Distributed Precoding by In-Network Processing for Small Cell Networks

Guang Xu†, Henning Paul†, Thorsten Schier‡, Patrick Svedman‡ and Armin Dekorsy†
† Department of Communications Engineering, University of Bremen, Bremen, Germany
‡ ZTE Wistron Telecom AB, Stockholm, Sweden
Email: {xu,paul,dekorsy}@ant.uni-bremen.de, {thorsten.schier,patricksvedman}@zte.com.cn

***Abstract*—This paper presents a new distributed precoder design for joint transmission in dense small cell networks. We exploit a numerical method called two-step Jacobi to distribute a centralized precoding scheme among local small cells using cooperation in between. Two variants of the novel distributed precoding algorithm have been specified with one focusing on the distributed update of local precoded signal vectors and another concentrating on the update of the local precoding matrices. The proof of convergence for the proposed algorithms is provided, and simulation results show a satisfying performance of the distributed precoding in a comparison with the benchmark of centralized precoding.**

I. Introduction

Ultra dense deployment of small cells (SCs) with massive multiple-input multiple-output (MIMO) technology as well as millimeter wave communication is becoming a promising approach to meet the wireless data traffic demand in the next decade [1]. However, inter-cell interference (ICI) among the SCs is a dominant limiting factor and can degrade the overall network performance. To handle the ICI problem, one technique termed coordinated multipoint (CoMP) transmission and reception for MIMO communications has drawn great interest in recent years to enhance the performance of multi-cell system by mitigating or eliminating the ICI [2].

For coordination in a multi-SC network, joint transmission can be applied to transform the interference into useful information by properly designing, e.g., a linear precoder using the zero forcing (ZF) or minimum mean square error (MMSE) criterion [3], or a non-linear precoder like dirty paper coding (DPC) [4]. However, these joint precoding schemes are usually carried out in a centralized way resulting in high computational complexity and high latency due to large channel information feedback, which motivates the design of a coordinated precoder in a distributed way. In [5], [6], [7], the authors proposed several distributed precoding schemes to achieve maximization of the system sum rate by minimizing the interference based on statistical channel state information (CSI). Whereas for the distributed precoding based on instantaneous CSI, e.g., MSE based distributed precoders have been presented in [8], [9], where the precoders are jointly designed with the receive filters. However, during the iterative processing, transmitters require feedback from users in order to update the local precoding matrices, which introduces a large overhead and non-negligible latency.

In this paper, we also consider instantaneous CSI for the distributed precoder design, but different to those approaches in [8], [9], we aim to develop a new linear distributed precoding (DiP) algorithm for the joint transmission without interaction between SCs and UEs. The information is only exchanged and processed among local SCs following the In-Network Processing (INP) principle [10]. One numerical approach for INP, the two-step Jacobi (TSJ) [11], is then applied to distribute the processing of centralized MMSE precoding among local SCs, which leads to our two DiP algorithm variants. The novelty regarding the first variant compared to the previous approaches is that each SC calculates the local precoded signal only with parameter vectors instead of precoding matrices exchanged between SCs, such that the amount of exchanged information is reduced particularly for rapidly varying channels. The second DiP variant focuses on the distributed update of local precoding matrices for each SC, which is also preferable in certain scenarios.

The paper is structured as follows: The system model is described in Section II. In Section III, the two-step Jacobi approach is introduced, and following is the development of the proposed DiP algorithms with a proof of convergence. Section IV presents the numerical evaluation of the DiP algorithm. Finally, the paper is concluded in Section V.

II. System description

We are considering a network composed of N_{SC} SCs with connections to a core network or to an edge-cloud where UE data can be downloaded and cached by each SC [12]. These SCs are deployed to jointly serve all N_{UE} UEs for a high quality downlink transmission. Without loss of generality, we assume that each UE and SC is equipped with N_{R} receiving and N_{T} transmitting antennas, respectively, leading to a overall $N_{\text{I}} \times N_{\text{O}}$ MIMO system ($N_{\text{I}} = N_{\text{SC}}N_{\text{T}} > N_{\text{O}} = N_{\text{UE}}N_{\text{R}}$).

For one time instance, a vector $\mathbf{s}$ of length N_{O} containing modulated data vectors for all UEs, i.e., $\mathbf{s} = [\mathbf{s}_1^{\text{T}}, .., \mathbf{s}_{N_{\text{UE}}}^{\text{T}}]^{\text{T}} \in \mathbb{A}^{N_{\text{O}}\times 1}$ is processed by the local precoding matrix $\mathbf{G}_j$ of SC j as shown in Fig. 1. The precoded signals $\mathbf{x}_j \in \mathbb{C}^{N_{\text{T}}\times 1}$ are then transmitted to the UEs simultaneously. At the receiver side, each UE u receives the superposition from all N_{SC} SCs with additive Gaussian noise $\mathbf{n}_j \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I}_{N_{\text{R}}})$ and then re-scales the received signals by a factor of β which depends on the precoder and the power constraint. Since only the precoding problem is focused here, no post processing is

