POSTPRINT VERSION. The final version is published here :
Ouameur, M. A. et Massicotte, D. (2019, 2-6 Sept. 2019). Efficient Distributed Processing for Large Scale MIMO Detection. Communication présentée 2019 27th European Signal Processing Conference (EUSIPCO), A Coruna.
doi: 10.23919/EUSIPCO.2019.8902787

# Efficient Distributed Processing for Large Scale MIMO Detection

Messaoud Ahmed Ouameur and Daniel Massicotte

Université du Québec à Trois-Rivières, Department of Electrical and Computer Engineering,
3351, Boul. des Forges, Trois-Rivières, Québec, Canada
Laboratoire des Signaux et Systèmes Intégrés,
{messaoud.ahmed.ouameur, daniel.massicotte}@uqtr.ca

***Abstract*—** In large scale multiple-input multiple-output (MIMO), high spectral and energy efficiencies comes at the expense of a high computational complexity baseband processing. Many contributions have been proposed to reduce such complexity using matrix inversion approximation techniques for instance. On the other hand, to reduce the constraint on the interconnects' bandwidth, fewer decentralized processing techniques have emerged. Here, we propose a computationally efficient technique based on embedding one single Gauss-Seidel iteration within every ADMM based detection iteration. The simulations are performed using an LTE-like TDD-OFDM frame structure and waveform, under perfect and non-perfect channel state information (CSI). Early results reveal that the proposed ADMM-GS algorithm can outperform the centralised GS based technique processing in a high SNR region and high load regime. In addition ADMM-GS' performance exhibits relatively less sensitivity to channel estimation error; a characteristic inherited from the centralised GS technique.

***Index Terms*—**Large scale multiple-input multiple-output (MIMO), zero forcing (ZF) detection, Maximum ratio combining (MRC), receiver combining, Gauss Seidel (GS), alternating direction method of multipliers (ADMM).

## I. Introduction

It has been recognized that large-scale MIMO (also referred to as massive MIMO) constitutes one of the main disruptive technology directions for 5G [1][2]. Massive MIMO is a form of multiuser MIMO where the number of serving antennas at the base transceiver station (BS) is much larger than the number of user terminals (UTs) served within each radio resource element. Because of its advantages in terms of very high spectral efficiency (sum rates), increased reliability, and power efficiency, massive MIMO has been the subject of a large amount of research activities [3]. Given the large number of antennas, reliance on time division duplex TDD channel reciprocity is essential [2]. Basically, massive MIMO systems exploit this reciprocity to estimate the channel responses on the uplink [4] and then use the acquired channel state information (CSI) for both uplink receive combining/detection and downlink transmit precoding/beamforming of the users' payload. Under favorable channel conditions and/or as the number of antennas increases, the UTs' channels are mutually orthogonal which makes linear processing based on maximum ratio combining (MRC), zero forcing (ZF) detection or minimum mean squared error (MMSE) detection, a suitable and optimal choice [5].

To reduce the implementation complexity of ZF and MMSE techniques, matrix inversion approximations such as polynomial expansion (PE) were proposed [6]. It has been applied to precoding in [6] and [7] under the form of a truncated polynomial expansion (TPE) and Neumann series expansion (NSE) respectively. Recently, a technique based on Gauss Seidel (GS) was shown to outperform NSE due to its fast convergence at considerably lower computational complexity [8]. However, this comes at the expense of higher latency and lower throughput [9]. To counter the load increase effect, GS can still afford using more iterations while maintaining lower computational complexity, albeit at the expense of reduced throughput [9]. It has therefore been argued to resort to exact matrix inversion [10]. Herein, direct matrix inversion based on Cholesky decomposition is considered as a reference from the performance stand point[1].

