Q} \widehat{\mathbf{G}}_1^H\right)^{-1} \\ &= \mathbf{P}_0 \widehat{\mathbf{H}}_1^H \left(\mathbf{\Sigma}_1 + \widehat{\mathbf{H}}_1 \mathbf{P}_0 \widehat{\mathbf{H}}_1^H\right)^{-1} \\ &= \overline{\mathbf{V}}_1, \end{aligned} \tag{60}$$

where the last equality follows from (36). Hence, for $L = 1$, we have proved that the receiver matrices of the two algorithms are the same. Next, using (60) combined with the MSE expression in (59), the equivalence of the MSE matrices for both algorithms can be established as follows:

$$\begin{aligned} \mathbf{P}_1^c &= \mathbf{Q} - \overline{\mathbf{V}}_1 \widehat{\mathbf{G}}_1 \mathbf{Q} \\ &\overset{(a)}{=} \left(\mathbf{I} - \mathbf{T}_1 \widehat{\mathbf{H}}_1\right) \mathbf{P}_0 \\ &= \mathbf{P}_1. \end{aligned} \tag{61}$$

In (61), (a) follows from (60) and (36). Hence, the estimate $\widehat{\mathbf{s}}_1$ obtained using OSLP algorithm and centralized scheme are equivalent for the case of $L = 1$.

Case (ii): Assume, that this equivalence of receiver matrices and it's MSE matrices holds for the cases $L \in \{1, \cdots, l\}$ i.e.,

$$\begin{aligned} \mathbf{V}_l^{\mathrm{c}} &= \overline{\mathbf{V}}_l \\ &\overset{(a)}{=} \mathbf{Q}\widehat{\mathbf{G}}_l^H \left(\mathbf{K}_l + \widehat{\mathbf{G}}_l\mathbf{Q}\widehat{\mathbf{G}}_l^H\right)^{-1} \\ &\overset{(b)}{=} \left[\overline{\mathbf{V}}_{(l-1)} - \mathbf{T}_l\widehat{\mathbf{H}}_l\overline{\mathbf{V}}_{(l-1)} \quad \mathbf{T}_l\right] \end{aligned} \tag{62}$$

and also the MSE matrices

$$\begin{aligned} \mathbf{P}_l^c &= \mathbf{P}_l \\ &\overset{(c)}{=} \mathbf{Q} - \mathbf{V}_l^c\widehat{\mathbf{G}}_l\mathbf{Q} \\ &\overset{(d)}{=} \mathbf{Q} - \overline{\mathbf{V}}_l\widehat{\mathbf{G}}_l\mathbf{Q}. \end{aligned} \tag{63}$$

In (62), (a) follows from (58) and (b) is from (36). Similarly, (c) in (63) is due to (59) and (d) follows from the assumption in (62).

We will use these expressions to show that the equivalence holds for the case of $L = (l+1)$ which would complete the proof. For simplicity we let the inverse term in the LMMSE receiver matrix in (58) with $L = (l+1)$ APs

$$\begin{aligned} \left(\begin{bmatrix}\mathbf{K}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{(l+1)}\end{bmatrix} + \begin{bmatrix}\widehat{\mathbf{G}}_l \\ \widehat{\mathbf{H}}_{(l+1)}\end{bmatrix}\mathbf{Q}\begin{bmatrix}\widehat{\mathbf{G}}_l \\ \widehat{\mathbf{H}}_{(l+1)}\end{bmatrix}^H\right)^{-1} &= \begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D}\end{bmatrix}^{-1} \\ &= \begin{bmatrix}\bar{\mathbf{A}} & \bar{\mathbf{B}} \\ \bar{\mathbf{C}} & \bar{\mathbf{D}}\end{bmatrix}. \end{aligned} \tag{64}$$

