

A Proximal Jacobian ADMM Approach for Fast Massive MIMO Signal Detection in Low-Latency Communications

*Anis Elgabri, Ali Elghariani, Vaneet Aggarwal, *Mehdi Bennis, and Mark R. Bell
Purdue University, West Lafayette IN, USA, * University of Oulu, Finland

Abstract—One of the 5G promises is to provide Ultra Reliable Low Latency Communications (URLLC) which targets an end to end communication latency that is $\leq 1\text{ms}$. The very low latency requirement of URLLC entails a lot of work in all networking layers. In this paper, we focus on the physical layer, and in particular, we propose a novel formulation of the massive MIMO uplink detection problem. We introduce an objective function that is a sum of strictly convex and separable functions based on decomposing the received vector into multiple vectors. Each vector represents the contribution of one of the transmitted symbols in the received vector. Proximal Jacobian Alternating Direction Method of Multipliers (PJADMM) is used to solve the new formulated problem in an iterative manner where at every iteration all variables are updated in parallel and in a closed form expression. The proposed algorithm provides a lower complexity and much faster processing time compared to the conventional MMSE detection technique and other iterative-based techniques, especially when the number of single antenna users is close to the number of base station (BS) antennas. This improvement is obtained without any matrix inversion. Simulation results demonstrate the efficacy of the proposed algorithm in reducing detection processing time in the multi-user uplink massive MIMO setting.

Index Terms—Masive MIMO, Proximal Jacobian ADMM, Computational Complexity, Parallel Processing

I. INTRODUCTION

Massive multiple-input multiple-output (MIMO) is one of the most promising techniques in 5G networks due to its potential for significant rate enhancement and energy efficiency [1], [2]. An important requirement for the emerging URLLC 5G services is to provide low latency and fast processing time for these services [3], especially in the multi-user massive MIMO scenario, in which the number of users is greatly increasing [4]. The challenge of providing the low latency and fast processing is approached in the literature through several aspects and at different network layers [5], [6]. For example, in [7] edge caching and high computation edge nodes are discussed as ways to reduce latency. In physical layer, and in particular in the resource allocation problem, Non Orthogonal Multiple Access (NOMA) in combination with grant-free can be used in URLLC [8]. On the other hand, for a waveform design problem [9] presents a choice of suitable modulation and coding schemes, and discuss the impact of different waveform candidates. In this paper, we focus on the signal detection component of the receiver, in

which we propose a fast signal detection algorithm based on parallel implementation.

Reducing the computational complexity of the detection algorithm is very important since the less the number of computations the algorithm needs, the faster processing time it requires. Thus, from this perspective, to reduce the computational complexity, linear MMSE detection scheme is widely considered, but this scheme involves matrix inversion which is computationally costly. A truncated Neumann series expansion can be used to approximate the matrix inversion [10]. With sufficient large truncation order, the approximation can be very close to real matrix inversion only in cases when the ratio between the number of BS antennas and the number of single antenna users is large (e.g. ≥ 16). However, [10] indicates that when the ratio becomes smaller, a large truncation order is required which makes the computational complexity more than the exact matrix inversion operations. Similar other ideas are presented in [11] focusing on how to reduce the complexity of the matrix inversion, but they suffer from the deficiency of deteriorated performance and increased computations as the number of single antenna users increases [11].

Another approach is based on iterative methods to reduce computational complexity. The iterations are implemented to transform the matrix inversion problem of MMSE matrix into solving linear equations [12], [13]. The iterative methods can also be used in a context other than finding the inversion of the MMSE matrix, such as [14], [15], where the detection problem is formulated as a convex optimization problem that can be solved in an iterative manner using alternating minimization techniques or quadratic programming techniques.

The above two approaches experience long waiting delay, especially in the case of a large number of users. For instance, the information symbol belongs to the last user in the received signal vector needs to wait until all previous symbols pertaining to all other users are detected [12]. This successive detection manner makes the detection scheme time inefficient. Besides, this is not efficient for hardware implementation [16].

