Semi-blind channel estimation based on modified CMA and unitary scrambling for massive MIMO systems

Noura Sellami · Mohamed Siala

Accepted: 21 October 2021 / Published online: 22 November 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
Pilot contamination is one of the main impairments in multi-cell massive Multiple-Input Multiple-Output systems. In order to improve the channel estimation in this context, we propose to use a semi-blind channel estimator based on the constant modulus algorithm (CMA). We consider an enhanced version of the CMA namely the Modified CMA which modifies the cost function of the CMA algorithm to the sum of cost functions for real and imaginary parts. Due to pilot contamination, the channel estimator may estimate the channel of a contaminating user instead of that of the user of interest (the user for which the Base Station wants to estimate the channel and then the data). To avoid this, we propose to scramble the users sequences before transmission. We consider different methods to perform unitary scrambling based on rotating the transmitted symbols (one Dimensional (1-D) scrambling) and using unitary matrices (two-Dimensional (2-D) scrambling). At the base station, the received sequence of the user of interest is descrambled leading to a better convergence of the channel estimator. We also consider the case where the Automatic Repeat reQuest protocol is used. In this case, using scrambling leads to a significant gain in terms of Block Error Rate due to the change of the contaminating users data from one transmission to another induced by scrambling.

Keywords Massive MIMO · Pilot contamination · Constant modulus algorithm (CMA) · Modified CMA (MCMA) · Unitary scrambling · ARQ

1 Introduction
Massive Multiple-Input Multiple-Output (MIMO) is one of the most powerful technologies proposed for 5G and beyond wireless communication systems since it offers many advantages such as throughput improvement and energy efficiency. It is based on using a number of antennas in the base station (BS) which is much larger than the number of served users [1–3].

In order to exploit all of the benefits offered by a massive MIMO system, accurate channel state information (CSI) is necessary. In systems using Time Division Duplexing (TDD) protocol, it is obtained during the uplink transmission, thanks to the channel reciprocity. Thus, all users in all cells send their uplink training sequences synchronously, to the base stations which estimate the uplink channels. However, the number of training sequences is limited since their duration can not exceed the coherence interval of the channels. Hence, users in different cells are forced to use the same training sequences which leads to pilot contamination [4].

Pilot contamination is one of the main impairments in massive MIMO systems. Many works in the literature have studied this problem (see [1,5,6] and references therein). In [7–9], semi-blind algorithms have been proposed to perform pilot decontamination by using the stochastic Constant Modulus Algorithm (CMA) [10,11]. Indeed, the CMA is one of the most famous blind equalization algorithms thanks to its simplicity and flexibility. However, the CMA suffers from an arbitrary phase ambiguity. It also leads to degraded performance for Quadrature Amplitude Modulation (QAM) constellations since its cost function takes into account only the amplitude of the equalizer output. In this context, the Modified CMA (MCMA) has been proposed in [12]. It modifies the cost function of the CMA algorithm to the sum of
cost functions for real and imaginary parts which is more suitable for QAM signals and simplifies the phase ambiguity correction. This algorithm has been used in many works in the literature and was also referred to as Multi-Modulus Algorithm [13–15]. In [16], authors proposed a blind equalizer based on MCMA for MIMO systems. In [18], a semi-blind uplink interference suppression scheme based on CMA and using decision feedback channel estimation has been proposed in massive MIMO systems. This work has been extended in [17] to the case of Multi-Modulus (QAM) signals by using the stochastic MCMA. In [18], a semi-blind receiver using an hybrid cost function based on the Multi-Modulus criterion for the unknown data and the Least Squares criterion for the pilot data has been proposed for pilot contamination mitigation in massive MIMO systems. This work has been extended to the case of massive MIMO-OFDM systems in [19]. In [18,19], the batch processing based on the use of block iterative implementation [20] is used instead of initially proposed stochastic algorithms [10–12]. Notice that batch processing leads to better performance at the cost of additional complexity computation.

In this paper, we propose a semi-blind algorithm based on the MCMA [12] to estimate the channels and recover the users transmitted data sequences. In our presentation we use the stochastic MCMA for sake of clarity and since it achieves a good complexity/performance tradeoff. We propose to improve its performance by scrambling users sequences before transmission and descrambling the user of interest data sequence at the BS. This method will improve the convergence of the channel estimator and help it to focus on the user of interest. We propose in this paper to perform unitary scrambling by rotating the transmitted data (1-D scrambling) or by using unitary matrices (2-D scrambling). We will show that our proposed scrambling methods can also be efficiently used in conjunction with other versions of CMA as the block MCMA [18].

Moreover, we will consider the Automatic Repeat reQuest (ARQ) protocol. When the BS fails to estimate a user block of data, this user retransmits it using a different scrambling rotating sequence or unitary matrix. This leads to a gain in terms of diversity mainly when the contaminating users and the contaminated one retransmit their blocks at the same time.

The paper is organized as follows. In Sect. 2, we describe the system model and present the scrambling methods based on 1-D and 2-D unitary transformations. Section 3 explains the descrambling techniques. Section 4 details the proposed semi-blind receiver based on the CMA then on the MCMA using descrambled data. In Sect. 5, we study the complexities of the proposed methods. In Sect. 6, we describe the proposed ARQ protocol in conjunction with MCMA and scrambling. In Sect. 7, we give simulation results.

Throughout this paper, scalars are lower or upper case. Vectors and matrices are bold, lower and upper case respectively. Moreover, $(\cdot)^*, (\cdot)^T$ and $(\cdot)^H$ denote respectively conjugation, transposition and trans-conjugation, $E(\cdot)$ denotes the expected value operator and $I_K$ is the $K \times K$ identity matrix.