The LMMSE receiver matrix is given by the

$$\begin{aligned} \mathbf{V}_{(l+1)}^{\mathrm{c}} &= \mathbf{Q}\begin{bmatrix}\widehat{\mathbf{G}}_l \\ \widehat{\mathbf{H}}_{(l+1)}\end{bmatrix}^H \begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D}\end{bmatrix}^{-1} \\ &= \left[\mathbf{Q}\widehat{\mathbf{G}}_l^H\bar{\mathbf{A}} + \mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{C}} \quad \mathbf{Q}\widehat{\mathbf{G}}_l^H\bar{\mathbf{B}} + \mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}\right] \\ &= \left[\mathbf{F}_1 \quad \mathbf{F}_2\right], \end{aligned} \tag{65}$$

where the new variables are defined as

$$\begin{aligned} \mathbf{A} &= \mathbf{K}_l + \widehat{\mathbf{G}}_l\mathbf{Q}\widehat{\mathbf{G}}_l^H, \\ \mathbf{B} &= \widehat{\mathbf{G}}_l\mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H, \\ \mathbf{C} &= \widehat{\mathbf{H}}_{(l+1)}\mathbf{Q}\widehat{\mathbf{G}}_l^H, \\ \mathbf{D} &= \mathbf{\Sigma}_{(l+1)} + \widehat{\mathbf{H}}_{(l+1)}\mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H \end{aligned} \tag{66}$$

and

$$\begin{aligned} \mathbf{F}_1 &= \mathbf{Q}\widehat{\mathbf{G}}_l^H\bar{\mathbf{A}} + \mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{C}}, \\ \mathbf{F}_2 &= \mathbf{Q}\widehat{\mathbf{G}}_l^H\bar{\mathbf{B}} + \mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}. \end{aligned} \tag{67}$$

To find the matrix inverse in (64), we utilize the standard block matrix inversion identity [26] given in (68) (bottom of this page). First, we compute $\bar{\mathbf{D}}$ because it simplifies other expressions:

$$\begin{aligned} \bar{\mathbf{D}}^{-1} &= \left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right) \\ &= \mathbf{\Sigma}_{(l+1)} + \widehat{\mathbf{H}}_{(l+1)}\mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H - \widehat{\mathbf{H}}_{(l+1)}\mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1}\widehat{\mathbf{G}}_l\mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H \\ &= \widehat{\mathbf{H}}_{(l+1)}\left(\mathbf{Q} - \mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1}\widehat{\mathbf{G}}_l\mathbf{Q}\right)\widehat{\mathbf{H}}_{(l+1)}^H + \mathbf{\Sigma}_{(l+1)} \\ &\overset{(a)}{=} \widehat{\mathbf{H}}_{(l+1)}\left(\mathbf{Q} - \overline{\mathbf{V}}_l\widehat{\mathbf{G}}_l\mathbf{Q}\right)\widehat{\mathbf{H}}_{(l+1)}^H + \mathbf{\Sigma}_{(l+1)} \\ &\overset{(b)}{=} \widehat{\mathbf{H}}_{(l+1)}\mathbf{P}_l\widehat{\mathbf{H}}_{(l+1)}^H + \mathbf{\Sigma}_{(l+1)}. \end{aligned} \tag{69}$$

In (69), (a) and (b) follows from the assumptions in (62) and (63), respectively. Next, (69) can be utilized to simplify the variables $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}$, and $\bar{\mathbf{D}}$ as follows:

$$\begin{aligned} \bar{\mathbf{D}} &= (\widehat{\mathbf{H}}_{(l+1)}\mathbf{P}_l\widehat{\mathbf{H}}_{(l+1)}^H + \mathbf{\Sigma}_{(l+1)})^{-1}, \\ \bar{\mathbf{A}} &= \mathbf{A}^{-1} + \mathbf{A}^{-1}\widehat{\mathbf{G}}_l\mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}^{-1}\widehat{\mathbf{H}}_{(l+1)}\mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1}, \\ \bar{\mathbf{B}} &= -\mathbf{A}^{-1}\widehat{\mathbf{G}}_l\mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}^{-1}, \\ \bar{\mathbf{C}} &= -\bar{\mathbf{D}}^{-1}\widehat{\mathbf{H}}_{(l+1)}\mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1}. \end{aligned} \tag{70}$$