Due to the increased interest for dealing with big data and large scale problems such as the problem at hand, parallel and distributed computational methods are highly desirable for faster processing time. Alternating Direction Method of Multipliers (ADMM), as a versatile algorithmic tool, has proven to be very effective at solving many large-scale problems and

well suited for distributed computing [17]. In this paper, we, first, we use our propose a novel formulation of the uplink massive MIMO detection problem in [18] in such a way that it fits the ADMM formulation. In particular, we introduce an objective function that is a sum of strictly convex and separable functions based on decomposing the received vector into multiple vectors. Each vector represents the contribution of one of the transmitted symbols in the received vector. Second, we use the Proximal Jacobian Alternating Direction Method of Multipliers (PJADMM) [17] to solve the new formulated problem in an iterative manner such that in every iteration all variables are updated in parallel and in a closed form expression, without requiring any matrix inversion. Therefore, at every time instance, information symbols of all users are updated in parallel at the receiver. This in turn provides much faster processing time compared to the conventional MMSE detection techniques, and also compared to the other iterative-based techniques, especially when the number of users (each with a single antenna) is close to the number of base station (BS) antennas. Note, unlike the existing techniques such as in [16], our proposed algorithm does not try to parallelize existing MMSE detection problem, instead it develops the parallel implementation of the detection problem based on a unique formulation of the maximum likelihood optimization problem combined with the application of PJADMM technique. In the end, our objective is to reach the exact MMSE performance but with a faster processing time. Simulation results demonstrate the efficacy of the proposed algorithm in the multi-user uplink massive MIMO setting.

The remainder of this paper is organized as follows. In Section II, the system model and problem formulation are described. Section III contains the proposed algorithm, and section IV presents simulation results, and finally the paper is concluded with a short summary.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

Consider the uplink data detection in a multi-user massive MIMO system with N_r BS antennas and N_t users each with a single antenna. The vector $\tilde{\mathbf{x}} = (x_1, x_2, \dots, x_{N_t})^T \in \mathbb{C}^{N_t \times 1}$ represents the complex transmitted signal, where x_k is the transmitted symbol for user k with $E|x_i|^2 = 1, \forall i$. Each user transmits symbols over a flat fading channels and the signals are demodulated and sampled at the receiver. The vector $\tilde{\mathbf{y}} = (y_1, y_2, \dots, y_{N_r})^T \in \mathbb{C}^{N_r \times 1}$ represents the complex received signal. The channel matrix $\tilde{\mathbf{H}} \in \mathbb{C}^{N_r \times N_t}$ can be represented as $(h_1, h_2, \dots, h_{N_t})$, where $h_i = (h_{1,i}, h_{2,i}, \dots, h_{N_r,i})^T \in \mathbb{C}^{N_r \times 1}$, and $h_{m,n}$ is the complex flat fading channel gain from transmit antenna n to the receive antenna m , with $h_{m,n} \sim \mathcal{CN}(0, 1)$. The system can be modeled as:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{v}} \quad (1)$$

where $\tilde{\mathbf{v}} = (v_1, v_2, \dots, v_{N_r})^T \in \mathbb{C}^{N_r \times N_t}$ is the complex additive white Gaussian noise (AWGN) vector whose elements are mutually independent with zero mean and variance σ_v^2 . The corresponding real-valued system model is $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$ [19].

The equivalent ML detection problem of the real model can be written in the form $\hat{\mathbf{x}} = \underset{\mathbf{x} \in \chi^{2N_t}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$, where $\chi = \frac{1}{\sqrt{M}}\{-\sqrt{M} + 1, \dots, -1, 1, \dots, \sqrt{M} - 1\}$, M is the QAM constellation size, and $\frac{1}{\sqrt{M}}$ the normalization factor.

B. Problem Formulation

First, we decompose the received vector \mathbf{y} into a linear combination of vectors so that $\mathbf{y} = \sum_{i=1}^{2N_t} \mathbf{y}_i$, where \mathbf{y}_i represents the contribution of the i -th transmitted symbol in the received vector. The element wise representation of the decomposed received vector is:

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(2N_r)} \end{bmatrix} = \begin{bmatrix} y_1^{(1)} \\ y_1^{(2)} \\ \vdots \\ y_1^{(2N_r)} \end{bmatrix} + \dots + \begin{bmatrix} y_i^{(1)} \\ y_i^{(2)} \\ \vdots \\ y_i^{(2N_r)} \end{bmatrix} + \dots + \begin{bmatrix} y_{2N_t}^{(1)} \\ y_{2N_t}^{(2)} \\ \vdots \\ y_{2N_t}^{(2N_r)} \end{bmatrix}$$