Nevertheless, all these centralized processing techniques still impose stringent constraints on the interconnects' bandwidth between the radio heads and the central processing unit. They also do suffer from increasing latency as the number of antennas is increased which affects scalability to a larger system. Distributed, or decentralised, Massive MIMO processing has been introduced to overcome such limitations. The authors in [11] have proposed an ADMM based processing where a BS is divided into a cluster of small independent groups of radio heads with fewer antenna each. By exploiting the ADMM framework [12], a novel decentralized processing is suggested. However, the algorithm exhibits higher computational complexity compared to a decentralized conjugate gradient (CG) method [12]. Other techniques do also rely on the iterative exchange of consensus information which limits the achievable rate due to the inherently higher latency [13]. This is the case in the distributed processing across the antennas for coordinated multipoint (CoMP) and cloud access radio networks (C-RANs) [14] [16]. To mitigate such latency, the approach in [13] is to avoid sharing the consensus information among the clusters by proposing a decentralized feedforward architecture. Our approach is still based on ADMM framework [12] but focuses on reducing the computational complexity by introducing one

[1] Scaling up optimized QR decomposition implementation to massive MIMO is quite expensive in terms of hardware [15- section 5.2]. However, it has been argued in [15] that under favorable channel conditions the Gram matrix become diagonally dominant which can be exploited to ease the implementation of the QR decomposition method.

single GS iteration per ADMM outer steps. As such our contributions are:

1. We propose a computationally efficient ADMM-GS based decentralized algorithm wherein one GS- based detection iteration is embedded as part of the ADMM-based detection steps. As such, one would expect reduced computational complexity and lower latency as GS requires fewer iteration to converge.
2. Discuss the effect of channel estimation errors as the decentralized processing relies on dividing the large number of antennas into a cluster wherein each group has fewer number of antenna elements. One would therefore, question if such approach would be sensitive to channel estimation errors now that the effect of large-scale have been reduced.

The paper is organized as follows: Section II presents the uplink signal model and the ZF detection technique. Section III details the proposed distributed detection. Using an LTE-like TDD-OFDM waveform/frame structure early performance results are discussed in section IV. Finally, the conclusions are drawn and some future research directions are outlined in section V.

## II. Signal Model and ZF Linear Detection Techniques

We consider an uplink transmission where $K$ single antenna UTs are communicating with a BS equipped with $M$ antennas (where $M \gg K$) in TDD duplex mode using OFDM modulation scheme. For the sake of simplicity, we consider a base band equivalent channel and expressions per subcarrier where the subcarrier index is suppressed. The data signal of the $k^{\text{th}}$ UT is denoted by $s_k \in \mathbb{C}$ and is normalized to unit power. The vector $\mathbf{h}_k \in \mathbb{C}^{M\times 1}$ represents the corresponding channel which is modeled, for simulation purposes, as a flat Rayleigh fading channel vector whose entries are assumed to be independent and identically distributed (i.i.d) with zero mean and unit variance. We model the received signal at the BS as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{M\times 1}$, $\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \cdots & \mathbf{h}_K\end{bmatrix}$ is the channel matrix and $\mathbf{s} = \begin{bmatrix}s_1 & s_2 & \cdots & s_K\end{bmatrix}^T$. $\mathbf{n} \in \mathbb{C}^{M\times 1}$ represents the additive receiver noise vector whose entries have a zero mean and a variance equal to $\sigma^2$.

The ZF detection technique applies $\mathbf{W} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}\mathbf{H}^H = \begin{bmatrix}\mathbf{w}_1, & ..., & \mathbf{w}_K\end{bmatrix} \in \mathbb{C}^{M\times K}$ on the received signal $\mathbf{y}$ to estimate the UTs' transmitted signal $\mathbf{s}$ as

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{y} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}\mathbf{H}^H\mathbf{y} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}\mathbf{y}_{MRC} = \mathbf{\Delta}_{ZF}\mathbf{y}_{MRC} \tag{2}$$

where $\mathbf{y}_{MRC} = \mathbf{y}_{MF} \triangleq \mathbf{H}^H\mathbf{y}$ [2]. Notice that the maximum ratio combining (MRC) technique considers $\mathbf{\Delta}_{ZF} \triangleq \left(\mathbf{H}^H\mathbf{H}\right)^{-1} = \mathbf{I}$.

[2] Subscripts MRC (maximum ratio combining) and MF (matched filter) are interchangeably used through this paper.

