Performance Evaluation and Implementation Complexity Analysis Framework for ZF Based Linear Massive MIMO Detection

Messaoud Ahmed Ouameur, Daniel Massicotte, Auon Muhammad Akhtar, and Reno Girard

***Abstract*— This paper discusses a framework for algorithm-architecture synergy for (i) performance evaluation and (ii) FPGA implementation complexity analysis of linear massive MIMO detection techniques. Three low complexity implementation techniques of the zero-forcing (ZF) based linear detection are evaluated, namely, Neumann series expansion (NSE), Gauss-Seidel (GS) and a proposed recursive Gram matrix inversion update (RGMIU) techniques. The performance analysis framework is based on software-defined radio (SDR) platform. By extrapolating the real data measured average error vector magnitude (EVM) vs a number of served single-antenna user terminals (UTs), GS and RGMIU are showing no performance degradation with respect to ZF with direct matrix inversion. It is shown that under high load regime NSE and GS require more processing iterations at the expense of increased processing latency. We, therefore, consider a unified approach for field-programmable gate array (FPGA) based implementation complexity analysis and discuss the required baseband processing resources for real-time transmission. Due to the wide differences of NSE, GS and RGMIU in terms of performance, processing complexity and latency, practical deployment and real-time implementation insights are derived.**

Index Terms—Massive MIMO, zero-forcing (ZF), Maximum ratio combining (MRC), receiver combining, detection, matrix inversion update, Neumann series expansion (NSE), Gauss-Seidel (GS), FPGA implementation, real data detection, real-time transmission.

1. Introduction

Being a promising concept for future cellular networks, massive multiple-input multiple-output (MIMO) has now made its way to 5G as one of the means to substantially improve both spectral and energy efficiencies [1]. As a matter of fact, base stations (BSs) with 64 fully digital transceiver chains are commercially deployed and the key component of massive MIMO has made its way into the 5G standard [2]-[3]. Nevertheless, the authors in [4] have pointed out that massive MIMO implementation continues to be at least as exciting as massive MIMO theory. Massive MIMO is a form of multiuser MIMO where the number of serving antennas at the base transceiver station (BS) is an order of magnitude larger than the number of user terminals (UTs) served within each radio resource element. Given the large number of antennas, reliance on time division duplex (TDD) channel reciprocity is essential [1].

Prototyping is one of the effective ways to answer a number of pending questions such as (i) how much of the theoretical gains can be harvested in real propagation channel and in the presence of hardware impairments and (ii) how efficiently does the system scale with the number of BS antennas and the number of user terminals (UTs) while maintaining the overall system energy efficiency lower. A 96 antennas BS Argos testbed [5] was among the early testbeds which demonstrated that massive MIMO can enable high spectral efficiency. A successful real-time uplink transmission was demonstrated using a massive MIMO testbed build at Lund University (LuMaMi) [6]. Although these works have reported large throughputs and spectral efficiencies on the uplink and downlink using zero-forcing (ZF) and maximum ratio combining (MRC), our contribution complements them by providing insights on deployment aspects if different detection techniques are used. More importantly, a different hardware (HW) platform based on Nutaq Innovation's software-defined radio (SDR) development platform is utilized [7].

Under favorable channel conditions and/or as the number of antennas increases, the UTs' channels are mutually orthogonal which makes linear processing (detection and precoding), such as MRC, ZF and minimum mean square error (MMSE) detection techniques, optimal [8]. The detection/precoding problem based on ZF or MMSE technique is an arithmetic operation with cubic computational complexity in the order of the matrix dimension. To reduce the implementation complexity, matrix inversion approximations such as Neumann series expansion (NSE) is proposed [9]. Recently, a technique based on Gauss-Seidel (GS) was shown to outperform NSE due to its fast convergence at considerably low computational complexity [10]. However, this comes at the expense of higher latency and lower throughput [10]. It has actually been shown that the NSE performance degrades as the number of UTs increases [11]. 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 [10]. It has therefore been argued to resort to exact

M. Ahmed-Ouameur is with Laboratoire des Signaux et Systèmes Intégrés, Université du Québec à Trois-Rivières, Department of Electrical and Computer Engineering, 3351, Boul. des Forges, Trois-Rivières, Québec, Canada, and also with NUTAQ Innovation, 2150 Rue Cyrille-Duquet, Québec, Québec, Canada (e-mail: messaoud.ahmed.ouameur@uqtr.ca)

D. Massicotte (*Corresponding author*) is with Laboratoire des Signaux et Systèmes Intégrés and holds a the Chaire de recherche sur les signaux et l'intelligence des systèmes haute performance, Université du Québec à Trois-Rivières, Department of Electrical and Computer Engineering, 3351, Boul. des Forges, Trois-Rivières, Québec, Canada. (e-mail: daniel.massicotte@uqtr.ca)

M. Akhtar and Reno Girard are with NUTAQ innovation, 2150 Rue Cyrille-Duquet, Québec, Québec, Canada. (e-mail: auon.akhtar@nutaq.com, reno.girard@nutaq.com)

matrix inversion [11]. On the other hand, it has also been argued that these centralized processing techniques still impose stringent constraints on the interconnects' bandwidth between the massive MIMO radio heads (RHs) and the central processing unit (CPU). Distributed, or decentralized, massive MIMO processing has been introduced to overcome such limitations [12] and [13]. Unfortunately, the decentralized processing computational complexity, and hence the energy efficiency, are also of concern [14]. On the other hand, to support ultra-reliable low-latency communications (URLLCs) low latency and high throughput processing is required. As such, we introduce a recursive Gram matrix inversion update (RGMIU) method as an extension to [16] wherein the inversion of the Gram matrix is performed by exploiting matrix inversion update of a matrix in the form of $\mathbf{H}^H\mathbf{H}$ when a new column is added/updated to a complex-valued matrix $\mathbf{H}$ (early work in [16] has already proposed matrix inverse update when a new column is added but did not apply it recursively considering one column at a time). Herein, direct matrix inversion based on Cholesky decomposition is considered as a reference from the performance and computational complexity standpoint.

As part of the massive MIMO prototyping effort, implementation complexity analysis is performed by adopting a unified field-programmable gate array (FPGA) based implementation framework that provides a fair assessment of the different methods mentioned above. Our approach is to *adopt* and *reuse* the same *pipelined array core* like the one used for Gram matrix computation to perform the matrix inversion operation for NSE, GS and RGMIU methods. As such, one would instantiate as many cores as possible depending on the FPGA's available resources (parallelism). As the core resources scale with K^2 (where K is the number of single-antenna UTs), we will resort to reusing the core as often as possible by exploiting the high operating frequency of the DSP48s multipliers. The reuse factor depends on the latency, which in turn is dictated by the inherent processing regularity and data dependencies.

Due to their wide differences in terms of performance, processing complexity and latency; NSE, GS, and RGMIU techniques represent a fair choice to enable the discuss and infer the key insights on the practical deployment and real-time transmission aspects.

The main contributions of the paper help to introduce the recursive Gram matrix inversion update (RGMIU) method and gain insights on

- The expected performance of the linear massive MIMO detection techniques with real propagation environment, channel estimation errors and hardware impairments that are *inherent* to the adopted SDR platform and the reference orthogonal frequency division multiplex (OFDM) waveform. Using the extrapolated measured error vector magnitude (EVM), deployment aspects are discussed based on two objectives namely (i) maximizing the cell throughput and (ii) maximizing per UT throughput.
- The impact of the implementation complexity and the latency on real-time transmission using FPGAs as computing nodes.

This paper will mainly focus on the uplink combining/detection but the problem formulation and solution can be extended to cover the downlink precoding as well.

The paper is organized as follows: Section 2 presents the uplink signal model and the low complexity implementations for the ZF detection technique. Performance evaluation, using an LTE-like TDD-OFDM waveform and frame structure running in an SDR platform, in a static indoor propagation channel, is discussed in Section 3. Section 4 is dedicated to FPGA based implementation analysis for a real-time transmission where a unified approach using a single pipelined architecture is adopted. Finally, the conclusions are drawn and some future research directions are outlined in Section 5.

Notations- This paper adopts the following notations: $(\bullet)^H$ represents the Hermitian transpose operator while $(\bullet)^T$ and $(\bullet)^{-1}$ represent the transpose and the matrix inverse operators respectively. $\mathbf{H}_{\mathrm{n:m}}$ denotes a matrix comprising of columns *n* to *m* of the original $M \times K$ matrix $\mathbf{H}$, whereas $\mathbf{H}_{K-1/k}$ represents a $M \times (K-1)$ matrix without the column *k* of $\mathbf{H}$. $\mathbf{\Delta}_k$ demotes a square $k \times k$ matrix whose dimension can change from 1×1 (for a scalar) to $K \times K$. $(\bullet)!$ and $\log_2(\bullet)$ are the factorial operator and logarithm base 2 function respectively.

2. Low Complexity Linear Detection Techniques

A. Signal model and zero-forcing detection technique

We consider an uplink transmission where K single-antenna UTs are communicating with a BS equipped with M antennas (where $M \gg K$) in a TDD duplex mode using the OFDM modulation scheme. For the sake of simplicity, we consider a baseband equivalent channel and expressions per subcarrier where the subcarrier index is suppressed. The data signal of the k^{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

$$\mathbf{y} = \mathbf{Hs} + \mathbf{n} \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{M\times 1}$, $\mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \cdots \quad \mathbf{h}_K]$ is the channel matrix and $\mathbf{s} = [s_1 \quad s_2 \quad \cdots \quad s_K]^T$. $\mathbf{n} \in \mathbb{C}^{M\times 1}$ represents the additive receiver noise vector whose entries are zero mean and variance equal to σ^2.