2 System model

2.1 General framework

We consider a multi-cell massive MIMO system containing $L$ cells. The Base Station (BS) of each cell is equipped with $M$ antennas and serves $K$ single antenna users, with $M \gg K$. We assume that TDD mode is used. We consider that the channels are narrowband as frequency-selective wideband channels can be converted into multiple parallel flat fading channels by using Orthogonal Frequency Division Multiplexing (OFDM). The input information bit sequences are mapped to the symbol alphabet of a Quadrature Amplitude Modulation (QAM) using the Gray mapping. We assume that the transmissions are organized into bursts of $N$ symbols each. The channels are supposed to be invariant during one burst and can change from burst to burst. In a conventional massive MIMO system, the signal sampled at symbol rate received at the $m$th antenna in the $l$th cell at time $n$ is given by

$$r_{l,m}(n) = \sum_{k=1}^{K} h_{l,i,m,k} d_{l,k}(n) + \sum_{i=1,i\neq l}^{L} \sum_{k=1}^{K} h_{l,i,m,k} d_{i,k}(n) + w_{l,m}(n)$$  \hspace{1cm} (1)

where $d_{l,k}(n)$ is the transmitted modulated QAM symbol of the $k$th user in cell $l$ at time $n$ and $w_{l,m}(n)$ are modeled as independent samples of a zero mean white complex Gaussian noise with variance $\sigma^2 = \mathcal{N}_0$. The coefficient $h_{l,i,m,k}$ is the channel gain between the $k$th user in cell $i$ and the $m$th antenna in cell $l$ and is given by

$$h_{l,i,m,k} = g_{l,i,m,k} \sqrt{\beta_{l,i,k}}$$  \hspace{1cm} (2)

where $g_{l,i,m,k}$ is the small-scale fading coefficient assumed to be complex Gaussian with zero mean and unit variance, and $\beta_{l,i,k}$ is the large-scale fading coefficient which changes slowly and is independent of the antenna index $m$.

In this paper, we assume that $E\{h_{l,i,m,k} h_{l',i',m',k'}\} = 0$ for $l \neq l'$, or $i \neq i'$, or $m \neq m'$ or $k \neq k'$, and the channels coefficients are independent with the noise. The received vector at the $M$ receiving antennas of cell $l$ at time $n$ is given by
Table 1 Parameters definitions

| Parameter | Definition |
|-----------|------------|
| $L$       | Number of cells |
| $M$       | Number of antennas at each BS |
| $K$       | Number of users per cell |
| $N$       | Number of symbols per burst |
| $\tau$   | Pilot sequence length |
| $\sigma^2 = N_0$ | Variance of the complex Gaussian noise |

$r_i(n) = \sum_{k=1}^{K} h_{i,k} d_{i,k}(n) + \sum_{i=1, i \neq l}^{L} \sum_{k=1}^{K} h_{i,k} d_{i,k}(n) + w_i(n)$  \hspace{1cm} (3)

where $r_i(n) = (r_{i,1}(n), r_{i,2}(n), \ldots, r_{i,M}(n))^T$, $h_{i,k} = (h_{1,i,k}, \ldots, h_{L,i,M,k})^T$ and $w_i(n) = (w_{i,1}(n), w_{i,2}(n), \ldots, w_{i,M}(n))^T$.

If the channels are perfectly known at the BSs, in order to estimate the symbol $d_{c,u}(n)$ transmitted by user $u$ of cell $c$ at time $n$, $1 \leq u \leq K$, $1 \leq c \leq L$ and $1 \leq n \leq N$, the BS of cell $c$ processes its received signal by multiplying it by the trans-conjugation of the corresponding channel leading to

$d_{c,u}(n) = h^H_{c,u} r_c(n) \approx \|h_{c,u}\|^2 d_{c,u}(n) + h^H_{c,u} w_c(n)$.  \hspace{1cm} (4)

The approximation in (4) is obtained since $M$ is high and $h^H_{c,u} h_{i,k}$ is negligible compared to $\|h_{c,u}\|^2$, when $(c, c, u) \neq (l, i, k)$.

In this paper, we assume that the channels are not perfectly known at the BSs and must be estimated. Thus, we consider that each user has a pilot sequence of length $\tau$. Pilot symbols are transmitted at the beginning of each block.

For sake of presentation clarity, we list in Table 1 the different parameters used in this paper.

In the following, we explain the channel estimation performed during the training phase and leading to the pilot contamination phenomenon.

2.2 Training phase

We assume that all users in all cells send their uplink training sequences synchronously. These sequences are used by the BSs to estimate the uplink channels of their users. We assume that the pilot sequences of users in the same cell are orthogonal. However, all the cells reuse the same pilot sequences which leads to pilot contamination. We assume that the $k$th user in each cell is assigned the $\tau \times 1$ pilot sequence $d_k$. Let $D = (d_1, d_2, \ldots, d_K)^T$ be the $K \times \tau$ matrix containing the pilot sequences of each cell. Thus, $DD^H = \tau I_K$. The matrix of received sequences at the BS of cell $l$, $1 \leq l \leq L$, corresponding to the transmission of the training sequences is the $M \times \tau$ matrix given by

$R_l = H_{l,i} D + \sum_{i=1, i \neq l}^{L} H_{l,i} D + W_l$  \hspace{1cm} (5)

where $H_{l,i}$ is the $M \times K$ channel matrix with $h_{l,i,m,k}$ is its $(m, k)$th element for $1 \leq i \leq L$, $1 \leq k \leq K$, $1 \leq m \leq M$ and $W_l$ is the $M \times \tau$ noise matrix with $w_{l,m}(n)$ is its $(m, n)$th element for $1 \leq m \leq M$ and $1 \leq n \leq \tau$.