### A. Extension to regularized ZF and MMSE

The problem formulation can be extended to consider a regularized ZF (RZF) and MMSE detection which calls for performing the following inversion

$$\mathbf{\Delta}_{RZF/MMSE} = \left(\mathbf{H}^H\mathbf{H} + \xi\,\mathbf{I}\right)^{-1} \tag{3}$$

where $\xi$ is a regularization parameter in RZF or equated to $\sigma^2$ for MMSE. The parameter, $\xi$, provides a balance between suppressing inter-cell interference when set to a lower value and maximizing the channel gain at each UT when set to a higher value. It therefore depends on SNRs, system dimensions and channel uncertainties. In regard to equation (3) we propose to re-arrange it as

$$\mathbf{\Delta}_{RZF/MMSE} = \left(\begin{bmatrix}\mathbf{H}\\ \sqrt{\xi}\mathbf{I}\end{bmatrix}^H\begin{bmatrix}\mathbf{H}\\ \sqrt{\xi}\mathbf{I}\end{bmatrix}\right)^{-1} = \left(\breve{\mathbf{H}}^H\breve{\mathbf{H}}\right)^{-1} \tag{4}$$

where $\breve{\mathbf{H}} = \begin{bmatrix}\mathbf{H}\\ \sqrt{\xi}\mathbf{I}\end{bmatrix}$. Notice that equation (4) suggests that applying ZF using $\breve{\mathbf{H}}$ is equivalent to RZF or MMSE where the computational complexity increases very slightly as multiplications by zeros can be skipped.

### B. Low complexity implementation techniques

There have been many contributions that addressed low complexity implementation techniques for uplink massive MIMO detection techniques involving explicit and non-explicit matrix inversion techniques. Namely, Neumann series expansion (NSE) [7], Gauss Seidel (GS) [9] and recursive Gram matrix inversion update (RGMIU) [4] are among the well documented techniques. Among all these techniques, GS stood out as the one with the lowest computational burden. However, two issues are worth noting; first, the processing shall be centralized which entails putting stringent requirements on the interconnects' bandwidth between the radio heads and the central processing unit. Second, even if a given detection method exhibits a lower computational complexity (usually measured in terms of the number of operations-multiplications/additions), it turns out that the data flow and the algorithm regularities dictate the overall latency in completing the required processing. One major part of the latency pertains to the calculation of the Gram matrix. An efficient FPGA based implementation suggests the use of a systolic array wherein the latency is linear in the number of antennas $M$. One would therefore expect that an efficient decentralized processing is devised to alleviate these burdens, i.e. relax the requirements on the interconnects' bandwidth and also reduce latency through concurrent parallel processing.

## III. Decentralized processing

The detection problem can be reformulated as a distributed processing based on ADMM optimization framework [12]. The ADMM framework has been adopted in [11] and has shown to

exhibit higher complexity compared to conjugate gradient (CG) based method. In this section we propose an efficient distributed processing based on embedding one single computationally efficient GS iteration within the ADMM iterative steps. Towards that end, we suggest to partition the received signal into groups of *C* clusters. Instead of processing *M* received signals at a time, one would consider *C* groups with *M*/*C* received signals each. Therefore, the received signal per cluster *c* can be written as

$$\mathbf{y}_c = \mathbf{H}_c\mathbf{s} + \mathbf{n}_c \quad c = 1,\ldots,C \tag{5}$$

where $\mathbf{y}_c \in \mathbb{C}^{M/C\times 1}$, $\mathbf{H}_c = \begin{bmatrix}\mathbf{h}_{c,1} & \mathbf{h}_{c,2} & \cdots & \mathbf{h}_{c,K}\end{bmatrix} \in \mathbb{C}^{M/C\times K}$ and $\mathbf{s} = \begin{bmatrix}s_1 & s_2 & \cdots & s_K\end{bmatrix}^T$. $\mathbf{n}_c \in \mathbb{C}^{M/C\times 1}$ represents the additive receiver noise vector.