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

$$\hat{\mathbf{s}} = \mathbf{Wy} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{y}_{MF} = \mathbf{\Delta}_{ZF}\mathbf{y}_{MF} \tag{2}$$

where $\mathbf{y}_{MF} \triangleq \mathbf{H}^H\mathbf{y}$. Notice that the MRC technique considers $\mathbf{\Delta}_{ZF} \cong (diag(\mathbf{H}^H\mathbf{H}))^{-1}$ where $diag(\bullet)$ represents the diagonal

TABLE 1. THE PROPOSED RGMIU FOR ZF COMBINING WEIGHTS COMPUTATION BY RECURSIVELY ADDING ONE UT AT A TIME.

```
INPUT: $\mathbf{H}$ (Consider the UTs' channel vectors as input.)
INITIALIZE: $\mathbf{\Delta}_1 = 1/\mathbf{H}_{1:1}^H\mathbf{H}_{1:1}$ (Pre-compute $\mathbf{\Delta}_1$ as a scalar division based on the first UE channel column vector.)
FOR $k=2$ to $K$ do ( $K$ being the maximum number of UTs)
    1. $\mathbf{z} = \mathbf{H}_{k:k}$ , The $k$-th column of $\mathbf{H}$ represents the next UT's channel column vector.
    2. $\mathbf{y}_1 = \mathbf{H}_{1:k-1}^H \mathbf{z}$
    3. $\mathbf{y}_2 = \mathbf{\Delta}_{k-1} \mathbf{y}_1$
    4. $c = 1/\left(\mathbf{z}^H\mathbf{z} - \mathbf{y}_1^H\mathbf{y}_2\right)$
    5. $\mathbf{y}_3 = c\,\mathbf{y}_2$
    6. $\mathbf{\Gamma} = \mathbf{\Delta}_{k-1} + c\,\mathbf{y}_2\mathbf{y}_2^H$
    7. $\mathbf{\Delta}_k = \begin{bmatrix} \mathbf{\Gamma} & -\mathbf{y}_3 \\ -\mathbf{y}_3^H & c \end{bmatrix}$
END FOR
Consider permutation if the last column/row needs to be repositioning at another column/row (the case if the matrix inversion needs to be updated when an existing UT channel changes for instance)
OUTPUT: $\mathbf{W} = \mathbf{\Delta}_K\mathbf{H}^H$
```

operator. The corresponding signal-to-interference-and-noise ratio (SINR) per UT k is [17]

$$SINR_k = \frac{q_k \left|\mathbf{h}_k^H \mathbf{w}_k\right|^2}{\sum_{i \neq k} q_i \left|\mathbf{h}_i^H \mathbf{w}_k\right|^2 + \sigma^2 \mathbf{w}_k^H \mathbf{w}_k} \tag{3}$$

upon which we define the achievable user rate for the k-th UT as $c_k = \log\left(1 + SINR_k\right)$ where q_k is the corresponding transmit power (we assume here to be equal to 1 for all users).

B. *Low complexity implementation techniques*

This section briefly summaries two low complexity approximation techniques namely NSE [9] and GS [10], and an exact matrix inversion method RGMIU.

a) Neumann series expansion approximation

The NSE approximates the inverse of the Gram matrix $\mathbf{\Delta} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}$ by keeping the first N (order) terms of the Neumann series [9] i.e.

$$\begin{aligned} \mathbf{\Delta} &\cong \sum\nolimits_{n=0}^{N-1} \left(-\mathbf{D}^{-1}\mathbf{E}\right)^n \mathbf{D}^{-1} \\ &= \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{E}\mathbf{D}^{-1} + \left(\mathbf{D}^{-1}\mathbf{E}\right)^2 \mathbf{D}^{-1} + \sum\nolimits_{n=3}^{N-1} \left(-\mathbf{D}^{-1}\mathbf{E}\right)^n \mathbf{D}^{-1} \end{aligned} \tag{4}$$

Where $\mathbf{D}$ and $\mathbf{E}$ are the main diagonal and the off-diagonal parts of the Gram matrix $\mathbf{H}^H\mathbf{H}$ respectively. The NSE approximation is computationally efficient if and only if $N \leq 3$ as depicted in the first three terms in (4).

b) Gauss-Seidel approximation

In the GS method, the Hermitian positive definite Gram matrix $\mathbf{H}^H\mathbf{H}$ is decomposed as $\mathbf{H}^H\mathbf{H} = \mathbf{D} + \mathbf{L} + \mathbf{L}^H$ where $\mathbf{D}$, $\mathbf{L}$ and $\mathbf{L}^H$ are the diagonal, lower triangular and upper triangular parts of the Gram matrix [10]. The received signal is estimated as

$$\mathbf{s}^{(i)} = \left(\mathbf{D} + \mathbf{L}\right)^{-1} \left(\mathbf{y}_{MF} - \mathbf{L}^H \mathbf{s}^{(i-1)}\right), i = 1, 2, ..., \tag{5}$$

where i is the number of iterations (order) and the initial solution $\mathbf{s}^{(0)}$ can be computed using the second-order Neumann series approximation i.e. $\mathbf{s}^{(0)} = \left(\mathbf{I} - \mathbf{D}^{-1}\mathbf{E}\right)\mathbf{D}^{-1}\mathbf{y}_{MF}$.

TABLE 2. THE PROPOSED RGMIU FOR ZF COMBINING WEIGHTS UPDATE WHEN REMOVING A UT AT COLUMN 'K'.

```
INPUT: $\mathbf{\Delta}_K = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}$
INITIALIZE: Permute column $k$ and row $k$ of $\mathbf{\Delta}_K = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}$ to the last column and last row, rename it $\mathbf{X}$ .
DO
    1. $\mathbf{\Gamma} = \mathbf{X}_{1:K-1,1:K-1}$
    2. $c = \mathbf{X}_{K,K}$
    3. $\mathbf{y}_2 = -\mathbf{X}_{1:K-1,K}$
    4. $\mathbf{y}_1 = \mathbf{y}_2/c$
    5. $\mathbf{\Delta}_{K-1} = \mathbf{\Gamma} - c\,\mathbf{y}_1\mathbf{y}_1^H$
END DO
OUTPUT: $\mathbf{W}_{K-1} = \mathbf{\Delta}_{K-1}\mathbf{H}_{K-1/k}^H$ ( $\mathbf{H}_{K-1/k}$ denotes $\mathbf{H}$ without the $k$-th column.)
```

c) Recursive Gram matrix inversion update (RGMIU)

By exploiting the Gram matrix structure in $\mathbf{\Delta} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}$ one can devise an efficient recursive algorithm based on matrix inversion update where a new column is added [16]. Herein a new column refers to a new UT channel vector. By recursively updating the matrix inverse as a new UT is scheduled, the proposed extension is outlined in Table 1. We refer to it as a recursive Gram matrix inversion update (RGMIU). To make the paper self-contained, Appendix A summarises the matrix inversion update to support Tables 1 and 2. On the other hand, implementing the matrix inversion update if a new UT leaves the network is outlined in Table 2. Therefore, the computational complexity is considerably low as the matrix inversion boils down to run one single iteration pass to remove the non-active UT only.

When a UT's channel state information (CSI) changes the matrix inversion update will be performed by first removing the associated UT and add it back with a new CSI which implies two passes; one pass to update the matrix inverse by removing the associated column (Table 2) and the next pass by updating the matrix inverse by adding a column (Table 1). The column here is the associated UT k channel vector. If a column shall be repositioned at column 'k', the algorithm shall permute the last row and column to the k-th row and the k-th column respectively. The recursive nature of the scheme renders it suitable when the UTs have different channel coherence time constrains where only UTs with short coherence time need faster updates [18]. The impact on computational complexity saving is substantial.

C. Performance analysis

The simulation is based on using an LTE-like TDD-OFDM waveform and frame structure discussed in Section 3. Since the impact of the channel estimation errors and the hardware impairments on the performance of the ZF-based processing is well documented [19]-[20], perfect CSI is assumed in this subsection. However, we assess the performance for 12 single-antenna UTs communicating with 64 or 32 antennas BS in terms of the average BER and root mean squared (rms) EVM (c.f. Fig. 1) where we kept the ratio of the number of users to the total number of antennas at the BS close to 5. In fact, this massive MIMO regime is shown to be the optimal point for ZF based detection to achieve maximum cell spectral efficiency (see figure 4 in reference [21]).

The relative performance of NSE and GS has already been reported in [9], [10] and [11] and confirmed in our simulations. Nevertheless, we attempt to point to the fact that as the ratio of the number of antennas at the BS to the number of the UTs gets lower (i.e. high load regime), NSE and GS need more iterations to keep up close to ZF with direct matrix inversion. Unfortunately, an extra iteration will translate into a substantial increase in the computational complexity and/or processing latency as will be discussed shortly. Figures 1.a and 1.b depict the bit error rate (BER) as a function of the signal-to-noise ratio (SNR) for a BS with 32 and 64 antennas serving 12 users respectively. Comparing the BER curves, as the system load increases, GS's performance degrades substantially which suggests that more iterations are required. Similar performance degradation is observed using NSE even at a higher order. On the other hand, being an exact matrix inversion technique, the proposed RGMIU shows no performance degradation.