The k -th element of \mathbf{y} can be represented as $y^{(k)} = \sum_{i=1}^{2N_t} y_i^{(k)}$, $k = 1, \dots, 2N_r$. Let \mathbf{h}_i the i^{th} column of the real channel matrix \mathbf{H} . Now, we relax the non-convexity constraint on the feasible set χ , and approximate the ML problem based on the above decomposition as follows:

$$\min_{x_i, y_i} \sum_{i=1}^{2N_t} \|\mathbf{y}_i - \mathbf{h}_i x_i\|_2^2 \quad (2)$$

subject to

$$\sum_{i=1}^{2N_t} y_i^{(k)} = y^{(k)}, \forall k = 1, \dots, 2N_r \quad (3)$$

$$-l \leq x_i \leq l, \forall i = 1, \dots, 2N_t \quad (4)$$

where $l = \frac{1}{\sqrt{M}}(\sqrt{M} - 1)$. The objective function in (2) is a sum of separable terms, each of which is a function of only one symbol and its contribution in the received vector. In the next section, we use AltMin to solve the proposed formulation.

C. K.K.T Conditions

The problem in (2)-(4) is a convex optimization problem with linear constraints, so the K.K.T. conditions are sufficient and necessary for the optimal solution. The Lagrangian function for (2)-(4) is defined as follows:

$$\begin{aligned} \mathcal{L} = & \sum_{i=1}^{2N_t} \|\mathbf{y}_i - \mathbf{h}_i x_i\|_2^2 + \sum_{k=1}^{2N_r} \lambda^k (y^{(k)} - \sum_{i=1}^{2N_t} y_i^{(k)}) \\ & + \sum_{i=1}^{2N_t} (\mu_1^{(i)}(l - x_i) + \mu_2^{(i)}(l + x_i)) \end{aligned} \quad (5)$$

Then, the following K.K.T. conditions, which are sufficient and necessary for the optimal solution to the convex optimization problem in (2)-(4), are obtained from the Lagrangian function:

$$y_i^{(k)} = h_i^{(k)} x_i + \lambda^{(k)}/2, \forall i, k \quad (6)$$

$$\lambda^{(k)} = \frac{1}{N_t} (y^{(k)} - \sum_{i=1}^{2N_t} h_i^{(k)} x_i), \forall i, k \quad (7)$$

$$2x_i \sum_{k=1}^{2N_t} h_i^{(k)2} - 2 \sum_{k=1}^{2N_t} y_i^{(k)} h_i^{(k)} - \mu_1^{(i)} + \mu_2^{(i)} = 0, \forall i, k \quad (8)$$

$$\mu_1^{(i)} (l - x_i) = 0, \forall i \quad (9)$$

$$\mu_2^{(i)} (l + x_i) = 0, \forall i \quad (10)$$

In order to solve K.K.T. conditions to find the optimal solution in a closed form expression, high complexity computations are required. Therefore, in the next section, we develop our iterative algorithm based on the PJADMM in which all variables are updated in a closed form expression per iteration. That is, transmitted symbols of different users and their corresponding contributions to the received vector are updated in parallel.

III. PROPOSED ALGORITHM BASED ON PROXIMAL JACOBIAN ADMM

In Proximal Jacobian ADMM all blocks of the variables are updated in parallel. Proximal Jacobian ADMM was shown to enjoy a global convergence with convergence rate of $o(1/t)$ [17]. One difference of Proximal Jacobian compared to Gauss Seidel ADMM is that the term $\frac{\tau}{2} \sum_{i=1}^{2N_t} (x_i - x_i^{(t-1)})^2 + \frac{\tau}{2} \|\mathbf{y}_i - \mathbf{y}_i^{(t-1)}\|_2^2$ is added in the augmented Lagrangian. Hence, at every iteration, the primal variables, $\mathbf{y}_i, x_i \forall i$, are updated as follows:

$$x_i, \mathbf{y}_i = \underset{x_i, \mathbf{y}_i}{\operatorname{argmin}} \mathcal{L}_{\rho, \tau} (x_i, \mathbf{y}_i, (x_j^{(t-1)}, \mathbf{y}_j^{(t-1)}) \forall j \neq i, \lambda^{(t-1)}) \quad (11)$$