The channel estimation according to the Least Squares criterion is given by

$\hat{H}_{l,i} = \frac{1}{\tau} R_l D^H = H_{l,i} + \sum_{i=1, i \neq l}^{L} H_{l,i} + \frac{1}{\tau} W_l D^H$.  \hspace{1cm} (6)

The channel estimate obtained in (6) is a combination of the desired channel matrix $H_{l,i}$ and the other channel matrices $H_{l,i}$ containing the gains of the channels between the $k$th users in cells $i$, for $1 \leq i \leq L$, $i \neq l$, and the $M$ antennas of the BS of cell $l$. This phenomenon is called pilot contamination [4]. The channel estimate (6) will be used to initialize the MCMA.

In the following, we propose to scramble the users information data sequences $(d_{l,k}(\tau+1), d_{l,k}(\tau+2), \ldots, d_{l,k}(N))$ for $1 \leq k \leq K$ and $1 \leq l \leq L$ before transmission. As will be explained in the next section, scrambling will help the semi-blind channel estimator based on the MCMA to focus on the user of interest. We consider unitary scrambling since it does not change the transmitted energy.

2.3 Unitary scrambling

We propose two methods to perform unitary scrambling: 1-D scrambling and 2-D scrambling. The 1-D scrambling consists in rotating the data before transmission by using scrambling sequences. For the 2-D scrambling, we propose two approaches. The first one uses unitary matrices and performs a treatment per block of two symbols. The second one, introduces a convolutional aspect. The scrambling sequences and the unitary matrices are assumed to be known to both the transmitters and the receivers. In all cases, the scrambled data sequence transmitted by user $k$ in cell $l$ will be denoted as $s_{l,k} = (s_{l,k} (\tau+1), s_{l,k}(\tau+2), \ldots, s_{l,k}(N))^T$ for $1 \leq k \leq K$ and $1 \leq l \leq L$.

2.3.1 1-D scrambling

We propose to multiply the data symbols to be transmitted $d_{l,k}(n)$ by $s_{l,k}(n)$, the coefficients of independent random scrambling sequences randomly chosen from a set of Phase Shift Keying (PSK) symbols for $\tau+1 \leq n \leq N$, $1 \leq k \leq K$ and $1 \leq l \leq L$. The scrambled symbols are

$s_{l,k}(n) = d_{l,k}(n) s_{l,k}(n)$.  \hspace{1cm} (7)
Fig. 1 The constellation of scrambled 4-QAM data with: a 1-D scrambling using 8-PSK sequences b 1-D scrambling using 16-PSK sequences c 2-D unitary scrambling d 2-D unitary convolutional scrambling

The scrambling here consists in rotating the symbols to be transmitted. Figure 1a and b show the constellation of the transmitted symbols originally modulated by a 4-QAM modulation and then scrambled by 8-PSK scrambling sequences and 16-PSK sequences respectively.

2.4 2-D scrambling

To perform 2-D unitary scrambling, we consider the special unitary group $SU(2)$ containing $2 \times 2$ unitary matrices $U$ with $\det(U) = 1$. A matrix $U \in SU(2)$ can be written as

$$U = e^{i \phi/2} \begin{bmatrix} e^{i \phi_1 \cos(\theta)} & e^{i \phi_2 \sin(\theta)} \\ -e^{-i \phi_2 \sin(\theta)} & e^{-i \phi_1 \cos(\theta)} \end{bmatrix}$$

where $\phi, \phi_1, \phi_2$, and $\theta$ are random and take values in finite groups. We assume that when the 2-D scrambling is used, $\tau$ and $N$ are even integers. The scrambling is performed by multiplying each pair of symbols $(d_{l,k}(2n-1), d_{l,k}(2n))$ for $\tau/2 + 1 \leq n \leq N/2$, $1 \leq k \leq K$ and $1 \leq l \leq L$ by the unitary matrix $U_{l,k}^{2n}$ chosen randomly in $SU(2)$ as follows

$$(s_{l,k}(2n-1), s_{l,k}(2n)) = (d_{l,k}(2n-1), d_{l,k}(2n))U_{l,k}^{2n}$$

The scrambling here consists not only in rotating the symbols to be transmitted but also in changing their amplitudes. Figure 1c shows the constellation of the scrambled 4-QAM symbols when $\phi$ and $\theta$ are in $\{e^{i (2n+1) \pi/8}, 0 \leq n \leq 7\}$ and...
\( \phi_1 \) and \( \phi_2 \) are in \( \{ e^{(2n+1)\pi i/4}, 0 \leq n \leq 3 \} \). Notice that considering the finite group of unitary matrices \( SU(2) \) allows to code the index of the used scrambling unitary matrix by a finite number of bits. In the case of Fig. 1c, each unitary matrix can be coded by 10 bits.

In order to make the scrambling more efficient, we propose to introduce a convolutional aspect. We consider the unitary matrices \( V_{l,k}^n \) taken randomly in \( SU(2) \), for \( \tau + 1 \leq n \leq N - 1 \), \( 1 \leq k \leq K \) and \( 1 \leq l \leq L \). The user \( k \) in cell \( l \) performs successively the following calculations:

\[
\begin{align*}
(\tilde{d}_{l,k}(\tau + 2), \tilde{d}_{l,k}(\tau + 3)) &= (d_{l,k}(\tau + 1) d_{l,k}(\tau + 2)) \mathbf{V}_{l,k}^{\tau + 1} \\
(\tilde{d}_{l,k}(\tau + 3), \tilde{d}_{l,k}(\tau + 4)) &= (d_{l,k}(\tau + 2) d_{l,k}(\tau + 3)) \mathbf{V}_{l,k}^{\tau + 2} \\
& \vdots \\
(\tilde{d}_{l,k}(N - 1), \tilde{d}_{l,k}(N)) &= (d_{l,k}(N - 1), d_{l,k}(N)) \mathbf{V}_{l,k}^{N - 1}
\end{align*}
\]

The transmitted vector of scrambled data is then

\[
s_{l,k} = (\tilde{d}_{l,k}(\tau + 1), \tilde{d}_{l,k}(\tau + 2), \ldots, \tilde{d}_{l,k}(N - 1), \tilde{d}_{l,k}(N))^T.
\]

Figure 1d shows the constellation of the scrambled 4-QAM symbols when the convolutional 2-D scrambling is used and the unitary matrices are generated in the same way as for Fig. 1c. We notice that the data is more efficiently scrambled than for previous methods.