Of particular interest, Fig. 1.c shows the post multiuser detection root mean squared (rms) EVM with QPSK (18.5% rms EVM), 16-QAM (12.5% rms EVM) and, 64-QAM (8.5% rms EVM) modulations limits[1]. Based on the extended relationships among rms EVM, BER and SNR [22], these limits correspond to a raw BER of 10^{-3}, 10^{-4} and 10^{-5} respectively. The raw BER P_b is shown to be related to the rms EVM as

$$P_b \approx \frac{2\left(1-1/\sqrt{M}\right)}{\log_2\left(\sqrt{M}\right)} Q\left[\sqrt{\left(\frac{3\log_2\left(\sqrt{M}\right)}{M-1}\right)\left(\frac{2}{EVM_{rms}^2 3\log_2\left(M\right)}\right)}\right]$$

for M-ary square QAM modulation where $M=64$ for 64-QAM modulation using coherent detection. $Q[\bullet]$ is the Gaussian co-error function and is given by $Q(x)=\int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy$. These limits can also be inferred from figure 2 in [22].

We define the per subcarrier sum rate (or cell rate per subcarrier) as $K \cdot \log_2 M$ bits/subcarrier if all K UTs share the same subcarrier using M-ary modulation with a given rms EVM limit. Therefore, with no channel estimation errors and no HW impairments (including timing mismatches), the proposed RGMIU can support 12× UTs using 64-QAM modulation at 10 dB SNR whereas GS and NSE would support 12× UTs and 6× UTs using 16-QAM modulation per subcarrier respectively. This translates to the expected per subcarrier sum rate of ($12 \cdot \log_2 64$) 72 bit/Subcarrier, ($12 \cdot \log_2 16$) 48 bit/Subcarrier and ($6 \cdot \log_2 16$) 36 bit/Subcarrier respectively. One can even expect to reach 96 bit/Subcarrier using RGMIU if 256-QAM modulation is used at 12 dB SNR. This corresponds to a maximum UL cell throughput of 576 Mbit/s or an aggregate UL and DL throughput over 1 Gb/s, where we have assumed that 1200 subcarriers per OFDM symbol and 5 OFDM data symbols

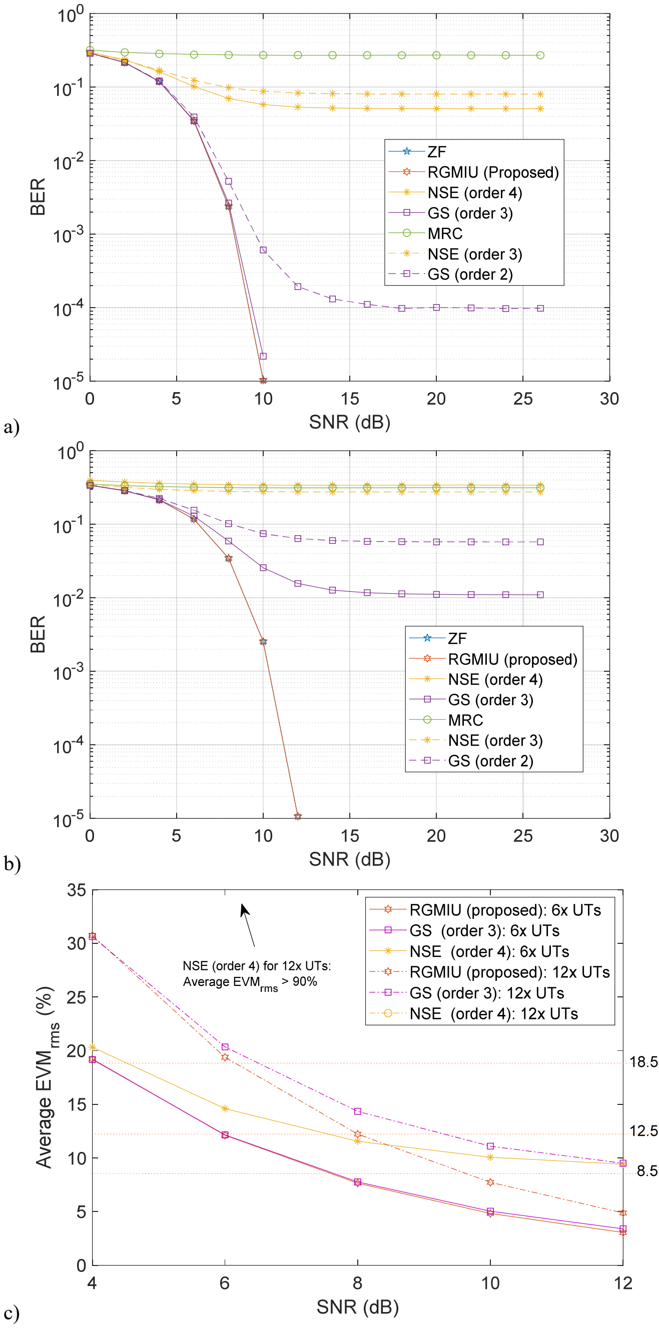


Fig. 1. Average BER for 12 UTs communicating with a) 64 antennas BS and b) 32 antennas BS, and c) RMS EVM (%) for 6x UTs and 12xUTs served by a 32 antennas BS (being similar to Proposed RGMIU, ZF curve is deleted for clarity).

[1] Note that these limits are also establish by 3GPP for LTE transmitter's modulation accuracy performance.

TABLE 3. TDD-OFDM WAVEFORM PARAMETERS

Parameter	Value
Sampling rate f_s	30.72 MHz
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 μs
UL-DL and DL-UL switching guard	75.00 μs

are reserved for UL data within 1 msec sub-frame duration (more details on the waveform is discussed in section 3-A).

3. PERFORMANCE EVALUATION FRAMEWORK USING STATE OF THE ART SDR PLATFORMS

This section discusses the SDR hardware architecture and the underlying TDD-OFDM waveform and frame structure used to build an uplink Matlab based reference design for massive MIMO detection techniques performance evaluation with real data. The real data reference prototyping system is based on a 32 antennas BS serving up to 6 single-antenna UTs *where the ratio of the number of the BS antennas to the number of UTs is greater or equal to 5*. It has been shown that, for ZF detection, this is an optimal operating point to achieve maximum cell spectral efficiency [21]. It is worth noting that [23] and [24] have reported experimental results with 128 antennas BS serving 12 single-antenna UTs. However, we are interested in massive MIMO operating regime where the number of BS antennas is less than 64 to shed light on the expected performance in systems similar to LTE-advanced pro with full-dimension MIMO (FD-MIMO) feature where the array size is set to a maximum of 64 elements[2]. Nevertheless, the evaluation based on the SDR set-up with 6 single-antennae UTs and 32 antennas BS keeps this ratio equal to 5 while considering the channel estimation errors and the hardware impairments that are *inherent* to the SDR-based MIMO-OFDM reference design. So, the channel is estimated using the uplink pilots based on the least square (LS) method. The hardware impairments are part of the UT transmitter chain's imperfections (e.g. pre-driver's non-linearities and local oscillator (LO) phase noise, etc.) and the BS receiver chains' imperfections (e.g., low noise amplifiers (LNA), LO phase noise and non-ideal zero-IF mixers, etc.). These are hard to quantify. Even in line-of-sight (LOS) scenario the transmitter-receiver chains introduce random gains and phases that are part of the complex-valued baseband channel gain. Since the reference design does not implement carrier frequency offset and sampling rate offsets (CFO and SRO) compensation methods, we had to share the same reference clock (but NOT the same local oscillators). This limited the use case to a LOS scenario. However, the impact of CFO and SRO estimation error is shown to be substantial enough (see our previous work [25]) that it will dominate the performance and mask the benefits of the detection techniques. Even with such limited set-up one can infer key insights related to the deployment scenarios under the limitation imposed by none CFO and SRO estimation errors.

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

An LTE-like TDD-OFDM frame structure and waveform is depicted below[3]. Fig. 2.a shows a 10 ms TDD frame structure divided into 10 sub-frames. Sub-frame 0 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 μs. As such, it is apparent that the waveform is suitable for channel coherence time higher than 1 ms. The TDD-OFDM waveform parameters are outlined in Table 3 where FFT, IFFT, and CP stand for the fast Fourier transform, the inverse fast Fourier transform and cyclic prefix respectively.

The pilot and data assignment over a total of 1200 subcarriers is shown in Fig. 2.b. The UL pilots are interleaved among the active UTs. This is a simple pilot allocation scheme but yet

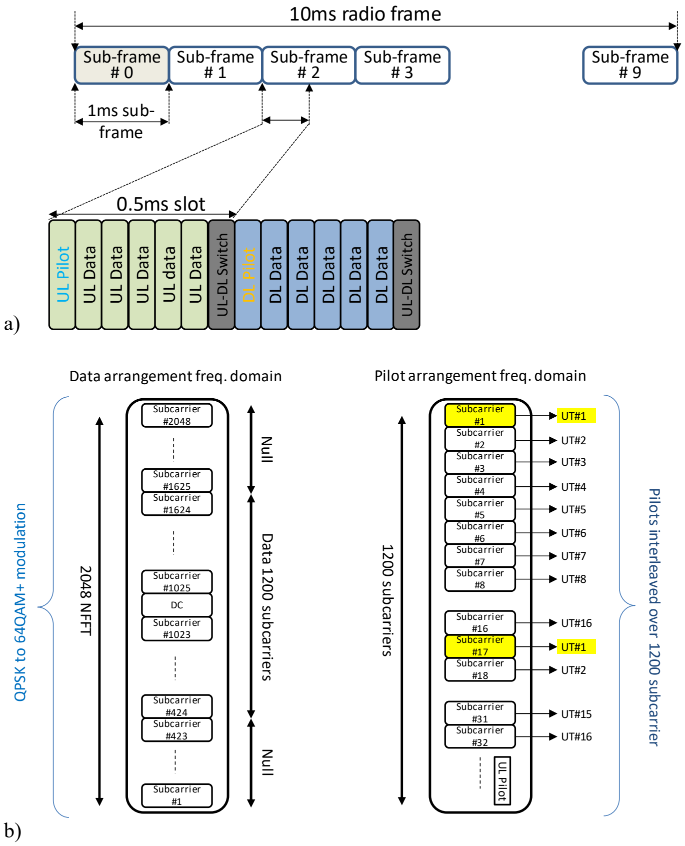


Fig. 2. a) TDD-OFDM frame structure and b) Pilot and data subcarrier allocation

[2] We also expect early deployments will be limited to less than 64 antenna elements for the sake of size and energy efficiency.