The ADMM framework [12] introduces one auxiliary variable $\mathbf{r}_c = \mathbf{s}$ per cluster and solves the following optimization problem (with $f(\mathbf{s})$ being a regularization function [12])

$$\begin{cases}\hat{\mathbf{s}} = \arg\min\limits_{\mathbf{s}\in\mathbb{C}^{Kx1}}\left(f(\mathbf{s}) + \frac{1}{2}\sum_{c=1}^{C}\left\|\mathbf{y}_c - \mathbf{H}_c\mathbf{r}_c\right\|^2\right) \\ \text{such that } \mathbf{r}_c - \mathbf{s} = \mathbf{0}, \quad c = 1,\ldots,C\end{cases} \tag{6}$$

The solution to the consensus problem (6) is carried out using the following iterative steps over $l = 1,\ldots,L$, by introducing one Lagrangian multiplier $\boldsymbol{\lambda}_c \in \mathbb{C}^{Kx1}$ per cluster c:

Step 1: Processing at every cluster:

$$\mathbf{r}_c^{(l+1)} = \arg\min_{\mathbf{r}_c\in\mathbb{C}^{Kx1}}\left(\frac{1}{2}\left\|\mathbf{y}_c - \mathbf{H}_c\mathbf{r}_c\right\|^2 + \frac{\beta}{2}\left\|\mathbf{s}^{(l)} - \left(\mathbf{r}_c + \boldsymbol{\lambda}_c^{(l)}\right)\right\|^2\right) \tag{7}$$

Step 2: Processing at the central processing unit:

$$\mathbf{s}^{(l+1)} = \arg\min_{\mathbf{s}\in\mathbb{C}^{Kx1}}\left(f(\mathbf{s}) + \frac{1}{2}\sum_{c=1}^{C}\left\|\mathbf{s} - \left(\mathbf{r}_c^{(l+1)} + \boldsymbol{\lambda}_c^{(l)}\right)\right\|^2\right) \tag{8}$$

Step 3: Langragian multiplier update at every cluster:

$$\boldsymbol{\lambda}_c^{(l+1)} = \boldsymbol{\lambda}_c^{(l)} + \alpha\left(\mathbf{r}_c^{(l+1)} - \mathbf{s}^{(l+1)}\right) \tag{9}$$

The penalty parameter $\beta$ is a positive real valued number set to control the convergence behavior while $\alpha = 1$ to guarantee the convergence. The convergence of the ADMM is beyond the scope of this paper but we refer the reader to [12] for a detailed discussion. For the sake of simplicity we assume that $K < \frac{M}{C}$, i.e. the number of antennas per cluster is higher than the number of served users.

It can be shown that step 1 resumes to solving the following system of linear equation

$$\left(\mathbf{H}_c^H\mathbf{H}_c + \beta\mathbf{I}_K\right)\mathbf{r}_c^{(l+1)} = \mathbf{H}_c^H\mathbf{y}_c + \beta\left(\mathbf{s}^{(l)} + \boldsymbol{\lambda}_c^{(l)}\right) \tag{10}$$

Since $\mathbf{H}_c^H\mathbf{H}_c + \beta\mathbf{I}_K$ is Hermitian positive definite, it can be decomposed as $\mathbf{H}_c^H\mathbf{H}_c + \beta\mathbf{I}_K = \mathbf{D}_c + \mathbf{L}_c + \mathbf{L}_c^H$ where $\mathbf{D}_c$, $\mathbf{L}_c$ and $\mathbf{L}_c^H$ are the diagonal, lower triangular and upper triangular parts [9]. The local estimate of the auxiliary variable can be iteratively computed as

$$\mathbf{r}_c^{(l+1,i)} = \left(\mathbf{D}_c + \mathbf{L}_c\right)^{-1}\left(\mathbf{g}_c - \mathbf{L}_c^H\mathbf{r}_c^{(l+1,i-1)}\right), \quad i = 1,\ldots,i_{MAX} \tag{11}$$

Note that equation (11) introduces iterative steps over $i = 1,\ldots,i_{MAX}$ within the ADMM outer steps (over $l = 1,\ldots,L$) where $\mathbf{g}_c = \mathbf{H}_c^H\mathbf{y}_c + \beta\left(\mathbf{s}^{(l)} + \boldsymbol{\lambda}_c^{(l)}\right)$.