ISBN 978-3-8007-4426-8

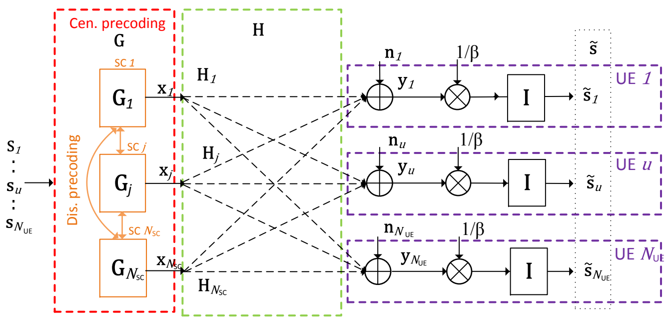

Fig. 1. Diagram of signal processing in central/distributed precoding

considered (denoted by a block with the identity matrix in Fig. 1) at UEs. The UE u's recovered signal is obtained as $\tilde{\mathbf{s}}_u = \frac{1}{\beta}(\sum_{j=1}^{N_{\text{SC}}} \mathbf{H}_{uj}\mathbf{x}_j + \mathbf{n}_j)$, where $\mathbf{H}_{uj}$ denotes the locally known channel between SC j and UE u. Taking the whole system into account, the total output signal $\tilde{\mathbf{s}} = [\tilde{\mathbf{s}}_1^{\text{T}}, .., \tilde{\mathbf{s}}_{N_{\text{UE}}}^{\text{T}}]^{\text{T}}$ can be written as

$$\tilde{\mathbf{s}} = \frac{1}{\beta}(\mathbf{HGs} + \mathbf{n}) \tag{1}$$

where the UE data vector $\mathbf{s}$ is precoded by a central precoding matrix $\mathbf{G} = [\mathbf{G}_1^{\text{T}}, .., \mathbf{G}_{N_{\text{SC}}}^{\text{T}}]^{\text{T}}$. The entire channel matrix $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_{N_{\text{SC}}}]$ is composed of local channel matrices $\mathbf{H}_j = [\mathbf{H}_{1j}^{\text{T}}, .., \mathbf{H}_{uj}^{\text{T}}, .., \mathbf{H}_{N_{\text{UE}}j}^{\text{T}}]^{\text{T}}$ of SCs j. The noise vector is stacked as $\mathbf{n} = [\mathbf{n}_1^{\text{T}}, ..., \mathbf{n}_{N_{\text{SC}}}^{\text{T}}]^{\text{T}}$. We aim to minimize the mean square error between the received signals and the original data considering a total transmit power constraint $P\sigma_s^2$, where σ_s^2 is the average power of the modulation alphabet. The corresponding objective problem reads as

$$\min_{\mathbf{G},\beta} \mathrm{E}\left\{\|\mathbf{s} - \tilde{\mathbf{s}}\|^2\right\} \quad \text{s.t.} \quad \mathrm{E}\left\{\|\mathbf{Gs}\|^2\right\} = P\sigma_s^2, \tag{2}$$

of which the solution is the MMSE precoding matrix $\mathbf{G}$ with the scaling factor β:

$$\mathbf{G} = \beta(\mathbf{H}^{\text{H}}\mathbf{H} + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I})^{-1}\mathbf{H}^{\text{H}} = \beta\mathbf{G}' \tag{3a}$$

$$\text{with } \beta = \sqrt{\frac{P\sigma_s^2}{\text{trace}(\mathbf{G}'\mathbf{G}'^{\text{H}})}}. \tag{3b}$$

Correspondingly, the total transmit vector $\mathbf{x}$ over all N_{SC} SCs can be linearly precoded as

$$\mathbf{x} = \mathbf{Gs} = \beta(\mathbf{H}^{\text{H}}\mathbf{H} + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I})^{-1}\mathbf{H}^{\text{H}}\mathbf{s} = \mathbf{A}^{-1}\mathbf{b} \tag{4}$$

where $\mathbf{A} = \mathbf{H}^{\text{H}}\mathbf{H} + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I}$ is a Hermitian positive definite matrix and the vector $\mathbf{b}$ is defined as $\mathbf{b} = \beta\mathbf{H}^{\text{H}}\mathbf{s}$. Note that in our system, no dedicated processing unit is deployed for the centralized precoding, thus we aim to achieve the joint precoding in a distributed way using the centralized precoding as a benchmark.

III. Distributed Precoder Design

In this section, we first introduce a numerical method named two-step Jacobi for solving a non-diagonally dominant linear equation system in an iterative way. Then we will show the development of our TSJ based distributed precoding algorithm with a convergence analysis in the following.