[3] In fact, the LTE uplink uses SC-FDMA whereas the downlink adopts FDMA. This is mainly due to the fact that handsets can not afford to support a high peak to average power ratio (PAPR) inherent in FDMA. Our reference SDR design uses FDMA on both links to keep it symmetrical and ease waveform development especially in assigning uplink pilots to the users. Since the processing is done after IFFT, we do not expect noticeable impact unless the radio at the UE side is running at high output power which would make PAPR affect the uplink performance. Particularly, the performance would be decreased in the UL for UEs operating at high power, e.g. signaling at the cell margin. We (coarse) tuned the output power so that the UEs operate in a safe non-saturated region especially that the set-up is constrained to line-of-sight.

ensures orthogonality among the UTs within the channel coherence bandwidth. Fig. 2.b illustrates allocation of up to 16 UTs as far as the channel coherence bandwidth of the propagation environment is lower than 240 kHz, which is quite feasible in typical indoor and outdoor environments. It is inherently expected that the UL channel estimates at other subcarriers that are not assigned to a given UT are computed using simple interpolation scheme [23]. One can exploit the ideas in [26] for effective uplink channel sounding.

B. Hardware architecture and components

The massive MIMO reference system consists of a 32 antennas BS and 6 single-antenna UTs. The antenna array uses 2.4 GHz/5.5 GHz patch antennas arranged in 4×8 planar array. The BS uses 4× TitanMIMO from NUTAQ innovation [7] while the UTs are based on Radio640 [27] from NUTAQ innovation and ZC706's Zynq based evaluation board [28] from Xilinx. Each Radio640-ZC706 represents 2 single-antenna UTs given that Radio640 supports two independent transceivers simultaneously. Fig. 3.a shows the overall massive MIMO system with the constituting synchronization and reference clock modules. It is worth noting that all radios share the same reference clock and transmission synchronization signal from the master module located in the BS (one of the 4 TitanMIMOs is chosen as a master module).

C. Performance evaluation in static indoor LOS propagation channel

The performance of the different detection methods discussed in Section 2 is evaluated herein using the massive MIMO reference prototyping system and the TDD-OFDM frame structure and waveform. The UL transmission is triggered by the BS to have all the UTs transmit a one time-slot worth of UL pilot and data (c.f Fig. 2.a). The transmission synchronization network ensures synchronous transmission from all UTs. The BS records the one time-slot and then transfers them to the host PC.

The host PC runs a Matlab program that performs frame synchronization based on CP correlation, IFFT, channel estimation, and interpolation, and then executes the different detection techniques on the data subcarriers. So far, such a massive MIMO reference prototyping system can also be used to investigate the different hardware impairments effects[4] [29] and assess different waveforms, detection, and processing methods over a real word propagation environment. The current massive MIMO reference prototyping system is limited LOS given the constraint imposed by sharing the reference clock and transmission synchronization signals[5]. The UTs are set in an arc at 30cm from each other. The BS is positioned at 3m equal distance from all UTs as shown in Fig. 3.b.

Fig. 4.a depicts *the simulation results* of the average rms EVM versus the number of UTs as a function of the SNR. Recall that the rms EVM limits for QPSK (18.5%), 16-QAM (12.5%) and 64-QAM (8.5%) modulations correspond to the raw BER of 10^{-3}, 10^{-4} and 10^{-5} respectively. At 8 dB SNR, the maximum cell rate per subcarrier is achieved using the proposed RGMIU that supports 12× UTs using 16-QAM (48 bit/subcarrier i.e. $12\cdot\log_2 16$ bit/subcarrier). Meanwhile, GS and NSE can support 7× UTs using 64QAM (42 bit/subcarrier i.e. $7\cdot\log_2 64$ bit/subcarrier) and 5× UTs using 64-QAM (30 bit/subcarrier i.e. $5\cdot\log_2 64$ bit/subcarrier) respectively.

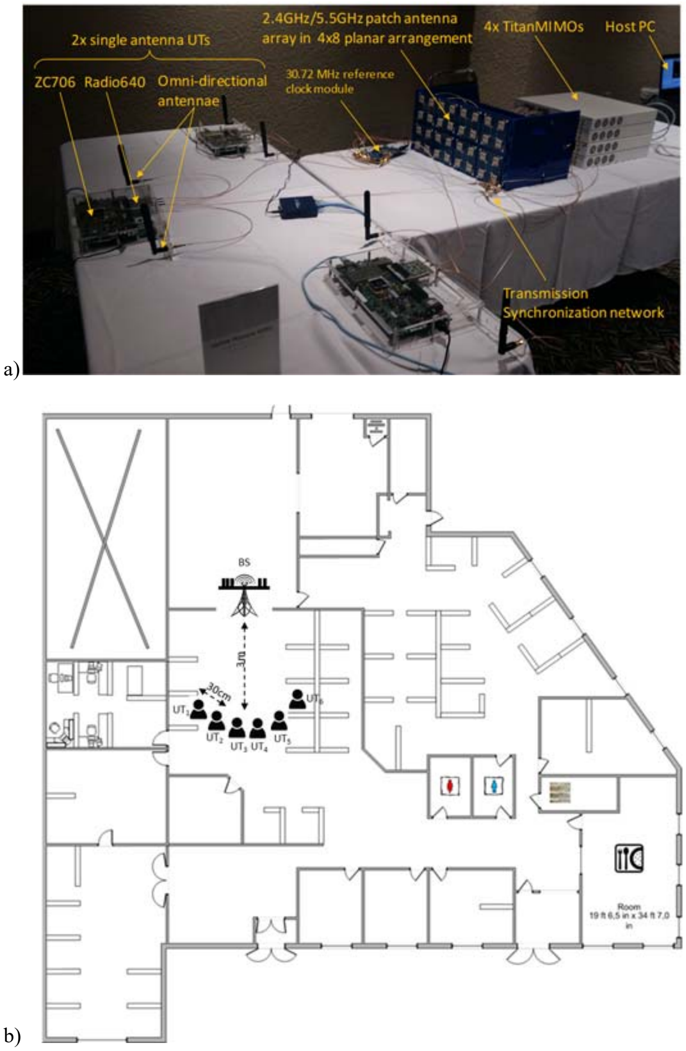

2x single antenna UTs
ZC706
Radio640
Omni-directional antennae
30.72 MHz reference clock module
2.4GHz/5.5GHz patch antenna array in 4x8 planar arrangement
4x TitanMIMOs
Host PC
Transmission Synchronization network
a)
BS
3m
30cm
UT1
UT2
UT3
UT4
UT5
UT6
Room 19 ft 6,5 in x 34 ft 7,0 in
b)

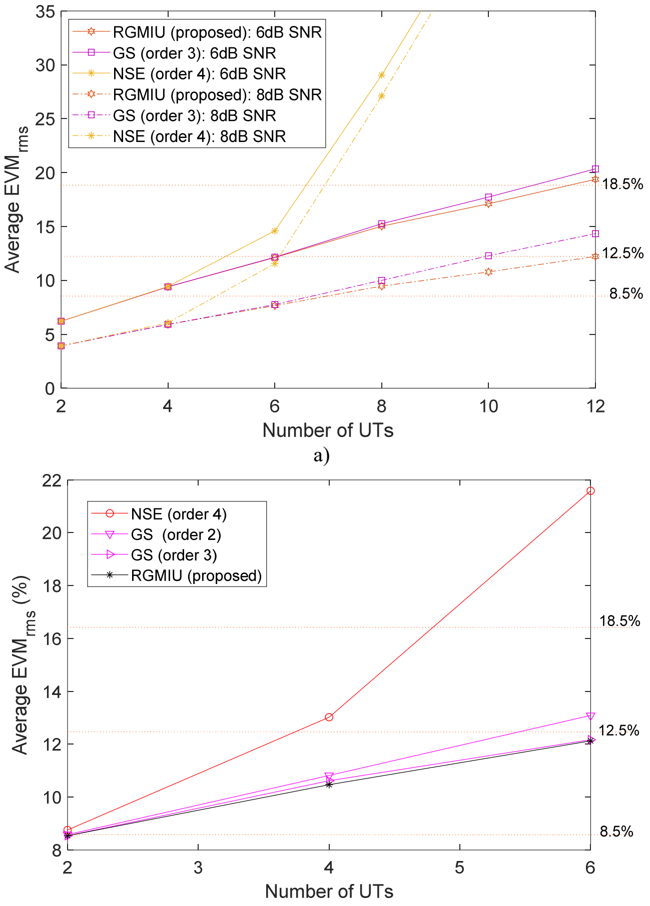
Fig. 3. a) Massive MIMO reference system with 32 antennas BS based on 4x TitanMIMOs and 6 single-antenna UTs based on 3x ZC706/Radio640, and b) NUTAQ's floor plan showing a 32 antennas BS serving 6 single-antenna UTs at equal 3m distance.