For all scrambling methods, the \( m \)th signal received by the \( l \)th antenna in the \( c \)th cell can be expressed as

\[
x_{l,m}(n) = \sum_{k=1}^{K} h_{l,m,k} s_{l,k}(n) + \sum_{i=1}^{L} \sum_{k=1}^{K} h_{l,i,m,k} s_{l,k}(n) + w_{l,m}(n)
\]

In the following, we present the descrambling techniques corresponding to the different methods of scrambling.

### 3 Descrambling

In this paper, we propose a semi-blind channel estimator which combines the training based channel estimator and the blind estimator based on the MCMA [12]. Each BS estimates the channel and data of users successively (from user 1 to user \( K \)). We assume in the following that the user of interest is user \( u \) of cell \( c \). The first task to be performed at the BS of cell \( c \) consists in descrambling the data of the user of interest.

In the following, we describe the descrambling process for the 1-D and 2-D scrambling techniques.

#### 3.1 1-D descrambling

When the 1-D scrambling is used at the transmitters, descrambling the data of user \( u \) at the BS of cell \( c \) consists in multiplying each received signal \( x_{c,m}(n) \), for \( 1 \leq m \leq M \) and \( \tau + 1 \leq n \leq N - 1 \), by \( \sigma_{c,u}^*(n) \) leading to

\[
z_{c,m,u}(n) = h_{c,c,m,u} d_{c,u}(n)
\]

\[
+ \sum_{k=1}^{K} h_{c,c,m,k} d_{c,k}(n) \sigma_{c,k}(n) \sigma_{c,u}^*(n)
\]

\[
+ \sum_{i=1}^{L} \sum_{k=1}^{K} h_{c,i,m,k} d_{i,k}(n) \sigma_{i,k}(n) \sigma_{c,u}^*(n)
\]

\[+ w_{c,m}(n) \sigma_{c,u}^*(n),
\]

#### 3.1.1 2-D Descrambling

When the 2-D scrambling is used at the transmitters, descrambling the data of user \( u \) at the BS of cell \( c \) consists in multiplying each vector of received pair of symbols \( (x_{c,m}(2n - 1), x_{c,m}(2n)) \), for \( 1 \leq m \leq M \) and \( \tau/2 + 1 \leq n \leq N/2 \), by \( (U_{c,u}^{2n})^H \) leading to

\[
z_{c,m,u}^{2n}(n) = h_{c,c,m,u}^{2n} d_{c,u}^{2n} + \sum_{k=1}^{K} h_{c,c,m,k}^{2n} d_{c,k}^{2n} U_{c,k}^{2n} (U_{c,u}^{2n})^H
\]

\[
+ \sum_{i=1}^{L} \sum_{k=1}^{K} h_{c,i,m,k}^{2n} d_{i,k}^{2n} U_{i,k}^{2n} (U_{c,u}^{2n})^H + w_{c,m}^{2n} (U_{c,u}^{2n})^H
\]

where \( z_{c,m,u}^{2n} = (z_{c,m,u}(2n - 1), z_{c,m,u}(2n)) \), \( d_{c,u}^{2n} = (d_{c,u}(2n - 1), d_{c,u}(2n)) \) and \( w_{c,m}^{2n} = (w_{c,m}(2n - 1), w_{c,m}(2n)) \).

#### 3.1.2 2-D convolutional descrambling

When the convolutional 2-D scrambling is used, the BS of cell \( c \) performs, for \( 1 \leq m \leq M \) successively the following calculations to descramble the data of user \( u \).
\[ (\tilde{z}_{c,m,u}(N-1), z_{c,m,u}(N)) = (x_{c,m}(N-1), x_{c,m}(N))(V_{u}^{N-1})^H \]
\[ (\tilde{z}_{c,m,u}(N-2), z_{c,m,u}(N-1)) = (x_{c,m}(N-2), \tilde{z}_{c,m,u}(N-1))(V_{u}^{N-2})^H \]
\[ \vdots \]
\[ (\tilde{z}_{c,m,u}(n+1), z_{c,m,u}(n+2)) = (x_{c,m}(n+1), \tilde{z}_{c,m,u}(n+2))(V_{u}^{n+1})^H \]
\[ \vdots \]
\[ (\tilde{z}_{c,m,u}(\tau + 1), z_{c,m,u}(\tau + 2)) = (x_{c,m}(\tau + 1), \tilde{z}_{c,m,u}(\tau + 2))(V_{u}^{\tau+1})^H \]
\[ (15) \]

In all cases, the sequences \( z_{c,u}(n) = (z_{c,1,u}(n), z_{c,2,u}(n), \ldots, z_{c,M,u}(n))^T \) for \( \tau + 1 \leq n \leq N \) will be provided at the input of the MCMA at the BS of cell \( c \) to estimate the channel of user \( u \). After descrambling, the user \( u \) data lie on the initial M-QAM constellation. However, the contaminating users data are still scrambled and lie on different constellations according to the considered scrambling technique. This will help the semi-blind channel estimator based on the MCMA to focus on the channel of the user of interest and not to try to estimate a channel of a contaminating user. It is mainly useful when the contamination level is high which happens when the contaminating users are on the boundary of the cell for example.