A. *Two-step Jacobi method*

For a linear system like $\mathbf{Ax} = \mathbf{b}$ in (4), where the matrix $\mathbf{A} = \mathbf{D}+\mathbf{R}$ can be decomposed into a block diagonal matrix $\mathbf{D}$ and a block off-diagonal matrix $\mathbf{R}$, the vector $\mathbf{x}$ could be solved for, e.g., by the widely known Jacobi method with parallel and distributed implementation if for the spectral radius of the iteration matrix holds $\rho(\mathbf{D}^{-1}\mathbf{R}) < 1$ [13]. However, in our system a Rayleigh fading channel is assumed, so that the system matrix $\mathbf{A}$ is normally non-diagonally dominant, i.e., $\rho(\mathbf{D}^{-1}\mathbf{R}) > 1$. To handle this problem, the TSJ method is applied, which introduces an additional relaxation parameter γ to enhance the diagonal dominance of the matrix $\mathbf{A}$. To this end, a modified linear system $\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}$ in (4) is given by

$$[\gamma\mathbf{D} + \mathbf{R}]\,\mathbf{x} = \mathbf{b} + (\gamma - 1)\mathbf{Dx}, \tag{5}$$

where $\tilde{\mathbf{A}} = \gamma\mathbf{D} + \mathbf{R}$ and $\tilde{\mathbf{b}} = \mathbf{b} + (\gamma - 1)\mathbf{Dx}$ are defined. Then, we can solve for the vector $\mathbf{x}$ in an iterative fashion as $\mathbf{x}^m = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}^{m-1}$, where the vector $\tilde{\mathbf{b}}^{m-1} = \mathbf{b}+(\gamma-1)\mathbf{Dx}^{m-1}$ is also updated in iteration $m-1$. More specifically, the update of vector $\mathbf{x}^m$ can be written as

$$\mathbf{x}^m = (\gamma\mathbf{D} + \mathbf{R})^{-1}\mathbf{b} + (\gamma - 1)\left(\gamma\mathbf{I} + \mathbf{D}^{-1}\mathbf{R}\right)^{-1}\mathbf{x}^{m-1} \tag{6}$$

The convergence can always be ensured with an appropriate γ, which will be proven in section III-C.

Moreover, to avoid the matrix inversion, we can now apply the Jacobi method to solve for $\mathbf{x}^m$ from the linear equation $\tilde{\mathbf{A}}\mathbf{x}^m = \tilde{\mathbf{b}}^{m-1}$, where the matrix $\tilde{\mathbf{A}}$ is diagonally dominant due to the enhancement by γ. The vector $\mathbf{x}^m$ from the outer loop m can then be obtained in an inner loop k with a diagonal matrix inversion:

$$\mathbf{x}^{m,k} = (\gamma\mathbf{D})^{-1}(\tilde{\mathbf{b}}^{m-1} - \mathbf{Rx}^{m,k-1}). \tag{7}$$

Here, we define a total number K for the inner loop, once inner iteration $k = K$ is reached, then the vector $\mathbf{x}^m$ of the outer iteration is determined by $\mathbf{x}^m = \mathbf{x}^{m,K}$.

As a variant of the Jacobi method, the TSJ method is also capable of being implemented in a distributed way, and its convergence can be guaranteed by a proper design which the Jacobi method cannot for the current system. Therefore, we exploit the TSJ method to develop our DiP algorithm.

B. *TSJ based distributed precoding algorithms*

For the design of the distributed precoder, we propose two variants of the DiP algorithm considering different application circumstances. One variant of the DiP algorithm focuses on the distributed calculation of local precoded signals with parameter vectors exchanged between SCs if the channel between SCs and UEs is fast time varying. Whereas, when the channel is static for a long coherence time, another variant is then developed for the distributed update of the local precoding matrices using parameter matrices instead of vectors exchanged per iteration between SCs. In the following, these two variants of the DiP algorithm will be explained in detail.

1) Variant 1. DiP on local precoded signals: As discussed in section III-A, the TSJ approach can be applied to obtain an nonnormalized central precoded signal $\mathbf{x}'$ by solving the linear equation $(\mathbf{H}^{\mathrm{H}}\mathbf{H} + \sigma_n^2/\sigma_s^2\mathbf{I})\mathbf{x}' = \mathbf{H}^{\mathrm{H}}\mathbf{s}$ (assuming $\beta = 1$ in (4)) according to the iterative update (6), (7) and get

$$\mathbf{x}'^{m,k} = (\gamma\mathbf{D})^{-1}(\mathbf{H}^{\mathrm{H}}\mathbf{s} + (\gamma - 1)\mathbf{D}\mathbf{x}'^{m-1} - \mathbf{R}\mathbf{x}'^{m,k-1}), \quad (8)$$

where the vector $\mathbf{x}'^{m,k} = [(\mathbf{x}_1^{m,k})^{\mathrm{T}}, (\mathbf{x}_2^{m,k})^{\mathrm{T}}, ..., (\mathbf{x}_{N_{\mathrm{SC}}}^{m,k})^{\mathrm{T}}]^{\mathrm{T}}$ is composed of local precoded signals $\mathbf{x}_j^{m,k}$. The block diagonal matrix $\mathbf{D}$ is detailed as $\mathbf{D} = \mathrm{blkdiag}\{(\mathbf{H}_1^{\mathrm{H}}\mathbf{H}_1 + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I})^{-1}, (\mathbf{H}_2^{\mathrm{H}}\mathbf{H}_2 + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I})^{-1}, .., (\mathbf{H}_{N_{\mathrm{SC}}}^{\mathrm{H}}\mathbf{H}_{N_{\mathrm{SC}}} + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I})^{-1}\}$, and the matrix $\mathbf{R}$ is defined as $\mathbf{R} = (\mathbf{H}^{\mathrm{H}}\mathbf{H} + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I}) - \mathbf{D}$.