However, we propose to run one single iteration of (11) at every ADMM pass by substituting the local auxiliary variable $\mathbf{r}_c^{(l+1,i-1)}$ by the consensus variable $\mathbf{s}^{(l)}$ to get

$$\mathbf{r}_c^{(l+1)} = \left(\mathbf{D}_c + \mathbf{L}_c\right)^{-1}\left(\mathbf{H}_c^H\mathbf{y}_c + \left(\beta\mathbf{I}_K - \mathbf{L}^H\right)\mathbf{s}^{(l)} + \beta\boldsymbol{\lambda}_c^{(l)}\right) \tag{12}$$

It shall be noted that the inverse of $\mathbf{D}_c + \mathbf{L}_c$ is not explicitly performed as the matrix is a lower triangular matrix which can be solved within *K* steps. Therefore, the overall computational complexity is on the order of $\mathrm{O}\left(K^2\right)$ per ADMM pass.

With ZF based detection step 2 reduces to

$$\mathbf{s}^{(l+1)} = \frac{1}{C}\sum_{c=1}^{C}\left(\mathbf{r}_c^{(l+1)} + \boldsymbol{\lambda}_c^{(l)}\right) \tag{13}$$

which is a consensus average of the sum of the auxiliary variables and the Lagrangian multipliers that are updated in the third step.

The proposed computationally efficient algorithm, in table 1, can be viewed as an ADMM based decentralized method [11] with an embedded single pass GS step. If the penalty parameter $\beta = 0$ and the Lagrangian multipliers $\boldsymbol{\lambda}_c^{(l)} = \mathbf{0}$ on every ADMM step, the algorithm boils down to a distributed GS technique where a consensus average $\mathbf{s}^{(l)}$ is broadcasted to every cluster. The proposed distributed GS technique is performed using the steps below

Step 1: Processing at every cluster:

$$\mathbf{r}_c^{(l+1)} = \left(\mathbf{D}_c + \mathbf{L}_c\right)^{-1}\left(\mathbf{H}_c^H\mathbf{y}_c - \mathbf{L}_c^H\mathbf{s}^{(l)}\right) \tag{14}$$

Step 2: Processing at the central processing unit:

$$\mathbf{s}^{(l+1)} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{r}_c^{(l+1)} \tag{15}$$

To kick off the algorithm these initialization settings are performed: $\mathbf{r}_c^{(1)} = \mathbf{D}_c^{-1}\mathbf{y}_c^{MF}$, $\boldsymbol{\lambda}_c^{(1)} = \mathbf{0}$ and $\mathbf{s}^{(1)} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{r}_c^{(1)}$ where $\mathbf{y}_c^{MF} = \mathbf{H}_c^H\mathbf{y}_c$.

Note that the algorithm does not require a prior matrix inversion which renders it suitable for low channel coherence time. The most time consuming operation, in the event that new channel estimates are available, is computing the Gram matrix $\mathbf{H}_c^H\mathbf{H}_c$ per cluster. Compared to the centralized processing, this represents a substantial reduction of the latency by a factor of *C*. All steps in Table 1 are performed in parallel in every cluster but step 3.4 which is performed at the central processing unit, which in turn will broadcast the consensus estimate back to every cluster.

## IV. Performance analysis

### *A. TDD-OFDM modulation scheme and frame structure*

A LTE like 10 msec TDD-OFDM frame structure is used [4]. The frame structure is divided into 10 sub-frames. Sub-frame 0

TABLE 1. ALGORITHM FOR DISTRIBUTED ADMM-GS BASED DETECTION

1 **Input:** $\mathbf{y}_c$ $\mathbf{H}_c$ for $c = 1,\ldots,C$, $\beta$ and $\alpha$