On the other hand, Fig. 4.b *depicts the measured average real data* rms EMV on the received signal after multi-user detection as a function of the number of UTs. It is worth noting that the measured rms EVM is subject to channel estimation errors, HW impairments, and transmission synchronization mismatches that are *inherent* to the SDR platform. These effects are not fine-tuned to reflect the expected performance using low-quality HW and/or low complexity processing (e.g. channel estimation schemes).

The real data performance in Fig. 4.b agrees with the simulation results, wherein in high load conditions, NSE (order 4) is showing relatively the poorest performance while

[4] For instance, one can alter the transceiver's local oscillator phase noise by tuning the charge-bump current.

[5] Over the air synchronization and coarse frequency and sampling rate offset compensation is consider for future work.



a)

b)

Fig. 4. Average RMS EVM as a function of the number of UTs communicating with 32 antenna BS a) simulation and b) real measured data (ZF performance curves were omitted as this is very similar to the proposed RGMIU).

GS (order 3) catches up using an extra iteration. As expected RGMIU is performing as well as ZF with direct matrix inversion. Based on EVM limits, one would expect to infer these deployment insights after extrapolating the curves,

- The maximum cell rate per subcarrier is achieved using ZF, RGMIU or GS order 3 with 12 UTs using QPSK modulation or 6 UTs using 16-QAM modulation per UT. Herein the per-subcarrier sum rate is 24 bits/subcarrier. Given the TDD frame structure, this translates to a cell throughput of 144Mbit/s/cell[6]. For the sake of comparison, this is almost 2× increase over the LTE peak uplink data rate in 20 MHz bandwidth. The maximum per UT throughput is 144/6=24 Mbit/s/UT.
- Compared to the recent experimental results in [23] ([24] reported similar results using similar HW), it has been inferred that a peak data rate of 268.8 Mbit/s for 12 single-antenna UTs can be achieved using QPSK in low SNR conditions. Gbit/s peak rates are expected using 256-QAM modulation at high SNR conditions. It, therefore, turns out that *massive MIMO effect can be effective with a number of antennas as low as 32.* However, extensive experimentations shall be conducted at varying propagation channels and load conditions to support such statement.
- The maximum per UT throughput is achieved with a BS serving 2 UTs using 64-QAM modulation with a per-subcarrier rate of 6bits/subcarrier. This amounts to a throughput of 36 Mbit/s/UT[7].
- The NSE (order 4) technique achieves the lowest cell rate per subcarrier of 12bit/subcarrier i.e. half the rate achieved by ZF, RGMIU and GS (order 3).
- The GS (order 2) technique achieves a maximum cell rate per subcarrier of 20bits/subcarrier by serving 10× UTs using QPSK modulation. The GS (order 2) technique can still achieve similar performance as GS (order 3) if the objective is to maximize the per UT throughput (which is achieved by serving 2× UT using 64-QAM).

4. Unified FPGA Based Implementation Analysis for Real-Time Transmission

Recent years have witnessed many proposals on efficient high throughput data detection architectures. NSE and GS based very large scale integration (VLSI) architectures are discussed in [9] and [10] respectively. An optimized coordinate descent (OCD) based method and its VLSI architecture are proposed in [11]. Therein the authors have compared the OCD to NSE, GS and conjugate gradient (CG) in terms of performance and throughput per look-up-table (LUT). The outcomes of such comparison are referenced herein to support our findings. It is worth noting that our work does not consider OCD which does not explicitly compute the inverse of the Gram matrix. Most of the real-time implementations [9], [10] and [11] targets FPGA because of its high computing capability

Nevertheless, it is worth noting that all these approximations (NSE, GS, and OCD) deviate from the optimal performance as the ratio of the number of the BS antennas to the number of UTs is low.

A. Computational complexity analysis

From the computational complexity standpoint, Fig. 5 shows the number of complex multiplications as a function of the number of UTs. Herein the Gram matrix computation is included for a fair comparison. The proposed RGMIU method has lower computational complexity than the NSE of order three. Meanwhile updating one single user involves considerably very low computational complexity as well. For reference, Fig. 5 shows NSE of orders 3 and 4 and ZF with direct matrix inversion using Cholesky decomposition. If NSE of order 4 is computationally higher than ZF with direct matrix inversion, increasing the number of iterations for GS is not introducing substantial computational complexity increase. Unfortunately, the number of operations does not guarantee an efficient implementation as this largely depends on data and processing dependencies as well. This explains in large part why GS has lower throughput compared to NSE [11]. The following subsection addresses and discusses this aspect within a unified FPGA implementation framework.

[6] (24 bits/subcarrier × 1200 subcarrier/symbol × 5 symbols/sub-frame)/(1 ms/sub-frame)

[7] (6 bits/subcarrier × 1200 subcarriers/symbol × 5 symbols)/1 ms.

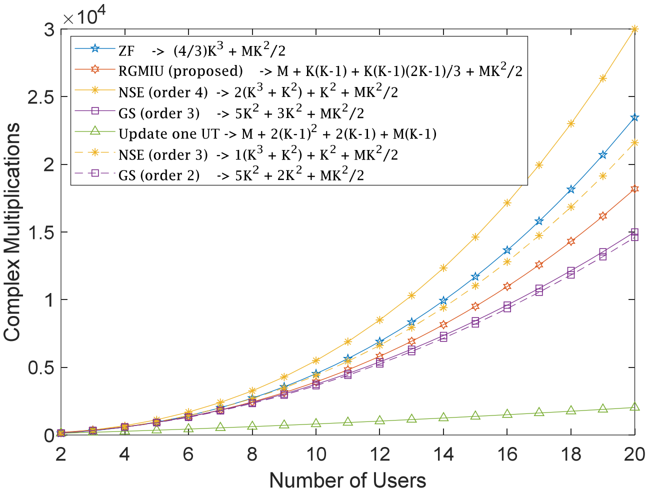


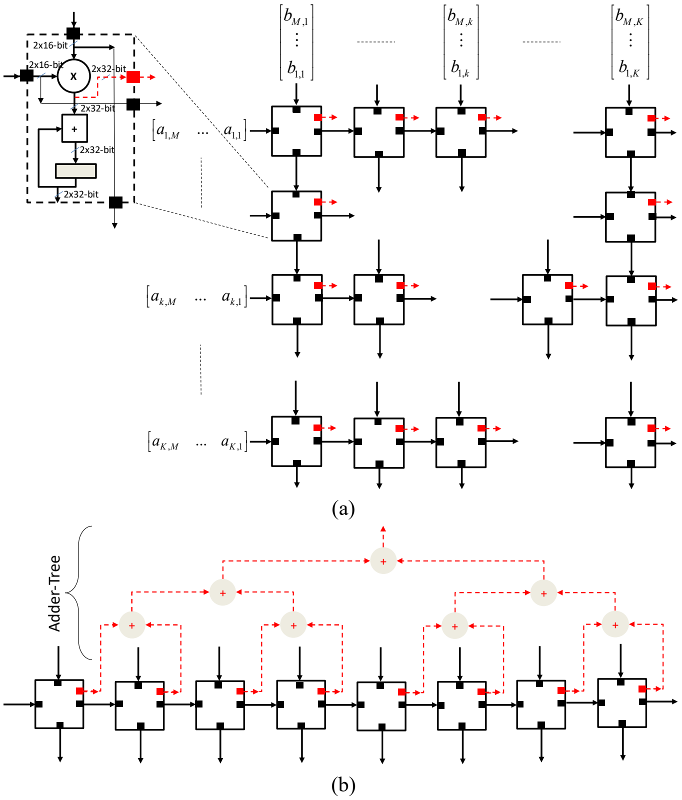
Fig. 5. Computational complexity in terms of complex multiplications for 64 antennas as a function of the number of user terminals.

B. *A unified FPGA processing core and latency analysis*

Most works have focused on matrix inversion implementation arguing that it requires a lot of processing resources. However, the computation of the Gram matrix $\mathbf{H}^H\mathbf{H}$, requiring a complex multiplication of $K \times M$ matrix with its $M \times K$ Hermitian counterpart, turns out to be the dominant operation. This is obviously performed efficiently using the upper or lower side of a $K \times K$ pipelined (systolic) array (see Fig. 6.a) where $M\alpha_{MAC}$ clock cycles are required to get the final result (herein α_{MAC} is the delay in clock cycles to perform one multiply-accumulate operation). In our approach a full $K \times K$ systolic array is used. This same array is *reused* for vector dot products and vector-scalar multiplications. In the vector dot product, of $1\times K$ vector by a $K\times 1$ vector, an adder-tree (ladder) is utilized for summation. Every row and column of the pipelined array is attached to an adder-tree (see Fig. 6.b). When fully used, it will introduce a latency of $\lceil \log_2(K) \rceil \alpha_{ADD}$ where $\lceil \bullet \rceil$ denotes the nearest high integer, $\log_2(\bullet)$ is the logarithm base 2 function, and α_{ADD} is the delay in clock cycles to perform one addition[8].

Fig. 6 depicts a $K \times K$ pipelined array augmented with adder-tree per row which we refer to as detection weight matrix computation (DWMC) core. Each element performs a complex MAC or a multiplication wherein four real multipliers are used (one can reduce it to three if the strength reduction technique is used).