Now, we attempt to achieve a distributed update of precoded signals among local SCs since the linear system of (8) is decomposable. Thus, we can split the centralized update of the vector $\mathbf{x}'$ into local updates among SCs in parallel. For each SC j, the update of local precoded signals $\mathbf{x}_j^{m,k}$ within two loops of TSJ can be summarized into one equation as

$$\mathbf{x}_j^{mK+k} = \left[\gamma\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I}\right]^{-1}\Bigg[(\gamma - 1)\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j\mathbf{x}_j^{(m-1)K} - \sum_{i\neq j}^{N_{\mathrm{SC}}}\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_i\mathbf{x}_i^{mK+k-1} + \mathbf{H}_j^{\mathrm{H}}\mathbf{s}\Bigg], \quad (9)$$

where K is the maximum number of inner iterations of the TSJ approach, when inner iteration k reaches K, the outer iteration m increases by 1. According to (9), each SC j updates its local precoded signals $\mathbf{x}_j^{mK+k}$ only requiring external parameter vectors $\mathbf{H}_i\mathbf{x}_i^{mK+k-1} \in \mathbb{C}^{N_{\mathrm{O}}\times 1}$ from other SCs i, which results in communication overhead $\mathcal{O}_1$ on the SC-SC links. When the outer iteration is terminated at $m = N_{\mathrm{it}}$, all SCs exchange information on the power of local precoded signals i.e., $\|\mathbf{x}_j^{N_{\mathrm{it}}}\|^2$, which can nearly be neglected when counting the overhead. Then, each SC normalizes the transmitted signal with a locally computed factor β to fulfill the power constraint $P\sigma_s^2$:

$$\mathbf{x}_j = \beta\mathbf{x}_j^{N_{\mathrm{it}}} \text{ with } \beta = \sqrt{\frac{P\sigma_s^2}{\sum_{j=1}^{N_{\mathrm{SC}}}\|\mathbf{x}_j^{N_{\mathrm{it}}}\|^2}}. \quad (10)$$

So if we assume that each SC broadcasts the local vector $\mathbf{H}_i\mathbf{x}_i$ during the update, then we can amount the total overhead produced per iteration within the network to $\mathcal{O}_1 = N_{\mathrm{SC}} \cdot N_{\mathrm{O}}$ complex numbers.

2) Variant 2. DiP on local precoding matrix: Different to Variant 1, the DiP algorithm for variant 2 is focused on the development of local precoding matrices $\mathbf{G}_j$ for each SC j. Assuming the scaling factor $\beta = 1$ in (3a), we then can obtain a linear system $(\mathbf{H}^{\mathrm{H}}\mathbf{H} + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I})\mathbf{G}' = \mathbf{H}^{\mathrm{H}}$ for an unknown matrix $\mathbf{G}'$. By using the TSJ approach, the matrix $\mathbf{G}'$ can also be solved for within two iterative steps given by

$$\mathbf{G}'^{m,k} = (\gamma\mathbf{D})^{-1}(\mathbf{H}^{\mathrm{H}} + (\gamma - 1)\mathbf{D}\mathbf{G}'^{m-1} - \mathbf{R}\mathbf{G}'^{m,k-1}). \quad (11)$$

Here the matrix $\mathbf{D}$ and $\mathbf{R}$ are still defined as the block diagonal matrix and off-block diagonal matrix of $(\mathbf{H}^{\mathrm{H}}\mathbf{H} + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I})$ like in variant 1. Note that the matrix $\mathbf{G}'^{m,k} = [(\mathbf{G}_1^{m,k})^{\mathrm{T}}, (\mathbf{G}_2^{m,k})^{\mathrm{T}}, ..., (\mathbf{G}_{N_{\mathrm{SC}}}^{m,k})^{\mathrm{T}}]^{\mathrm{T}}$ is composed of local precoding matrices $\mathbf{G}_j^{m,k}$ of SCs j and the inverse $\mathbf{D}^{-1}$ in (11) is also decomposable. Therefore, the central update of $\mathbf{G}'^{m,k}$ can be distributed among SCs leading to the local update of precoding matrix $\mathbf{G}_j^{mK+k}$:

$$\mathbf{G}_j^{mK+k} = \left[\gamma\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j + \frac{\sigma_n^2}{\sigma_s^2}\mathbf{I}\right]^{-1}\Bigg[(\gamma - 1)\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j\mathbf{G}_j^{(m-1)K} - \sum_{i\neq j}^{N_{\mathrm{SC}}}\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_i\mathbf{G}_i^{mK+k-1} + \mathbf{H}_j^{\mathrm{H}}\Bigg]. \quad (12)$$