2 **Preprocessing per cluster:**

2.1 Matched filtering: $\mathbf{y}_c^{MF} = \mathbf{H}_c^H \mathbf{y}_c$

2.2 Decomposition: $\mathbf{H}_c^H \mathbf{H}_c + \beta \mathbf{I}_K = \mathbf{D}_c + \mathbf{L}_c + \mathbf{L}_c^H$

3 **ADMM-GS iterations:**

**Initialization:** $\mathbf{r}_c^{(1)} = \mathbf{D}_c^{-1}\mathbf{y}_c^{MF}$, $\boldsymbol{\lambda}_c^{(1)} = \mathbf{0}$ and $\mathbf{s}^{(1)} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{r}_c^{(1)}$

3.1 **for** $l = 2,\ldots,L$ do

3.2 $\mathbf{r}_c^{(l)} = \left(\mathbf{D}_c + \mathbf{L}_c\right)^{-1}\left(\mathbf{y}_c^{MF} + \left(\beta \mathbf{I}_K - \mathbf{L}^H\right)\mathbf{s}^{(l-1)} + \beta \boldsymbol{\lambda}_c^{(l-1)}\right)$

3.3 $\mathbf{b}_c^{(l)} = \mathbf{r}_c^{(l)} + \boldsymbol{\lambda}_c^{(l-1)}$

3.4 $\mathbf{s}^{(l)} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{b}_c^{(l)}$

3.5 $\boldsymbol{\lambda}_c^{(l)} = \boldsymbol{\lambda}_c^{(l-1)} + \alpha\left(\mathbf{r}_c^{(l)} - \mathbf{s}^{(l)}\right)$

3.6 **end for**

4 **Output:** $\hat{\mathbf{s}} = \mathbf{s}^{(L)}$

is reserved for downlink control and synchronization while each subsequent sub-frame consists of two time slots. The first time slot is dedicated to uplink (UL) pilot and data transmissions whereas the second time slot is used for downlink (DL) pilot and data transmissions. The UL-DL and DL-UL switching interval is 75 $\mu$sec. As such, it is apparent that the waveform is suitable for channel coherence time higher than 1 msec. The TDD-OFDM waveform parameters are outlined in Table 2.

TABLE 2. TDD-OFDM WAVEFORM PARAMETERS

| Parameter | Value |
|---|---|
| Sampling rate $f_s$ | 30.72MHz |
| FFT/IFFT size $N_{FFT}$ | 2048 bins |
| Occupied and useful bins | 1200 bins |
| Subcarrier spacing $f_0$ | 15 kHz |
| OFDM symbol CP | 1/16 of OFDM symbol |
| Total OFDM symbol duration (including CP) | 70.833 $\mu$sec |
| UL-DL and DL-UL switching guard | 75.00 $\mu$sec |

## B. *Simulation results*

This subsection discusses the simulation results of the proposed technique wherein the centralized ZF (with direct matrix inversion) and GS based detection techniques are used as reference methods. Figure 1 depicts the post detection average RMS error vector magnitude (EVM) versus the number of users at 8dB and 10dB SNR. The number of iterations is shown within the parenthesis in the figure's legend. The centralised GS and the proposed distributed ADMM-GS are implemented with three and five iterations respectively. The choice of these settings are made to roughly balance the overall latency. The

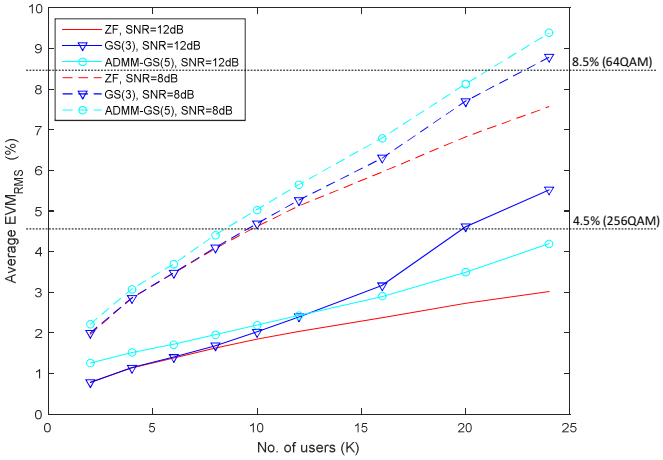


Figure 1. Average RMS EVM (%) versus the number of users at 8 dB and 12 dB SNR.

total number of antennas is set to 128 elements while the number of clusters is fixed to 4 so that every cluster has 32 antennas. A detailed analysis on the performance and complexity trade-offs as a function of the number of iteration and cluster partitioning shall be performed in a separate future contribution.