The DWMC core is first used to compute the Gram matrix. It is *then* efficiently reused to carry the computation of the matrix inversion based on either NSE, GS or RGMIU. Fig. 6.c shows explicitly how the DWMC core is used for RGMIU (Table 1). Table 4 depicts the total latency for ZF (using Cholesky decomposition), NSE, GS, and RGMIU when the DWMC core is used. The $\log_2(K!)$ term is derived from the iterative reuse of


2x16-bit
2x16-bit
x
2x32-bit
2x32-bit
+
2x32-bit
2x32-bit
$\begin{bmatrix} b_{M,1} \\ \vdots \\ b_{1,1} \end{bmatrix}$
$\begin{bmatrix} b_{M,k} \\ \vdots \\ b_{1,k} \end{bmatrix}$
$\begin{bmatrix} b_{M,K} \\ \vdots \\ b_{1,K} \end{bmatrix}$
$[a_{1,M} \quad \dots \quad a_{1,1}]$
$[a_{k,M} \quad \dots \quad a_{k,1}]$
$[a_{K,M} \quad \dots \quad a_{K,1}]$
Adder-Tree
+

(a)

(b)

INPUT: H		
INITIALIZE: $\mathbf{\Delta}_1 = 1/\mathbf{H}_{1:1}^H\mathbf{H}_{1:1}$		
FOR $k=2$ **to** K **DO**		
1.	$\mathbf{z} = \mathbf{H}_{k:k}$	-
2.	$\mathbf{y}_1 = \mathbf{H}_{1:k-1}^H \mathbf{z}$	Gram matrix computation using full systolic array. The result is available after M complex MAC: $Ncycles = M\alpha_{MAC}$
3.	$\mathbf{y}_2 = \mathbf{\Delta}_{k-1}\mathbf{y}_1$	At iteration k perform a (k-1)×(k-1) matrix and (k-1)×1 vector multiplication using (k-1) parallel complex MULT followed by (k-1) parallel addition within one complex MULT and log2(k-1) ADD: $Ncycles = \alpha_{MULT} + log_2(k-1)\alpha_{ADD}$
4.	$c = 1/\left(\mathbf{z}^H\mathbf{z} - \mathbf{y}_1^H\mathbf{y}_2\right)$	At iteration k perform 1×(k-1) vector and (k-1)×1 vector dot product using (k-1) parallel complex MULT followed by (k-1) parallel addition within one complex MULT and log2(k-1) ADD. Then one division is performed: $Ncycles = \alpha_{MULT} + log_2(k-1)\alpha_{ADD} + \alpha_{DIV}$
5.	$\mathbf{y}_3 = c\,\mathbf{y}_2$	At iteration k perform a scalar and (k-1)×1 vector product using (k-1) parallel complex MULT within one complex MULT: $Ncycle = \alpha_{MULT}$
6.	$\mathbf{\Gamma} = \mathbf{\Delta}_{k-1} + c\,\mathbf{y}_2\mathbf{y}_2^H$	At iteration k perform (k-1)×1 vector 1×(k-1) vector element product using parallel complex MULT within one complex MULT followed by matrix sum element wise with one ADD: $Ncycles = \alpha_{MULT} + \alpha_{ADD}$
7.	$\mathbf{\Delta}_k = \begin{bmatrix} \mathbf{\Gamma} & -\mathbf{y}_3 \\ -\mathbf{y}_3^H & c \end{bmatrix}$	$Total\ of\ Ncycles = 4\alpha_{MULT} + (2log_2(k-1)+1)\alpha_{ADD} + \alpha_{DIV}$
END FOR		
OUTPUT: $\mathbf{W} = \mathbf{\Delta}_K\mathbf{H}^H$		$Total\ of\ Ncycles = M\alpha_{MAC} + \sum_{k=2}^{K}(4\alpha_{MULT} + (2log_2(k-1)+1)\alpha_{ADD} + \alpha_{DIV})$

(c)

Fig. 6. Detection weight matrix computation (DWMC) core: (a) a $K \times K$ pipelined (systolic) array (showing the data flow for a multiplication of matrix $\mathbf{A}_{K\times M}$ with matrix $\mathbf{B}_{M\times K}$ for illustration) (b) with adder-tress (ladder) attached to a given row and (c) the data flow of RGMIU reusing the DWMC code for Gram matrix and performing the operations in Table 1.

[8] This expression is accurate if the number of UTs $K \cong 2^N$ where N is a positive integer.

TABLE 4. LATENCY ESTIMATION FOR ZF, NSE, GS AND RGMIU AS A FUNCTION OF $\alpha_{ADD}, \alpha_{MULT}, \alpha_{MAC}$ and α_{DIV} .

Method	Latency in the number of clock cycles
ZF	$M\alpha_{MAC} + 2K\alpha_{MULT} + \left(\frac{3K+2}{2}\log_2(K!) + 2K\log_2(K)\right)\alpha_{ADD} + \left(\frac{K(K-1)}{2} + 2K + 4\right)\alpha_{DIV}$
NSE (order 3)	$(M+K)\alpha_{MAC} + 2\alpha_{MULT} + 3\alpha_{ADD} + \alpha_{DIV}$
NSE (order 4)	$(M+2K)\alpha_{MAC} + 2\alpha_{MULT} + 4\alpha_{ADD} + \alpha_{DIV}$
GS (order 2)	$(M+K)\alpha_{MAC} + 2K\alpha_{MULT} + 2K\log_2(K)\alpha_{ADD} + 2K\alpha_{DIV} + 5$
GS (order 3)	$(M+K)\alpha_{MAC} + 3K\alpha_{MULT} + 3K\log_2(K)\alpha_{ADD} + 3K\alpha_{DIV} + 5$
RGMIU (proposed-full)	$M\alpha_{MAC} + 4(K-1)\alpha_{MULT} + \left(2\log_2((K-1)!) + (K-1)\right)\alpha_{ADD} + (K-1)\alpha_{DIV}$
RGMIU (proposed-update one UT)	$\left(\frac{M}{K} + 1\right)\alpha_{MAC} + 3\alpha_{MULT} + \left(2\log_2((K-1)) + 1\right)\alpha_{ADD} + \alpha_{DIV}$

the adder-tress where for RGMIU method, for instance, at iteration k, the latency through the adder-tress is $\log_2(k)$ as only k inputs are summed. When k spans from 1 to K the total latency is $\sum_{k=1}^{K}\log_2(k) = \log_2\left(\prod_{k=1}^{K}k\right) = \log_2(K!)$. The latency for a matrix inverse update requires the multiplication of a $K \times M$ matrix by a $M \times 1$ vector which can be performed with one row of the pipelined array. However, reusing all the pipelined rows, this reduces to performing K parallel $K \times M/K$ matrices by a $M/K \times 1$ vectors multiplications followed by K additions.

The DWMC core is implemented using System Generator(™) for DSP, from the Mathworks, to estimate the required FPGA resources (namely flip-flops (FFs), loop-up-tables (LUTs) and dedicated signal processing cores (DSP48s)). Table 5[9] shows the estimated resources for 4, 8, 16 and 32 UTs. As for the related works, Table 5 depicts the FPGA resources for NSE, GS, and OCD available for 8 users. The DWMC core is very area-efficient which allows the core to be instantiated as many times as possible depending on the available FPGA resources. It shall be noted that the core is designed as a co-processor to implement RGMIU, NSE, and GS as well. Notice that the DWMC core does not scale with the number of BS antennas (M). However, the number of antennas will contribute to the total latency to compute the detection weight matrix. Note that the number of DSP48s resources is $4K^2$.

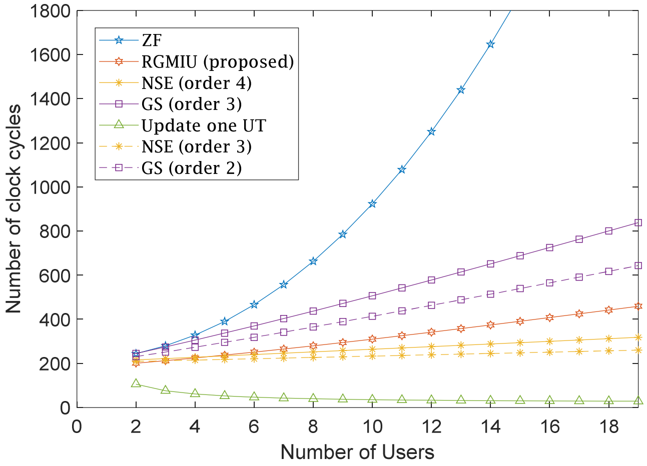


Fig. 7. Latency as a function of the number of UTs communication with a 64 antenna BS with $\alpha_{ADD} = 1, \alpha_{MULT} = 2, \alpha_{MAC} = 3, \alpha_{DIV} = 4$.

The latency is depicted in Fig. 7 with $\alpha_{ADD} = 1, \alpha_{MULT} = 2, \alpha_{MAC} = 3, \alpha_{DIV} = 4$. It is computed based on the full and efficient use of the DWMC core. As such, a reuse factor is computed based on the TDD-OFDM frame length (which is designed to cope with the channel coherence time). The reuse factor determines how often the DWMC core can be utilized to compute the detection weight matrices for other subcarriers[10].

In agreement with [11], GS is showing higher latency due to the data and processing dependencies (see Fig. 7). NSE remains the method with the lowest latency but at the cost of a large deviation from the optimal performance. RGMIU has a relatively lower latency and hence higher reuse factor at the benefit of no performance degradation.

Fig. 8 shows the required latency as a function of the minimum required SNR to achieve a user sum rate of 72b/s/Hz for a BS, equipped with 64 antennas, serving 12 UTs simultaneously. With a slight increase in latency NSE (order 4) can achieve substantial improvement compared to NSE (order 3). This is valid under favorable propagation channels and no HW impairments. It is not worth implementing GS (order 3)

TABLE 5. ESTIMATED FPGA RESOURCES FOR THE DWMC CORE FOR 4, 8, 16 AND 32 UTs.