The outer iteration m increases by 1 if the inner iteration k reaches the maximum number K. For the update of the local precoding matrix $\mathbf{G}_j^{mK+k}$, SC j requires external information of matrices $\mathbf{H}_i\mathbf{G}_i^{mK+k-1} \in \mathcal{C}^{N_{\mathrm{O}}\times N_{\mathrm{O}}}$ from other SCs i. Thus, every SC shares its local matrix over SC-SC links leading to an overhead of $\mathcal{O}_2 = N_{\mathrm{SC}} \cdot N_{\mathrm{O}}^2$ complex numbers per update, which is much higher compared to the variant 1. Once the update of local precoding matrices terminates when the maximum number N_{it} of outer iteration is reached, i.e., $m = N_{\mathrm{it}}$, all SCs share information on the power of local precoded signals i.e., $\|\mathbf{G}_j^{N_{\mathrm{it}}}\mathbf{s}\|^2$, with each other to obtain the power scaling factor β for normalizing the local precoding matrices as

$$\mathbf{G}_j = \beta\mathbf{G}_j^{N_{\mathrm{it}}} \text{ with } \beta = \sqrt{\frac{P\sigma_s^2}{\sum_{j=1}^{N_{\mathrm{SC}}}\|\mathbf{G}_j^{N_{\mathrm{it}}}\mathbf{s}\|^2}}. \quad (13)$$

Then, the local transmitted signals $\mathbf{x}_j$ can be obtained from UE data $\mathbf{s}$ by the linear expression $\mathbf{x}_j = \mathbf{G}_j\mathbf{s}$.

C. Convergence analysis of the TSJ-DiP algorithm

In general, both variants of the DiP algorithm adopt the TSJ approach to implement the centralized precoding in a distributed fashion. Thus, the convergence of the DiP algorithm mainly depends on the TSJ method. As discussed in section III-A, the inner update of TSJ method is used only to calculate the vector $\mathbf{x}^m$ from the outer iteration, while the update of $\mathbf{x}^m$ will finally converge to the system optimal solution of $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$, which can be rewritten as

$$\mathbf{x}^* = (\gamma\mathbf{D} + \mathbf{R})^{-1}\mathbf{b} + (\gamma - 1)\left(\gamma\mathbf{I} + \mathbf{D}^{-1}\mathbf{R}\right)^{-1}\mathbf{x}^*. \quad (14)$$

Now, we define an update residual $\mathbf{e}^m = \mathbf{x}^m - \mathbf{x}^*$ and subtract (14) from (6), obtaining

$$\mathbf{e}^m = (\gamma - 1)\left(\gamma\mathbf{I} + \mathbf{D}^{-1}\mathbf{R}\right)^{-1}\mathbf{e}^{m-1} = (\mathbf{M})^m\mathbf{e}^0, \quad (15)$$

where the iteration matrix $\mathbf{M} = (\gamma - 1)\left(\gamma\mathbf{I} + \mathbf{D}^{-1}\mathbf{R}\right)^{-1}$ is defined. According to [13], for the update residual holds $\mathbf{e}^m \to \mathbf{0}$ for $m \to \infty$ if the spectral radius of iteration matrix is smaller than 1, i.e., $\rho(\mathbf{M}) < 1$. Thus, the convergence of TSJ-DiP basically depends on the system matrix $\mathbf{A}$ (i.e., $\mathbf{D}$ and $\mathbf{R}$) as well as the relaxation parameter γ. Moreover, we define $\nu_n, n = 1, .., N_{\mathrm{O}}$ as eigenvalues of the iteration matrix $\mathbf{D}^{-1}\mathbf{R}$ of the Jacobi method [13], which in our system fulfills the condition $\rho(\mathbf{D}^{-1}\mathbf{R}) = \max_{n=1,..,N_O}\{|\nu_n|\} > 1$, and $\min_{n=1,..,N_O}\{\nu_n\} > -1$ since the matrix $\mathbf{D}^{-1}\mathbf{A} = \mathbf{D}^{-1}\mathbf{R} + \mathbf{I}$ is also a Hermitian matrix and has non-negative eigenvalues. Then, the spectral radius $\rho(\mathbf{M})$ can be further determined as

$$\rho(\mathbf{M}) = \max_{n=1,..,N_O}\left|\frac{\gamma-1}{\gamma+\nu_n}\right| = \frac{\gamma-1}{\min\limits_{n=1,..,N_O}|\gamma+\nu_n|}. \tag{16}$$

Proof: We define that η is an eigenvalue of matrix $\mathbf{M}$ and $\mathbf{z}$ is an eigenvector of $\mathbf{M}$, i.e., $(\gamma-1)\left(\gamma\mathbf{I}+\mathbf{D}^{-1}\mathbf{R}\right)^{-1}\mathbf{z} = \eta\mathbf{z}$, then, we have a relation

$$\mathbf{D}^{-1}\mathbf{R}\mathbf{z} = \left(\frac{(\gamma-1)}{\eta} - \gamma\right)\mathbf{z}. \tag{17}$$