In the following, we present the semi-blind channel estimator. For the sake of comparison, we start by presenting the channel estimator using the stochastic CMA.

4 Semi-blind channel estimation

4.1 Semi-blind channel estimation using the stochastic CMA

To extract the data sequence of user \( u \) of cell \( c \), the stochastic CMA computes iteratively the \( M \times 1 \) extracting vector \( w_{c,u} \) for \( 1 \leq c \leq L \) and \( 1 \leq u \leq K \). Let

\[ y_{c,u}(n) = w_{c,u}^H n \]
\[ \text{where } z_{c,u}(n) = (z_{c,1,u}(n), z_{c,2,u}(n), \ldots, z_{c,M,u}(n))^T. \]
\[ \text{The cost function of the CMA is given by} \]
\[ J(w_{c,u}) = E([y_{c,u}(n)]^2 - R)^2 \]
\[ \text{where } R = \frac{E[|d_{c,u}(n)|^2]}{E[|d_{c,u}(n)|^2]}. \]
\[ (16) \]

The stochastic CMA is a gradient descent iterative procedure to minimize the cost given in (17). Thus, after computing the gradient and replacing the expectation by the instantaneous value, we obtain the update rules as

\[ y_{c,u}(n) = w_{c,u}^H(n)z_{c,u}(n) \]
\[ w_{c,u}(n + 1) = w_{c,u}(n) - \mu z_{c,u}(n) ([y_{c,u}(n)]^2 - R) y_{c,u}^*(n) \]
\[ (18) \]

where \( \mu \) is the step size, \( w_{c,u}(n + 1) \) and \( w_{c,u}(n) \) are respectively the updated and old value of \( w_{c,u} \).

Notice that the CMA is initialized with the estimation \( \hat{h}_{c,u} \) of the channel \( h_{c,u} = (h_{c,1,u}, \ldots, h_{c,M,u})^T \) obtained in the training phase and given by

\[ \hat{h}_{c,u} = h_{c,u} + \sum_{i=1, i \neq c}^{L} h_{i,u} + \tilde{w}_{c,u} \]
\[ (19) \]

where \( h_{c,i,u} = (h_{c,i,1,u}, \ldots, h_{c,i,M,u})^T \) and \( \tilde{w}_{c,u} \) is the \( u \)th column of \( \frac{1}{T} W_c D^H \). This estimation is a linear combination of the channel vectors of the \( u \)th users in the different cells using the same pilot sequence. Thus, without scrambling, when the contamination level is high, the CMA risks making a mistake and extracting the channel from one of the contaminating users. Notice that the 1-D scrambling will not influence the CMA performance since it does not change the amplitudes of the symbols. However, the 2-D scrambling which changes the phases as well as the amplitudes of the symbols will prevent the CMA from trying to extract the channel of a contaminating user.

Notice also that the CMA leads to a phase ambiguity \( \phi_{CMA} \in [0, 2\pi] \). Thus, assuming that the CMA tries to extract the channel of the user of interest, in case of convergence of the algorithm, it will converge to

\[ w_{c,u}(N) \approx e^{j\phi_{CMA}}h_{c,c,u}. \]
\[ (20) \]

Hence,

\[ w_{c,u}^H(N)\hat{h}_{c,c,u} \approx e^{-j\phi_{CMA}} ||h_{c,c,u}||^2 + e^{-j\phi_{CMA}} \sum_{i=1, i \neq c}^{L} h_{c,i,u}^H h_{c,i,u} + e^{-j\phi_{CMA}} \tilde{w}_{c,u}^H \hat{w}_{c,u}. \]
\[ (21) \]

Since \( M \) is high, the first term in (21) is dominant. Then, we consider the estimation of the phase ambiguity given by

\[ \hat{\phi}_{CMA} = -arg(w_{c,u}^H(N)\hat{h}_{c,c,u}) \]
\[ (22) \]

4.2 Semi-blind channel estimation using the stochastic MCMA

In order to make the phase ambiguity correction more accurate, the cost function of the CMA algorithm is modified in [12] to the sum of cost functions for real and imaginary parts leading to the Modified CMA (MCMA). Indeed, this estimator penalizes the deviation of the real and imaginary parts of
computing the gradient and replacing the expectation by the procedure to minimize the cost given in (24). Thus, after

\[ \tilde{y}_{c,u}(n) = \tilde{w}_{c,u}^H z_{c,u}(n). \]  

(23)

The cost function of the MCMA is given by

\[ J(\tilde{w}_{c,u}) = E((|\tilde{y}_{c,u}^R(n)|^2 - R^R)^2 + (|\tilde{y}_{c,u}^I(n)|^2 - R^I)^2) \]  

(24)

where \( R^R = \frac{E|d_{c,u}^R(n)|^4}{E|d_{c,u}^R(n)|^2} \), \( R^I = \frac{E|d_{c,u}^I(n)|^4}{E|d_{c,u}^I(n)|^2} \), \( d_{c,u}^R(n) \) and \( d_{c,u}^I(n) \) are respectively the real and imaginary parts of \( d_{c,u}(n) \), \( \tilde{y}_{c,u}^R(n) \) and \( \tilde{y}_{c,u}^I(n) \) are respectively the real and imaginary parts of \( \tilde{y}_{c,u}(n) \).