Resources	DWMC Core size ($K \times K$)				NSE [9] ($K \times K$)	GS [10] ($K \times K$)	OCD [11] ($K \times K$)
	4×4	8×8	16×16	32×32	8×8	8×8	8×8
Slices	1557	6228	24912	99648	48 244	n.a.	11 094
FFs	1451	5803	23211	92843	161 934	15 864	43 008
LUTs	3024	12096	48384	193536	148 797	18 976	23 914
DSP48s	64	256	1024	4096	1016	232	774

[9] The BRAM resources are not shown here due to their low use ratio.

[10] Out of the scope of this work, it is interesting to investigate how one can leverage on weights interpolation instead of computing them on every subcarrier.

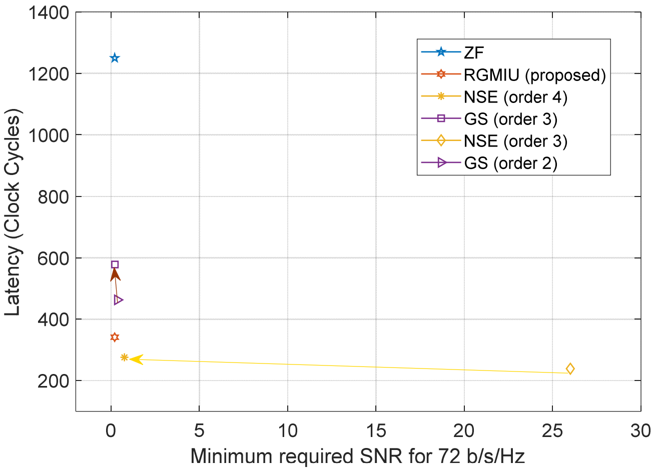


Fig. 8. Latency vs the minimum required SNR to achieve a user sum rate of 72b/s/Hz for a BS, equipped with 64 antennas, serving 12 UTs simultaneously.

given the slight performance improvement compared to GS (2). At midway latency between NSE (order 4) and GS (order 2), RGMIU methods show ZF-like performance at very slight latency increase with respect to NSE.

C. *Processing distribution and streaming requirements for real-time transmission*

The previous sections addressed a real data massive MIMO reference prototyping system with a host-based non-real time processing. This section discusses how to build a real-time transmission system for 64 antenna BS serving 12 single-antenna UTs. One challenge a system engineer faces is how signal processing can be efficiently partitioned among the computing nodes and determine the internode streaming requirements. Being a per antenna processing task, the TDD-OFDM processing is implemented at the radio head nodes (i.e. TitanMIMOs) whereas the multiuser detection is targeted on a central processing node such as Single KU115 Prodigy Logic Module[11], a Xilinx Kintex UltraScale FPGA based board developed by S2C. Therefore, eight TitanMIMOs are used as radio heads. Each radio head implements a 20MHz TDD-OFDM processing for 8x antenna streams [30]. As such each TitanMIMO streams in/out up to 8×1200 subcarriers per 70.83 µs. If each subcarrier is represented in 16-bit wordlength, this amounts to 4337 Mbit/s which fits within the TitanMIMO's 4× PCIe gen-1 lanes to stream to/from the central processing node.

Based on the FPGA estimated resources, DSP48s resources determine how many cores can be instantiated. The KU115 module's onboard FPGA has a total of 5520 DSP48s resources. Allocating up to 50% of the total resources (50% being reserved for channel estimation, downlink processing, and other processing) only four cores are instantiated[12]. These four cores are reused to compute the detection weights for 1200 subcarriers within 429 µs (i.e. right after receiving the uplink pilot and before starting downlink transmission). Based on the lowest FPGA speed grade, Table 6 shows the latency and reuse factor for DWMC core for 64× antenna BS serving 12 UTs[13]. To get 1200 detection weight matrices within one time-slot, a reuse factor of at least 300 is required.

Table 6. Latency and reuse factor for DWMC for 64 antenna BS serving 12 UTs

Method	Latency in clock cycles	Latency in usec	Reuse factor
NSE (order 3)	239	0.919	466
NSE (order 4)	276	1.062	403
GS (order 2)	462	1.777	241
GS (order 3)	578	2.223	192
ZF	1250	4.808	89
Proposed RGMIU	342	1.315	326

Some useful insights, related to detection techniques' complexity impact on designing a real-time transmission massive MIMO system, are derived:

- Only NSE[14] and RGMIU are favorable for a real-time implementation using a single KU115 Module. Recall from Section 3.C that NSE achieves lowest cell throughput rate.
- If a higher speed grade FPGA is used, the reuse factor can be increased by a factor of 1.29 (335MHz/260MHz) which enables to support GS (order 2). However, one shall raise a concern with respect to energy efficiency as this is also a key 5G parameter that needs to be given considerable attention [31].
- If the TDD frame structure is shortened by a factor of 2 to cope with high mobility UTs (ie the channel coherence time is rather 500 µs), one can envisage using a Quad KU115 Prodigy™ Logic Module. Each FPGA core processes 300 subcarriers at a time. The Quad KU115 module is suitable to cope with lower channel coherence time up to a factor of 4.

5. Conclusion

Being a disruptive 5G technology, massive MIMO has shown to provide a substantial improvement in spectral and energy efficiencies. However, to figure out how much of such gain can be harvested in real propagation channels and in the presence of hardware impairments and channel estimation errors, prototyping using an SDR platform is required. This enables us to discuss the real data performance of three different implementation methods for ZF-based receiver combining to investigate the performance loss using approximation techniques, such as NSE and GS, where more iterations are required especially in high load conditions. It is shown that the real data performance agrees with the simulation results, wherein in high load conditions, NSE (order 4) is showing relatively the poorest performance while GS (order 3) catches up using an extra iteration. As expected RGMIU is performing as well as ZF with direct matrix inversion. Based on LTE's EVM limits, interesting insights on maximizing per cell or per

[11] http://www.s2cinc.com/products/prodigy-logic-modules/kintex-ultrascale-prodigy-logic-modules/single-ku115-prodigy-logic-module

[12] 576 DSP48s are required for 12x12 DWMC core.

[13] The non-pipelined DSP48s maximum frequency is 260MHz and 335MHz for low and high speed grade Xilinx's Kintex UltraScale XCKU115 FPGA.

[14] For the sake of clarification; based on the DWMC core, data and processing flow dependencies determine the overall latency which explains why NSE (order 4) has lower latency compared to ZF (based on Cholesky decomposition).

user throughputs were derived. Nevertheless, when it comes to real-time transmission the implementation complexity in terms of operation counts is not sufficient to make an educated decision. As such, a framework for real-time implementation analysis is proposed. It relies on the re-use of a pipelined array to perform both Gram matrix computation and matrix inversion per NSE, GS, and RGMIU. With such approach, data and processing flow regularities and dependencies have dictated the expected real-time performance for these methods. To sum up RGMIU enables favorable real-time implementation at no performance degradation.

To complement the current work, one can use the proposed framework to investigate the channel estimation error effect using different pilot pattern/design schemes and channel estimation techniques. Similarly, over the air synchronization techniques can be implemented while scaling up the system dimensions in terms of the number of antennas at BS and the number of UTs. The least but not the last, HW impairments effects (including timing/synchronization mismatches) can be evaluated while analyzing the overall system's energy efficiency.

Acknowledgments

This work was supported in part by NUTAQ innovation and the Natural Sciences and Engineering Research Council of Canada.

References

[1] Björnson, E., Sanguinetti, L., Wymeersch, H., Hoydis, J., & Marzetta, T. L. (2019). Massive MIMO is a Reality – What is Next?: Five Promising Research Directions for Antenna Arrays. *Digital Signal Processing*, 94, 3–20.

[2] Björnson, E., (2018). A look at an LTE-TDD Massive MIMO product.
http://ma-mimo.ellintech.se/2018/08/27/.
Accessed 17 November 2019.

[3] von Butovitsch, P., Astely, D., Friberg, C., Furuskär, A., Göransson, B., Hogan, B., Karlsson, J., & Larsson, E., (2018). Advanced antenna systems for 5G networks. Ericsson white paper.
https://www.ericsson.com/4a8a87/assets/local/publications/white-papers/10201407_wp_advanced_antenna_system_nov18_181115.pdf.
Accessed 17 November 2019.

[4] Qu, Y., Lozano, A., & Gatherer, A., (2019). Nine Communications Technology Trends for 2019. Communication society technology news.
https://www.comsoc.org/publications/ctn/nine-communications-technology-trends-2019.
Accessed 17 November 2019.

[5] Shepard, C., Yu, H., Anand, N. et al. (2012). Argos: practical many-antenna base stations. *Annual International Conference on Mobile Computing and Networking*, New York, 53–64.

[6] Malkowsky, S. *et al*. (2017). The world's first real-time testbed for massive MIMO: design, implementation, and validation. *IEEE Access*, 5, 9073–9088.

[7] (2015). TitanMIMO-6: The sub-6 GHz 5G massive MIMO testbed. Product sheet. Nutaq Innovation.
https://www.nutaq.com/wp-content/uploads/2015/07/TitanMIMO6_09_16_2014_Final.pdf.
Accessed 17 November 2019.

[8] Ngo, H.Q. (2015). *Massive MIMO: fundamentals and system designs*, Ph.D. Thesis. Linköping University Electronic Press.

[9] Wu, M., Yin, B., Wang, G., Dick, C., Cavallaro, J.R., & Studer, C. (2014). Large-scale MIMO detection for 3GPP LTE: algorithms and FPGA implementations. *IEEE Journal of Selected Topics in Signal Processing*, 8(5), 916–929.