It can be seen from the equation (17) that $\left(\frac{(\gamma-1)}{\eta} - \gamma\right)$ equals to an eigenvalue ν_n of matrix $\mathbf{D}^{-1}\mathbf{R}$, such that we can obtain $\eta = \frac{\gamma-1}{\gamma+\nu_n}$. To this end, the maximum eigenvalue of matrix $\mathbf{M}$ can be written as $\eta_{\max} = \frac{\gamma-1}{\min\limits_{n=1,...,N_O}|\gamma+\nu_n|} = \rho(\mathbf{M})$. ■

To ensure that the TSJ-DiP algorithm converges, an appropriate relaxation parameter γ is required. For the inner update of the TSJ approach (7), the spectral radius of iteration matrix $\rho\left(\frac{1}{\gamma}\mathbf{D}^{-1}\mathbf{R}\right) < 1$ should be fulfilled, hence we have $\gamma > \max_{n=1,..,N_O}|\nu_n| > 1$. For the outer update, as discussed above, a lower bound of $\rho(\mathbf{M})$ can be approximated by γ for a minimum-maximum problem:

$$\min_{\gamma}\max_{n=1,..,N_O}\left|\frac{\gamma-1}{\gamma+\nu_n}\right| = \min_{\gamma}\frac{\gamma-1}{\min\limits_{n=1,..,N_O}|\gamma+\nu_n|}. \tag{18}$$

There is no optimal solution for γ, since $\frac{\gamma-1}{\min\limits_{n=1,...,N_O}|\gamma+\nu_n|}$ monotonously increases with a growing γ. Nevertheless, the conditions $\gamma > \max_{n=1,..,N_O}|\nu_n| > 1$ and $\min_{n=1,..,N_O}\{\nu_n\} > -1$ should always be fulfilled. Therefore, the spectral radius $\rho(\mathbf{M})$ satisfies

$$\frac{\max\limits_{n=1,..,N_O}|\nu_n| - 1}{\min\limits_{n=1,..,N_O}\left|\max\limits_{n=1,..,N_O}|\nu_n| + \nu_n\right|} < \rho(\mathbf{M}) < 1, \tag{19}$$

which ensures that the TSJ-DiP algorithm converges. For the distributed implementation, a proper γ can be chosen heuristically by local SCs depending on the current system.

IV. Performance Evaluation

In this section, the performance of the proposed distributed precoding algorithm is evaluated and compared with the benchmark, i.e., centralized MMSE precoding. Besides, the communication overhead for both DiP variants is also analyzed. For the evaluation, we consider an example of indoor hotspot (InH) scenario [14], where $N_{\mathrm{SC}} = 4$ SCs are deployed on the same floor with a constant distance of 30 m in between and are connected with ideal links. $N_{\mathrm{UE}} = 12$ UEs are uniformly dropped each in an individual distance d to the SC. The path-loss model is defined as $PL = 16.9\log_{10} d + 32.8 + 20\log_{10} f_c$, where the carrier frequency is $f_c = 3.5$ GHz [14]. We set the max. no. of inner iteration to $K = 1$ for the proposed algorithm, then, the total number of iterations is denoted as N_{it} in the following evaluation.

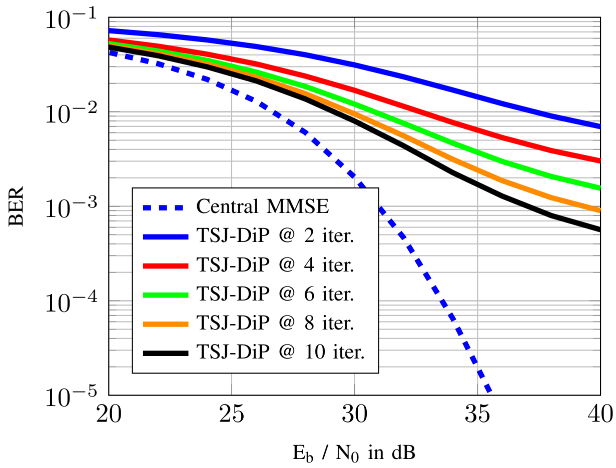

Fig. 2. BER performance for TSJ-DiP algorithm and centralized MMSE precoding, $N_{\mathrm{SC}} = 4, N_{\mathrm{UE}} = 12, N_{\mathrm{T}} = 4, N_{\mathrm{R}} = 1$.