The stochastic MCMA is a gradient descent iterative procedure to minimize the cost given in (24). Thus, after computing the gradient and replacing the expectation by the instantaneous value, we obtain the update rules as

\[
\begin{align*}
\tilde{y}_{c,u}(n) &= \tilde{w}_{c,u}^H n z_{c,u}(n) \\
\tilde{w}_{c,u}(n + 1) &= \tilde{w}_{c,u}(n) - \mu z_{c,u}(n)(|\tilde{y}_{c,u}^R(n)|^2 - R^R)|\tilde{y}_{c,u}^R(n) \\
&+ j(|\tilde{y}_{c,u}^I(n)|^2 - R^I)|\tilde{y}_{c,u}^I(n)] \\
\end{align*}
\]

(25)

where \( \mu \) is the step size, \( \tilde{w}_{c,u}(n + 1) \) and \( \tilde{w}_{c,u}(n) \) are respectively the updated and old value of \( \tilde{w}_{c,u} \).

Notice that when the MCMA is used, even the 1-D scrambling will improve the performance since rotating data symbols changes their real and imaginary parts.

Notice also that the MCMA leads to a phase ambiguity \( \phi_{MCMA} \) in the finite set \( \{0, \pi/2, \pi, 3\pi/2\} \). Thus, assuming that the MCMA tries to estimate the user of interest channel, in case of convergence of the algorithm, it will converge to

\[ \tilde{w}_{c,u}(N) \approx e^{j\phi_{MCMA}} h_{c,c,u}. \]  

(26)

The estimation of the phase ambiguity is the phase in the set \( \{0, \pi/2, \pi, 3\pi/2\} \) which is the nearest to

\[ \phi_{MCMA} = -\arg(\tilde{w}_{c,u}^H(n) \tilde{h}_{c,c,u}) \]  

(27)

Since \( \phi_{MCMA} \) is in a finite set, the correction of the phase ambiguity for the MCMA is more accurate than for the CMA.

We give here more elements to explain why the proposed scrambling techniques allow to improve the performance of the semi-blind receiver based on the MCMA. We recall that the MCMA is initialized with the channel estimation given in (19). If the contamination is high (\( \beta_{c,i,u} \) is high when \( c \neq i \)), the contribution of the contaminating channels \( h_{c,i,u} \) in (19) is comparable with that of the user of interest channel \( h_{c,c,u} \).

Thus, the MCMA risks extracting the channel of a contaminating user instead of that of the user of interest since the data of all users lie on the initial M-QAM constellation. However, when scrambling is used, after descrambling at the BS, the user of interest data lie on the initial M-QAM constellation but the contaminating users data are still scrambled and lie on different constellations according to the considered scrambling technique. Thus, the MCMA will not extract the channel of a contaminating user since it does not allow to minimize the cost function which has been designed specially for the M-QAM data.

In the following, we study the computation complexities of our proposed algorithms.

## 5 Computation complexity

In this section, we present the computation complexity in terms of the number of complex multiplications required to process one user burst of \( N \) symbols in the scrambling and descrambling phases. When the 1-D scrambling is used, \( (N - \tau) \) complex multiplications are performed. When the 2-D scrambling is used, \( 2(N - \tau) \) complex multiplications are required. At the receiver, descrambling the data burst of one user requires the same number of multiplications as the corresponding scrambling. This increase in complexity is reasonable given the improvement in performance obtained using our proposed scrambling methods as will be seen in the simulation results section.

Notice that our proposed methods can be used with different versions of CMA and their complexity remains the same regardless of the version used. Indeed, many works have studied the CMA algorithm and proposed low complexity versions or better performance versions at the cost of higher complexity. The batch based MCMA, used in [18], approximates the gradient from the burst of received samples iteratively unlike the stochastic approach presented in this paper, which approximates it by using one sample estimate. The number of complex multiplications required in the batch based algorithms is equal to the one required in the stochastic algorithms times the number of iterations which can be quite high. In the simulation results section, we will show the improvement in terms of performance obtained by using our scrambling/descrambling methods in conjunction with the batch based MCMA.

In the following, we assume that an Automatic Repeat reQuest (ARQ) protocol is used. We explain the impact of scrambling on the ARQ protocol.

## 6 ARQ in conjunction with scrambling

When the ARQ protocol is used, each user performs packets retransmissions until error free detection or a maximum
number of transmissions $R_{\text{max}}$ is reached. When scrambling is used, the scrambling sequences or the unitary matrices are changed at each retransmission. We consider two scenarios:

- Scenario 1: the channels do not change from burst to burst during the retransmissions. In this case, we propose to average the sequences received during the transmissions of the same burst. At each transmission, the MCMA is initialized with the channel estimation obtained at the previous one since the channel did not change.

- Scenario 2: the channels change from burst to burst during the retransmissions. The received sequences corresponding to the retransmissions of the same burst of the user of interest are placed side by side. In this case, the system is equivalent to a system having $rM$ antennas where $r$ is the transmission number. Thus, at the $r$th transmission, the length of the channel vector to be estimated is $rM$. Indeed, a new channel vector of size $M \times 1$ has to be estimated. The $(r-1)M \times 1$ channel vector estimate already obtained by the MCMA will be used to initialize the algorithm at the $r$th transmission. The initial estimate of the new channel is obtained thanks to pilot symbols as explained in Sect. 2.2.

Scrambling leads to a decrease of the inter-cell interference due to pilot contamination mainly in the case where the contaminating users also have to retransmit their data while the user of interest transmits his own. In this case, we detail in the following the benefits of scrambling.

In scenario 1:

- without scrambling: the transmitted data of contaminating users do not change as well as their channels. The only gain obtained by using ARQ is due to noise reduction.
- with scrambling: thanks to scrambling, everything happens as if the transmitted data of contaminating users change during the transmissions which leads to a performance improvement since it decreases inter-cell interference due to pilot contamination.