[10] Wu, Z., Zhang, C., Xue, Y., Xu, S., & You, Z. (2016). Efficient architecture for soft-output massive MIMO detection with Gauss-Seidel method. *IEEE International Symposium on Circuits and Systems*, Montreal, 1886–1889.

[11] Wu, M., Dick, C., Cavallaro, J.R., and Studer, C. (2016). High-throughput data detection for massive MU-MIMO-OFDM using coordinate descent. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(12), 2357–2367.

[12] Jeon, C., Li, K., Cavallaro, J.R., & Studer, C. (2019). Decentralized Equalization with Feedforward Architectures for Massive MU-MIMO. *IEEE Transactions on Signal Processing*, 67(17), 4418–4432.

[13] Ahmed Ouameur, M. & Massicotte, D. (2019). Efficient Distributed Processing for Large Scale MIMO Detection. *European Signal Processing Conference (Eusipco)*, A Coruna, Spain, 2-6 Sept., 2019, 1–4.

[14] Ahmed Ouameur, M., & Massicotte, D., (2019). Deep Autoencoder for Interconnect's Bandwidth Relaxation in Large Scale MIMO-OFDM Processing.
https://arxiv.org/abs/1907.12613.
Accessed 17 November 2019.

[15] Rosário, F., Monteiro, F.A., & Rodrigues, A. (2016). Fast matrix inversion updates for massive MIMO detection and precoding. *IEEE Signal Processing Letters*, 23(1), 75–79.

[16] Khan, M.E., (2008). *Updating inverse of a matrix when a column is added/removed*. Technical Report, Computer Science of University of British Columbia, 3 pages.

[17] Björnson, E., Bengtsson, M., & Ottersten, B. (2014). Optimal multiuser transmit beamforming: a difficult problem with a simple solution structure. *IEEE Signal Processing Magazine*, 31(4), 142–148.

[18] Zhang, R., Ai, B., Yang, L., Song, H., & Li, Z.Q., (2014). A precoding and detection scheme for OFDM based wireless communication system in high-speed environment. *IEEE Transactions on Consumer Electronics*, 60(4), 558-566.

[19] Björnson, E., Hoydis, J., Kountouris, M., & Debbah, M., (2014). Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits. *IEEE Transactions on Information Theory*, 60(11), 7112–7139.

[20] Mi, D., Dianati, M., Zhang, L., Muhaidat, S., & Tafazolli, R., (2017). Massive MIMO Performance With Imperfect Channel Reciprocity and Channel Estimation Error. *IEEE Transactions on Communications*, 65(9), pp. 3734–3749.

[21] Björnson, E., Larsson, E.G., & Marzetta, T.L., (2016). Massive MIMO: ten myths and one critical question. *IEEE Communications Magazine*, 54(2), 114–123.

[22] Shafik, R.A., Rahman, M.S., & Islam, A.R., (2006). On the extended relationships among EVM, BER and SNR as performance metrics. *International Conference on Electrical and Computer Engineering*, Dhaka, 408–411.

[23] Yang, X., et al. (2017). Design and implementation of a TDD-based 128-antenna massive MIMO prototype system, *China Communications*, 14(12), 162–187.

[24] Harris, P., *et al*. (2016). LOS Throughput measurements in real-time with a 128-antenna massive MIMO testbed. *IEEE Global Communications Conference*, Washington, DC, 1–7.

[25] Ahmed-Ouameur, M., Massicotte, D., & Zhu, W., (2015). Carrier frequency and sampling rate offsets effect on sub 6 GHz Massive MIMO. *Nordic Circuits and Systems Conference (NORCAS)*, Oslo, 1–4.

[26] Ma, X., Gao, Q., Wang, J., Marojevic, V., & Reed, J.H., (2017). Dynamic sounding for multi-user MIMO in wireless LANs. *IEEE Transactions on Consumer Electronics*, 63(2), 135–144.

[27] (2015). 2nd Gen. PicoSDR, 2nd generation 70-6000MHz SDR development platform. Nutaq Innovation.
https://www.nutaq.com/picosdr8x8.
Accessed 17 November 2019.

[28] (2019). Xilinx Zynq-7000 All Programmable SoC ZC706 Evaluation Kit, Xilinx
https://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html.
Accessed 17 November 2019.

[29] Björnson, E., Matthaiou, M., & Debbah, M., (2015). Massive MIMO with non-ideal arbitrary arrays: hardware scaling laws and circuit-aware design. *IEEE Transactions on Wireless Communications*, 14(8), 4353–4368.

[30] (2015). Nutaq OFDM Reference Design: FPGA based SISO/MIMO PHY transceiver. Nutaq Innovation.
https://www.nutaq.com/sites/default/files/ofdm-wireless-lowres.pdf.
Accessed 17 November 2019.

[31] (2015). IMT vision - framework and overall objectives of the future development of IMT for 2020 and beyond. ITU-R Working Party WP 5D: Draft New Recommendation. Doc. R12-SG05-C-0199.
https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf.

Accessed 17 November 2019.

APPENDIX A

To make the paper self-contained this appendix derives the matrix inversion of a large $K \times K$ matrix of the form $\mathbf{H}^H \mathbf{H}$. We propose to apply the matrix inversion update lemma when a new column is added [16]. Assume we have the inverse of a $(K-1)\times(K-1)$ matrix $\boldsymbol{\Delta}_{K-1}^{-1} = \mathbf{H}_{1:\mathrm{K}-1}^{H}\mathbf{H}_{1:\mathrm{K}-1}$ (note that we have adopted the Matlab index notation $1:K-1$ in $\mathbf{H}_{1:\mathrm{K}-1}$ to designate columns 1 to $K-1$ of $\mathbf{H}$). Therefore, the inverse of a $K \times K$ matrix $\boldsymbol{\Delta}_{K}^{-1} = \mathbf{H}_{1:\mathrm{K}}^{H}\mathbf{H}_{1:\mathrm{K}}$ can be computed as follow

$$\begin{aligned}
\boldsymbol{\Delta}_K &= \left(\mathbf{H}_{1:\mathrm{K}}^{H}\mathbf{H}_{1:\mathrm{K}}\right)^{-1} \\
&= \left(\begin{bmatrix} \mathbf{H}_{1:\mathrm{K}-1}^{H} \\ \mathbf{H}_{\mathrm{K}:\mathrm{K}}^{H} \end{bmatrix}\begin{bmatrix} \mathbf{H}_{1:\mathrm{K}-1} & \mathbf{H}_{\mathrm{K}:\mathrm{K}} \end{bmatrix}\right)^{-1} \\
&= \begin{bmatrix} \mathbf{H}_{1:\mathrm{K}-1}^{H}\mathbf{H}_{1:\mathrm{K}-1} & \mathbf{H}_{1:\mathrm{K}-1}^{H}\mathbf{H}_{\mathrm{K}:\mathrm{K}} \\ \mathbf{H}_{\mathrm{K}:\mathrm{K}}^{H}\mathbf{H}_{1:\mathrm{K}-1} & \mathbf{H}_{\mathrm{K}:\mathrm{K}}^{H}\mathbf{H}_{\mathrm{K}:\mathrm{K}} \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \boldsymbol{\Gamma} & -c\boldsymbol{\Delta}_{K-1}^{H}\mathbf{H}_{1:\mathrm{K}-1}^{H}\mathbf{H}_{\mathrm{K}:\mathrm{K}} \\ -c\mathbf{H}_{\mathrm{K}:\mathrm{K}}^{H}\mathbf{H}_{1:\mathrm{K}-1}\boldsymbol{\Delta}_{K-1}^{H} & c \end{bmatrix}
\end{aligned} \tag{A.1}$$

where $c = \dfrac{1}{\left(\mathbf{H}_{\mathrm{K}:\mathrm{K}}^{H}\mathbf{H}_{\mathrm{K}:\mathrm{K}}\right) - \left(\mathbf{H}_{1:\mathrm{K}-1}^{H}\mathbf{H}_{\mathrm{K}:\mathrm{K}}\right)^{H}\boldsymbol{\Delta}_{K-1}\left(\mathbf{H}_{1:\mathrm{K}-1}^{H}\mathbf{H}_{\mathrm{K}:\mathrm{K}}\right)}$ and

$\boldsymbol{\Gamma} = \boldsymbol{\Delta}_{K-1} + c\boldsymbol{\Delta}_{K-1}\left(\mathbf{H}_{\mathrm{K}:\mathrm{K}}^{H}\mathbf{H}_{1:\mathrm{K}-1}\right)^{H}\left(\mathbf{H}_{\mathrm{K}:\mathrm{K}}^{H}\mathbf{H}_{1:\mathrm{K}-1}\right)\boldsymbol{\Delta}_{K-1}^{H}$.

If we set $\mathbf{z} \triangleq \mathbf{H}_{\mathrm{K}:\mathrm{K}}$, $\mathbf{y}_1 \triangleq \mathbf{H}_{1:\mathrm{K}-1}^{H}\mathbf{z}$, $\mathbf{y}_2 \triangleq \boldsymbol{\Delta}_{K-1}\mathbf{y}_1$, and $\mathbf{y}_3 \triangleq c\,\mathbf{y}_2$ then $c = 1 \big/ \left(\mathbf{z}^H\mathbf{z} - \mathbf{y}_1^H\mathbf{y}_2\right)$ and $\boldsymbol{\Gamma} = \boldsymbol{\Delta}_{K-1} + c\mathbf{y}_2\mathbf{y}_2^H$.

Applying equation (A.1) *successively* from the second column all the way to the last column K, the algorithm, dubbed recursive gram matrix inversion update (RGMIU), is outlined in Table 1.