In (70), the term $\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)$ in all the variables is substituted by simplified expression from (69). Now that we have obtained all the required variables to compute the inverse term in (65), we proceed to first simplify $\mathbf{F}_1$ as

$$\begin{aligned} \mathbf{F}_1 &= \mathbf{Q}\widehat{\mathbf{G}}_l^H\bar{\mathbf{A}} + \mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{C}} \\ &= \mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1}\widehat{\mathbf{G}}_l\mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}^{-1}\widehat{\mathbf{H}}_{(l+1)}\mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1} \\ &\quad + \mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1} + \mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{C}} \\ &= \overline{\mathbf{V}}_l - \left(\mathbf{Q} - \overline{\mathbf{V}}_l\widehat{\mathbf{G}}_l\mathbf{Q}\right)\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}^{-1}\widehat{\mathbf{H}}_{(l+1)}^H\overline{\mathbf{V}}_l \\ &= \overline{\mathbf{V}}_l - \mathbf{P}_l\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}^{-1}\widehat{\mathbf{H}}_{(l+1)}^H\overline{\mathbf{V}}_l \\ &\overset{(a)}{=} \overline{\mathbf{V}}_l - \mathbf{T}_{(l+1)}\widehat{\mathbf{H}}_{(l+1)}\overline{\mathbf{V}}_l, \end{aligned} \tag{71}$$

where (a) follows from (30) and (69). Next, we simplify the remaining term $\mathbf{F}_2$ as

$$\begin{aligned} \mathbf{F}_2 &= \mathbf{Q}\widehat{\mathbf{G}}_l^H\bar{\mathbf{B}} + \mathbf{Q}\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}} \\ &= \left(\mathbf{Q} - \mathbf{Q}\widehat{\mathbf{G}}_l^H\mathbf{A}^{-1}\widehat{\mathbf{G}}_l\mathbf{Q}\right)\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}^{-1} \\ &= \mathbf{P}_l\widehat{\mathbf{H}}_{(l+1)}^H\bar{\mathbf{D}}^{-1} \\ &= \mathbf{T}_{(l+1)}. \end{aligned} \tag{72}$$

Therefore, (65) using (36) becomes

$$\begin{aligned} \mathbf{V}_{(l+1)}^{\mathrm{c}} &= \left[\overline{\mathbf{V}}_l - \mathbf{T}_{(l+1)}\widehat{\mathbf{H}}_{(l+1)}\overline{\mathbf{V}}_l \quad \mathbf{T}_{(l+1)}\right] \\ &= \overline{\mathbf{V}}_{(l+1)} \end{aligned} \tag{73}$$

$$\begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D}\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1} \\ -\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{A}^{-1} & \left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\end{bmatrix} \tag{68}$$

which is the desired result for the equivalence of the receiver matrices. Moreover, the error covariance equivalence is established as

$$\begin{aligned}\mathbf{P}^{c}_{(l+1)} &= \mathbf{Q} - \mathbf{V}^{c}_{(l+1)} \begin{bmatrix} \widehat{\mathbf{G}}_{l} \\ \widehat{\mathbf{H}}_{(l+1)} \end{bmatrix} \mathbf{Q} \\ &\overset{(a)}{=} \mathbf{Q} - \overline{\mathbf{V}}_{l}\widehat{\mathbf{G}}_{l}\mathbf{Q} + \mathbf{T}_{(l+1)}\widehat{\mathbf{H}}_{(l+1)}\overline{\mathbf{V}}_{l}\widehat{\mathbf{G}}_{l}\mathbf{Q} - \mathbf{T}_{(l+1)}\widehat{\mathbf{H}}_{(l+1)}\mathbf{Q} \\ &\overset{(b)}{=} \mathbf{P}_{l} - \mathbf{T}_{(l+1)}\widehat{\mathbf{H}}_{(l+1)}\mathbf{P}_{l} \\ &= \left(\mathbf{I} - \mathbf{T}_{(l+1)}\widehat{\mathbf{H}}_{(l+1)}\right)\mathbf{P}_{l} \\ &= \mathbf{P}_{(l+1)}. \end{aligned} \tag{74}$$