In scenario 2:

- without scrambling: at transmission $r$, the system is equivalent to a system having $rM$ antennas. The channels of the contaminating users change but their data do not change from a transmission to another.
- with scrambling: at transmission $r$, the system is also equivalent to a system having $rM$ antennas. The channels of the contaminating users change as well as their data from a transmission to another thanks to scrambling. Everything happens as if the contaminating users have changed.

7 Simulation results

We consider a multi-user massive MIMO system containing $L = 3$ cells. The BS of each cell is equipped with $M = 60$ antennas and serves $K = 4$ single antenna users. The pilot sequence of each user has the length of $r = 8$ symbols. The burst length is set to $N = 300$ symbols. The transmitted symbols are modulated by a 4-QAM modulation. The step size is set to $\mu = 0.005$. We assume as in [6,7], that the large-scale coefficients for all users in all cells are set to $\beta_{l,j,k} = 1$ and $\theta_{l,j,k} = \beta$ where $\beta$ is a constant, for $1 \leq l \leq L, 1 \leq i \leq L, 1 \leq k \leq K$ and $l \neq i$. In order to generate the unitary matrices in $SU(2)$, we consider as for Fig. 1 that $\phi_1, \phi_2$, and $\theta$ are random such as $\phi$ and $\theta$ are in $\{e^{(2n+1)/4}, 0 \leq n \leq 3\}$. Unless otherwise specified, we use the stochastic CMA and the stochastic MCMA in our simulations.

Figure 2 shows the Symbol Error Rate (SER) versus $\beta$ when the CMA or the MCMA is used. The solid curves (respectively dotted curves) show the SER obtained without scrambling, with 1-D scrambling using 8-PSK scrambling sequences, with 2-D scrambling using 16-PSK scrambling sequences, with 2-D convolutional scrambling when the MCMA (respectively CMA) is used. We assume that $E_b/N_0 = -2dB$, $E_b$ being the average energy per information bit. Figure 2 confirms that the 1-D scrambling does not improve the performance obtained when the CMA is used. However, the 1-D scrambling leads to performance improvement when the MCMA is used. We notice that using 16-PSK scrambling sequences does not lead to any improvement compared to using 8-PSK ones. The gain increases when the 2-D scrambling is considered. More gain is obtained with the 2-D convolutional scrambling. We also notice that the improvement is important specially when $\beta \geq 0.6$. Indeed, scrambling does not improve significantly the performance when the contamination is very little since in this case the channel estimator is rarely likely to try to estimate a contaminating channel instead of the channel of interest. We also notice that the improvement obtained by using the scrambling is more significant when the MCMA is used.

Figure 3 shows the SER versus $E_b/N_0$, for the receiver using the CMA (dotted curves) and the MCMA (solid curves) without scrambling, with 1-D scrambling using the 8-PSK sequences, with 2-D scrambling and with 2-D convolutional scrambling when $\beta = 0.6$. We notice that a significant gain in terms of SER is obtained when scrambling is used. The gain increases with $E_b/N_0$.

Figure 4 shows the SER versus $E_b/N_0$, for the receiver using the stochastic MCMA with 2-D convolutional scrambling (solid curve) and the block MCMA (dotted curves) without scrambling and with 2-D convolutional scrambling.
when $\beta = 0.6$. Obviously, the block MCMA leads to better performance than the stochastic MCMA but at the cost of higher complexity (as explained in Sect. 5). We also notice that the use of our proposed scrambling method in conjunction with the block MCMA leads to further improved performance and even allows eliminating the error floor obtained due to pilot contamination.

Figures 5 and 6 show the BLER Error Rate (BLER) versus $E_b/N_0$ for $\beta = 0.7$, for the MCMA algorithm without scrambling and the MCMA with 1-D scrambling using 8-PSK sequences when the ARQ is used and $R_{\text{max}} = 2$. In Fig. 5, the channels do not change during the transmissions of the same burst (scenario 1 of Sect. 6). In Fig. 6, channels change independently from burst to burst (scenario 2 of Sect. 6). Solid curves and dotted ones show the BLER at the first and the second transmission respectively. Figures 5 and 6 show that the gain in terms of BLER obtained at the second transmission is more important when scrambling is used. We also notice that the BLER obtained at the 2nd transmission in scenario 2 is worse than that obtained in scenario 1 at low $E_b/N_0$. This is due to the fact that, in case of retransmission of the same block due to detection failure, the channel to be estimated is longer in scenario 2 than in the other scenario and the estimation is not reliable enough at low $E_b/N_0$. At higher $E_b/N_0$ values, the BLER is better in scenario 2 thanks to the diversity gain.
since the system is equivalent, in case of retransmission of the same block, to a system with $2M$ receive antennas.

8 Conclusion

In this paper, we proposed a semi-blind channel estimator based on the modified CMA (MCMA) for multi-cell massive MIMO systems. Since the initial estimation suffers from pilot contamination and in order to improve the performance of the channel estimator, we proposed to scramble the users data before transmission then descramble the data of the user of interest before performing the MCMA. This technique prevented the MCMA from estimating the channel of a contaminating user instead of that of the user of interest. We also considered the case where the ARQ protocol is used. We considered 2 scenarios depending on whether the channel changes from one transmission to another or not. Simulation results showed that the use of scrambling leads to a significant BLER improvement at the second transmission.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Lu, J. L., Li, G. Y., Swindlehurst, A. L., Ashikhmin, A., & Zhang, R. (2014). An overview of massive MIMO: Benefits and challenges. IEEE Journal of Selected Topics in Signal Processing, 8(5), 742–758.

2. Larsson, E. G., Edfors, O., Tufvesson, F., & Marzetta, T. L. (2014). Massive MIMO for next generation wireless systems. IEEE Communications Magazine, 52(2), 186–195.

3. Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C. K., & Zhang, J. C. (2014). What will 5G be? IEEE Journal on Selected Areas in Communications, 32(6), 1065–1082.

4. Marzetta, T. L. (2010). Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Transactions on Wireless Communications, 9(11), 3590–3600.

5. Leow, C. Y., Rahman, T. A., Nunoo, S., & Iliya, S. Z. (2016). A comprehensive survey of pilot contamination in massive MIMO-5G. IEEE Communications Surveys & Tutorials, 18(2), 905–923.

6. Hu, D., He, L., & Wang, X. (2016). Semi-blind pilot decontamination for Massive MIMO systems. IEEE Transactions on Wireless Communications, 15(1), 525–536.

7. Han, H.-D., Nguyen, D.-A., Nguyen, V.-D., Nguyen, T.-H., & Zia, M. (2018). Pilot decontamination for multi-cell massive MIMO systems using asynchronous pilot design and data-aided channel estimation. Physical Communication, 76–85.

8. Nguyen, D.-A., Nguyen, A.-T., Han, H.-D., Nguyen, V.-D., & Zia, M. (2018). Pilot decontamination using time-shifted pilot and data-aided channel estimation for massive MIMO. IEEE ICCE, Vietnam.

9. Maruta, K., & Ahn, Ch.-J. (2018). Uplink interference suppression by semi-blind adaptive array with decision feedback channel estimation on multicell massive MIMO systems. IEEE Transactions on Communications, 66(12), 6123–6134.

10. Godard, D. N. (1980). Self-recovering equalization and carrier tracking in two-dimensional data communication systems. IEEE Transactions on Communications, 28, 1867–1875.

11. Johnson, R., Schniter, P., Endres, T. J., Behm, J. D., Brown, D. R., & Casas, R. A. (1998). Blind equalization using the constant modulus criterion: a review. Proceedings of the IEEE, 86(10), 1927–1950.

12. Oh, K. N., & Chin, Y. O. (1995). Modified constant modulus algorithm: Blind equalization and carrier phase recovery algorithm. In IEEE Conference on Communications, USA.

13. Yang, J., Werner, J. J., & Dumont, G. A. (2002). The multimodulus blind equalization and its generalized algorithms. IEEE Journal on Selected Areas in Communications, 20(5), 997–1015.

14. Yang, J., Werner, J. J., & Dumont, G. A. (1997). The multimodulus blind equalization algorithm. IEEE ICDSP, Santorini, Greece.

15. Abrar, S., & Nandi, A. K. (2010). Blind equalization of square-QAM signals: A multimodulus approach. IEEE Transactions on Communications, 58(6), 1674–1685.

16. Sun, Y., Zhu, L., Li, D., & Liu, Z. (2019). Convex combination of SISO equalization and blind source separation for MIMO blind equalization. Wireless Personal Communications. pp. 1397–1409.

17. Maruta, K., & Ahn, Ch.-J. (2019). Enhanced semi-blind uplink interference suppression on multicell massive MIMO systems for multi modulus signals. IEEE VTC2019-Fall, pp. 1–6, USA.

18. Rekik, O., Mokraoui, A., Ladaycia, A., & Abed Meraim, K. (2019). Semi-blind source separation based on multi-modulus criterion? Application for pilot contamination mitigation in massive MIMO communications systems. In IEEE ISCIT 2019. (Vol. 9(11), pp. 3590–3600).

19. Rekik, O., Abed Meraim, K., & Mokraoui, A. (2020). Multi-Modulus based semi-blind source separation for MIMO-OFDM communications systems. In IEEE SAM (pp. 1–5). China.

20. Comon, P. (2000). Block methods for channel identification and source separation. In IEEE ASSPCC (pp. 87–92). Lake Louise, AB, Canada.
Noura Sellami received the Preliminary Doctorate Certificate in signal processing from the University of Cergy-Pontoise (UCP), Cergy-Pontoise, France; the Engineer Diploma from the École Nationale Supérieure de l’Électronique et de ses Applications, Cergy, France, in 1999; and the Ph.D. degree from the UCP in 2002. From 1999 to 2002, she was with Orange Labs, Issy-les Moulineaux, France. From 2003 to 2008, she was an Assistant Professor with the Institut Supérieur de l’Électronique et de Communication de Sfax, Sfax, Tunisia. In 2008, she joined the École Nationale d’Ingénieurs de Sfax (ENIS), Sfax, where she is currently a Full Professor. Her research interests are in the area of digital communications with special emphasis on turbo receivers, cognitive radio, cooperative networks, optimization of the physical layer resources and massive MIMO systems.

Mohamed Siala received his general engineering degree from Ecole Polytechnique, Palaiseau, France, in 1988, his specialization engineering degree in telecommunications from Telecom ParisTech, Paris, France, in 1990, and his Ph.D. in digital communications from Telecom ParisTech, Paris, France, in 1995. From 1990 to 1992, he was with Alcatel Radio-Telephones, Colombes, France, working on the implementation of the GSM physical layer. In 1995, he joined Wavecom, Issy-les-Moulineaux, France, where he worked on advanced multicarrier communications and channel estimation for low-orbit mobile satellite communications. From 1997 to 2001, he worked at Orange Labs, Issy-les-Moulineaux, France, on the assessment of the performance of the physical layer of 3G systems and participated actively in their standardization. In 2001, he joined SUP’COM, Tunis, Tunisia, where he is now a full Professor. His research interests are in the areas of digital and wireless communications with special emphasis on waveform design for advanced multicarrier systems and ARQ, channel estimation, synchronization, adaptive modulation and coding, MIMO systems, massive MIMO, space-time coding, relaying, cooperative networks and cognitive radio. He holds respectively 40 and 221 papers in prestigious journals and conferences and 6 patents.