The simulation results show that, at high SNR and interference dominant regime, the proposed ADMM-GS outperforms the GS techniques. This is mainly due the fact that GS requires more extra iterations at the expense of higher latency. However, this trend is not maintained at lower SNR where the ADMM-GS performance can be improved by fine turning the penalty parameter. At high SNR (12dB), the proposed ADMM-GS can support up to 25 UTs with 256QAM modulation per subcarrier compared to only 20 UTs using centralized GS. This translate to a per subcarrier sum rate of 200 bit/s/Subcarrier and 160 bit/s/Subcarrier respectively. Therefore a maximum UL throughput of 1.2Gbit/s[3] is expected, using ADMM-GS based distributed processing.

## C. *Sensitivity to channel estimation error*

It has been argued that the channel estimation error effect tend to vanish as the number of antennas gets higher. One would therefore question if such benefit is lost with the distributed processing since the BS is divided into clusters with fewer antennas. Tables 3 and 4 below show the relative performance loss at different channel estimation error levels. The number of users is set to 16 and SNR=12dB. The channel estimation errors are modeled as in [17] wherein the magnitude and phase errors are white Gaussian noise with zero mean and variances of $\sigma_a^2$ and $\sigma_\theta^2$ in dBs and degrees, respectively.

The authors in [17] have shown that ZF is very sensitive to channel estimation errors compared to the maximum radio combining. Our results reveal an interesting findings on the effect of such errors on the performance of the low complexity

[3] Based on 1200 subcarrier per OFDM symbol, 5 OFDM data symbols reserved for UL data within 1msec sub-frame duration. The number shall be scaled with 9/10 to account for the overhead of using sub-frame 0 for dedicated downlink synchronization and control.

TABLE 3. PERFORMANCE LOSS IN % DUE TO AMPLITUDE MISMATCH $\sigma_a^2$

| $\sigma_a^2$ dB | ZF | ADMM-GS(5) | GS(3) |
|---|---|---|---|
| 1 | 9.77 | 8.48 | 5.65 |
| 3 | 23.21 | 20.46 | 13.69 |
| 5 | 52.24 | 46.49 | 32.02 |

TABLE 4. PERFORMANCE LOSS IN % DUE TO AMPLITUDE MISMATCH $\sigma_\theta^2$

| $\sigma_\theta^2$ deg. | ZF | ADMM-GS(5) | GS(3) |
|---|---|---|---|
| 1 | 10.02 | 8.63 | 5.78 |
| 2 | 20.59 | 17.86 | 12.05 |
| 3 | 36.43 | 31.82 | 21.80 |
| 5 | 78.11 | 69.07 | 48.95 |

approximation methods where the centralized GS is relatively more robust to such errors. The proposed ADMM-GS sits in between. We attribute this to two facts; (i) the ADMM-GS inherits the robustness of GS to channel estimation errors, however, (ii) it loses the benefits introduced by the 'large scale' as the BS is divided into clusters of smaller antenna sizes. Nevertheless, an in-depth analysis is required to investigate the effect of channel estimation errors (as well as hardware impairments) on distributed large scale MIMO detection. Due to limited space, this is left for future work as well.