In (74), (a) follows from (73) and (b) is due to (63). Thus, the (62) and (63) holds for any general L. This concludes the proof.

References

[1] Z. H. Shaik, E. Björnson, and E. G. Larsson, "Cell-free massive MIMO with radio stripes and sequential uplink processing," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[2] H. Yang and T. L. Marzetta, "Total energy efficiency of cellular large scale antenna system multiple access mobile networks," in *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2013, pp. 27–32.

[3] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 916–929, 2014.

[4] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.

[5] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The new 5G radio access technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, 2017.

[6] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality—what is next?: Five promising research directions for antenna arrays," *Digital Signal Processing*, vol. 94, pp. 3–20, 2019.

[7] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, 2017.

[8] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99 878–99 888, 2019.

[9] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, 2020.

[10] C. Jeon, K. Li, J. R. Cavallaro, and C. Studer, "Decentralized equalization with feedforward architectures for massive MU-MIMO," *IEEE Transactions on Signal Processing*, vol. 67, no. 17, pp. 4418–4432, 2019.

[11] A. Shirazinia, S. Dey, D. Ciuonzo, and P. Salvo Rossi, "Massive MIMO for decentralized estimation of a correlated source," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2499–2512, 2016.

[12] K. Li, R. R. Sharan, Y. Chen, T. Goldstein, J. R. Cavallaro, and C. Studer, "Decentralized baseband processing for massive MU-MIMO systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 491–507, 2017.

[13] A. Burr, M. Bashar, and D. Maryopi, "Cooperative access networks: Optimum fronthaul quantization in distributed massive MIMO and cloud RAN - invited paper," in *IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.

[14] M. Sadeghi, C. Yuen, and Y. H. Chew, "Sum rate maximization for uplink distributed massive MIMO systems with limited backhaul capacity," in *IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 308–313.

[15] K. Li, J. McNaney, C. Tarver, O. Castañeda, C. Jeon, J. R. Cavallaro, and C. Studer, "Design trade-offs for decentralized baseband processing in massive MU-MIMO systems," in *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 906–912.

[16] S. Bassoy, H. Farooq, M. A. Imran, and A. Imran, "Coordinated multipoint clustering schemes: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 743–764, 2017.

[17] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 197, 2019.

[18] E. Bertilsson, O. Gustafsson, and E. G. Larsson, "A scalable architecture for massive MIMO base stations using distributed processing," in *50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 864–868.

[19] J. Rodríguez Sánchez, F. Rusek, O. Edfors, M. Sarajlić, and L. Liu, "Decentralized massive MIMO processing exploring daisy-chain architecture and recursive algorithms," *IEEE Transactions on Signal Processing*, vol. 68, pp. 687–700, 2020.

[20] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247–4261, 2020.

[21] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. USA: Prentice-Hall, Inc., 1993.

[22] E. Nayebi, A. Ashikhmin, T. L. Marzetta, and B. D. Rao, "Performance of cell-free massive MIMO systems with MMSE and LSFD receivers," in *50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 203–207.

[23] C. P. Kruskal, L. Rudolph, and M. Snir, "Techniques for parallel manipulation of sparse matrices," *Theoretical Computer Science*, vol. 64, no. 2, pp. 135–157, 1989.

[24] *Further advancements for E-UTRA physical layer aspects (Release 9)*. 3GPP TS 36.814, Mar. 2010.

[25] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.

[26] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. USA: Cambridge University Press, 2012.