A. BER performance

To investigate the performance of the proposed DiP algorithms, we consider an uncoded transmission for simplicity. Nevertheless, an extension to coded transmission is straightforward and promising. Here we use the bit error rate (BER) as an evaluation metric for QPSK symbols For both DiP variants, their BER performance is identical for the same number of iterations due to the linear relation between each other. In Fig. 2, the BER performance of the DiP for various numbers of iterations is shown in comparison to the centralized precoding. In general, the performance of DiP is improved with increasing iterations, e.g., a gain of roughly 5 dB is achieved by DiP with 4 iterations compared to 2 iterations for a BER of 10^{-2}. Moreover, it can be seen in the figure that the difference between DiP and centralized precoding becomes large for a higher SNR range, since the impact of noise on the performance becomes weaker for both DiP and centralized precoding at high SNR, while the interference generated from DiP with an insufficient number of iterations degrades the performance. Thus, the number of iterations has a relatively high influence at high SNR compared to low SNR. The convergence behavior of the DiP algorithm is more explicitly illustrated in Fig. 3, where the BER performance of DiP reduces with an increased number of iterations and can approach the centralized precoding with a sufficient number of iterations, while for a high SNR, more iterations are required for the DiP algorithm to converge.

B. Communication overhead

For the distributed precoding as discussed in III, either local precoded signal vectors (variant 1) or local precoding matrices (variant 2) are exchanged between SCs during the

ISBN 978-3-8007-4426-8

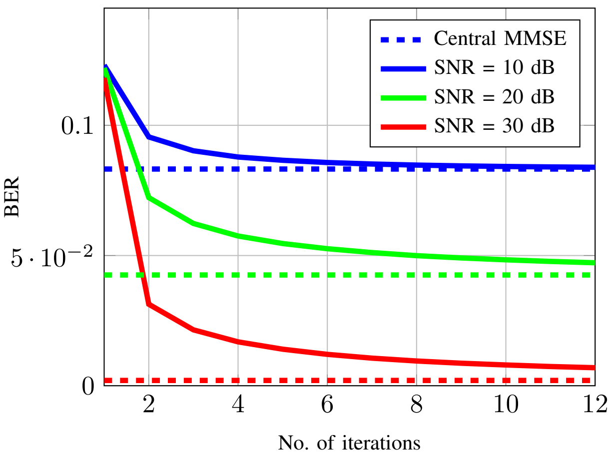

Fig. 3. Convergence behavior of TSJ-DiP algorithm at different SNRs $N_{SC} = 4, N_{UE} = 12, N_T = 4, N_R = 1$.

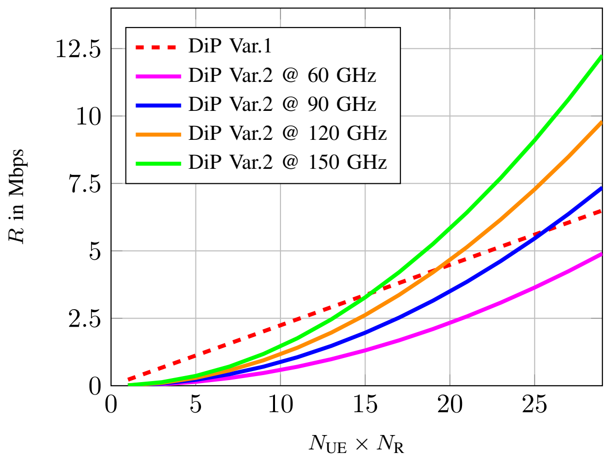

Fig. 4. Communication overhead rate R vs. total output dimension w.r.t. different carrier frequencies for both DiP variant 1 and variant 2, $N_Q = 4$.

iterative processing. In variant 1, for each input UE data vector per time instance, a corresponding precoded signal vector is re-computed. If we consider a multi-carrier system, e.g., OFDM in LTE, each Tx signal needs to be updated within an OFDM symbol time duration $\Delta t = 71.4$ μs. A total bandwidth $B_{tot} = 1.4$ MHz is considered to be spanned by subcarriers with frequency spacing $\Delta f = 15$ KHz each. We also assume that each Tx symbol is quantized into N_Q bits for the transmission. To this end, the overhead rate R_1 for variant 1 per SC-SC link over N_{it} iterations is defined:

$$R_1 = \frac{OV_1 \cdot N_Q \cdot N_{it} \cdot B_{tot}}{N_{SC} \cdot \Delta t \cdot \Delta f} = \frac{N_O \cdot N_Q \cdot N_{it} \cdot B_{tot}}{\Delta t \cdot \Delta f}. \quad (20)$$

While for DiP variant 2, which is applicable for the case when the radio channel does not change significantly over time (coherence time) and frequency (coherence frequency), the precoding matrix only needs to be updated once regardless of the input UE data. As the UE moving speed $v = 3$ km/h is defined in InH scenario, the corresponding coherence time can be ca calculated as $T_c = \frac{c}{8 f_c \cdot v}$ [15], where c is the speed of light. Moreover, the coherence bandwidth is defined as $B_c = \frac{1}{5\tau_{rms}}$ [16], where $\tau_{rms} = 39$ ns is the root-mean-square (RMS) delay spread defined for InH NLOS transmission [14]. The overhead rate R_2 for variant 2 is then defined:

$$R_2 = \frac{OV_2 \cdot N_Q \cdot N_{it} \cdot B_{tot}}{N_{SC} \cdot T_c \cdot B_c} = \frac{N_O^2 \cdot N_Q \cdot N_{it} \cdot B_{tot}}{T_c \cdot B_c}. \quad (21)$$