## V. CONCLUSION

Being a disruptive 5G technology, large scale MIMO has shown to provide substantial improvement in spectral and energy efficiencies at a cost of implementing computationally prohibitive algorithms. Even the centralized low complexity linear detection methods put a lot of stress on the interconnects' bandwidth to/from the central processing unit. This issue prevents system's scalability to encompass hundreds or thousands of antenna elements. Decentralized processing has recently been proposed to address these bottlenecks by partitioning the large base station antenna array into clusters, wherein each group comprises of small number of independent radio chains. Adopting similar architecture, we have proposed an efficient ADMM-GS algorithm which is based on embedding one single GS iteration within the outer ADMM iterations. Doing so, the overall processing exhibits lower computational complexity *without an explicit matrix inversion* (even during the initialization phase) per cluster. Being a distributed processing and inheriting the fast convergence characteristics of the GS, ADMM-GS is expected (in-depth analysis is required in future works) to improve the processing latency. Only preliminary results have been publish in this paper. Future contributions includes a detailed analysis on the performance and complexity trade-offs, as a function of the number of iteration and cluster partitioning parameters, shall be performed. An in-depth analysis is required to investigate the effect of channel estimation errors (as well as hardware impairments) on distributed large scale MIMO detection.

## REFERENCES


[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74 – 80, Feb. 2014.

[2] T. L. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590 – 3600, Nov. 2010.

[3] J. Hoydis, K. Hosseini, S. T. Brink and M. Debbah, "Making Smart Use of Excess Antennas: Massive MIMO, Small Cells, and TDD," Bell Labs Technical Journal, vol. 18, no. 2, Sep. 2013, pp. 5-21.

[4] M. Ahmed Ouameur, D. Massicotte and M. A. Akhtar, "Performance Evaluation and Implementation Complexity Analysis Framework for ZF Based Linear Massive MIMO Detection" submitted to IEEE Transactions on consumer electronics, July 2018.

[5] H. Q. Ngo, *Massive MIMO: Fundamentals and System Designs*, PhD. Thesis, Linköping University Electronic Press, 2015.

[6] A. Mueller, A. Kammoun, E. Björnson, M. Debbah, "Linear Precoding Based on Polynomial Expansion: Reducing Complexity in Massive MIMO," *EURASIP Journal on Wireless Communications and Networking*, pp. 1-22, July 2016, doi: 10.1186/s13638-016-0546-z.

[7] 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 J. Sel. Topics in Sig. Proc., vol. 8, no. 5, pp.916–929, Oct. 2014.

[8] X. Gao, L. Dai, J. Zhang, S. Han, and Chih-Lin I, "Capacity-Approaching Linear Precoding with Low-Complexity for Large-Scale MIMO Systems," *IEEE International Conference on Communications*, London, UK, pp. 1577 – 1582, 8-12 June 2015.

[9] Z. Wu C. Zhang Y. Xue S. Xu Z. You "Efficient Architecture for Soft-Output Massive MIMO Detection with Gauss-Seidel Method," IEEE International Conference on Circuits and Systems, Montreal, May 2016, pp. 1886-1889.

[10] M. Wu , C. Dick , J.-R. Cavallaro and C. Studer, "High-Throughput Data Detection for Massive MU-MIMO-OFDM Using Coordinate Descent," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2357 – 2367, Dec. 2016.

[11] K. Li, R. R. Sharan, Y. Chen, T. Goldstein, J. R. Cavallaro and C. Studer, "Decentralized Baseband Processing for Massive MU-MIMO Systems," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 491-507, Dec. 2017.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, Jan. 2011.

[13] C. Jeon, K. Li, J. R. Cavallaro, C. Studer, , "Decentralized Equalization with Feedforward Architectures for Massive MU-MIMO," available at: https://arxiv.org/abs/1808.04473

[14] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H. P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," IEEE Commun. Mag., vol. 49, no. 2, pp. 102–111, Feb. 2011.

[15] L. Liu et al., "Distributed and Centralized Baseband Processing Algorithms, Architectures, and Platforms," *Massive MIMO for Efficient Transmission (MAMMOET) Project Report*, Reference Number ICT-619086-D3.2, January 2016, 87 pages.

[16] M. Peng, Y. Li, Z. Zhao, and C. Wang, "System architecture and key technologies for 5G heterogeneous cloud radio access networks," IEEE Netw., vol. 29, no. 2, pp. 6–14, Mar. 2015

[17] F. Athley, G. Durisi and U. Gustavsson, "Analysis of Massive MIMO with hardware impairments and different channel models," *2015 9th European Conference on Antennas and Propagation (EuCAP)*, Lisbon, 2015, pp. 1-5