Now, if the SCs are assumed to use extreme high carrier frequencies e.g., $f_c = 60, 90, 120, 150$ GHz to transmit the UE data. Then, we can observe the growth of the data rate w.r.t. the system output dimension $N_{UE} \cdot N_R$ for both variants in Fig. 4, where the data rate of variant 1 increases linearly with the total output dimension, while variant 2 is in a quadratic growth, but in general both variants can be supported by several backhaul (BH) technologies, e.g., sub 6-GHz, millimeter wave or Ethernet [17]. When we compare both variants, variant 2 may require a higher overhead rate than variant 1 for a large system output at high frequencies, e.g., when $f_c = 150$ GHz, $R2 > R1$ for $N_O > 15$. As the channel coherence time becomes small at high frequencies, SCs need to exchange parameter matrices more frequently for variant 2, while the overhead rate of variant 1 is not limited by the coherence time but the fixed symbol duration, then the variant 1 algorithm is preferred at high frequencies. Whereas for a system with small output, e.g., $N_O < 14$, the variant 2 can be applied, since its overhead rate is lower compared to variant 1 particularly for a low carrier frequency.

V. Conclusion

In this paper, we adopt an iterative method named two-step Jacobi approach to develop our novel distributed precoding algorithm for a joint transmission. Two variants of the DiP algorithm have been proposed for the iterative update of either transmitted signals or precoding matrices to cope with different application scenarios. We also prove the convergence of the proposed algorithm when the relaxation parameter is appropriately set. Promising numerical results have been achieved by the proposed DiP algorithm, but approaches to accelerate the convergence speed or reducing the communication overhead are still of interest for future work. In addition, from a practical perspective, the distributed precoding with coded transmission is also important and the performance is currently under investigation.

References

[1] X. Ge, S. Tu, and et al., “5G Ultra-Dense Cellular Networks,” *IEEE Wireless Communications*, vol. 23, no. 1, pp. 72–79, Feb. 2016.

[2] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, “Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges,” *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, February 2012.

[3] M. Joham, W. Utschick, and J. A. Nossek, “Linear transmit processing in MIMO communications systems,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2700–2712, Aug 2005.

[4] S. Shamai and B. M. Zaidel, “Enhancing the cellular downlink capacity via co-processing at the transmitting end,” in *Vehicular Technology Conference, 2001. VTC 2001 Spring*, vol. 3, May 2001, pp. 1745–1749.

[5] Y. Zou, C. Li, and L. Yang, “Distributed precoding for multicell MIMO networks,” in *Conference on Wireless Communications and Signal Processing (WCSP), 2010*, Oct 2010, pp. 1–4.

[6] E. Björnson and B. Ottersten, “On the principles of multicell precoding with centralized and distributed cooperation,” in *Conference on Wireless Communications Signal Processing (WCSP), 2009*, Nov 2009, pp. 1–5.
[7] E. Bjornson, R. Zakhour, D. Gesbert, and B. Ottersten, “Cooperative Multicell Precoding: Rate Region Characterization and Distributed Strategies With Instantaneous and Statistical CSI,” *IEEE Transactions on Signal Processing*, vol. 58, no. 8, Aug 2010.
[8] T. E. Bogale, L. Vandendorpe, and B. K. Chalise, “MMSE Transceiver design for coordinated base station systems: Distributive algorithm,” in *2010 Asilomar Conference on Signals, Systems and Computers*, Nov 2010, pp. 1222–1226.
[9] Z. Kong, S. Yang, F. Wu, S. Peng, L. Zhong, and L. Hanzo, “Iterative Distributed Minimum Total MSE Approach for Secure Communications in MIMO Interference Channels,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 594–608, March 2016.
[10] H. Paul, J. Fliege, and A. Dekorsy, “In-Network-Processing: Distributed Consensus-Based Linear Estimation,” *IEEE Communications Letters*, vol. 17, no. 1, pp. 59–62, Jan. 2013.
[11] V. Manoranjan and M. Gomez, “A two-step Jacobi-type iterative method,” *Computers and Mathematics with Applications*, vol. 34, no. 1, pp. 1–9, 1997.
[12] Y. Ugur, Z. H. Awan, and A. Sezgin, in *WSA 2016; 20th International ITG Workshop on Smart Antennas*.
[13] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, USA, 1989.
[14] 3GPP TS 36.814, “Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access; Further advancements for E-UTRA physical layer aspects (Release 9),” Mar. 2010.
[15] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
[16] S. C. Yang, *OFDMA system analysis and design*. Artech House Mobile Communication Series, 2010.
[17] iJOIN project, “D4.2- network-layer algorithms and network operation and management: candidate technologies specification,” Tech. Rep., Nov. 2014.