In order to update \mathbf{x} , the following optimization problem is solved.

$$\begin{aligned} x_i, \mathbf{y}_i = \underset{x_i, \mathbf{y}_i}{\operatorname{argmin}} & \| \mathbf{y}_i - \mathbf{h}_i x_i \|_2^2 - \sum_{k=1}^{2N_r} \lambda^{k(t-1)} y_i^{(k)} \\ & + \frac{\rho}{2} \sum_{k=1}^{2N_r} (y^{(k)} - y_i^{(k)} - \sum_{j=1, j \neq i}^{2N_t} y_j^{(k)(t-1)})^2 \\ & + \frac{\tau}{2} (x_i - x_i^{(t-1)})^2 + \frac{\tau}{2} \sum_{k=1}^{2N_r} (y_i^{(k)} - y_i^{(k)(t-1)})^2 \end{aligned} \quad (12)$$

subject to

$$-l \leq x_i \leq l \quad (13)$$

To solve this problem, we first write the corresponding Lagrangian function of (12)-(13) which is:

$$\text{eq}(12) + \mu_1^{(i)} (l - x_i) + \mu_2^{(i)} (l + x_i) \quad (14)$$

Now, we take the derivative of (14) with respect to variables x_i, \mathbf{y}_i and set it to zero, for different combinations of $\mu_1^{(i)}$ and $\mu_2^{(i)}$, we get:

- $\mu_1^{(i)} = 0$, and $\mu_2^{(i)} = 0$. Therefore,

$$\begin{aligned} x_i = & \frac{\frac{1}{2+\rho+\tau} \sum_{k=1}^{2N_r} h_i^{(k)} \lambda^{(k)(t-1)}}{(1 - \frac{2}{2+\rho+\tau}) \sum_{k=1}^{2N_r} h_i^{(k)2} - \tau} \\ & + \frac{\frac{\rho}{2+\rho+\tau} \sum_{k=1}^{2N_r} h_i^{(k)} (y^{(k)} - \sum_{j=1, j \neq i}^{2N_t} y_j^{(k)(t-1)})}{(1 - \frac{2}{2+\rho+\tau}) \sum_{k=1}^{2N_r} h_i^{(k)2} - \tau} \\ & + \frac{\frac{\tau}{2+\rho+\tau} \sum_{k=1}^{2N_r} h_i^{(k)} y_i^{(k)(t-1)} - \tau x_i^{(t-1)}}{(1 - \frac{2}{2+\rho+\tau}) \sum_{k=1}^{2N_r} h_i^{(k)2} - \tau} \end{aligned} \quad (15)$$

- $\mu_1^{(i)} = 0$, and $\mu_2^{(i)} \neq 0$. Therefore,

$$x_i = -l \quad (16)$$

- $\mu_1^{(i)} \neq 0$, and $\mu_2^{(i)} = 0$. Therefore,

$$x_i = l \quad (17)$$

There is still one more case in which $\mu_1^{(i)}$ and $\mu_2^{(i)}$ both are not equal to zero, but it is void because x_i cannot be equal to 1 and -1 at the same time. Next, the update of \mathbf{y}_i is as follows:

$$\begin{aligned} y_i^{(k)} = & \frac{1}{2 + \rho + \tau} \left(2h_i^{(k)} x_i^{(t)} + \lambda^{(k)(t-1)} \right. \\ & \left. + \rho (y^{(k)} - \sum_{j=1}^{2N_t} y_{j \neq i}^{(k)(t-1)}) + \tau y_i^{(k)(t-1)} \right), \forall k \end{aligned} \quad (18)$$

The ideal case is to choose x_i based on the above combinations of $\mu_1^{(i)}$ and $\mu_2^{(i)}$ such that it minimizes the objective function (12). However, since we perform hard quantization for detecting symbols, Eq(15) is enough. Finally, each dual variable $\lambda^{(k)} \forall k$ is updated as follows:

$$\lambda^{(k)(t)} = \lambda^{(k)(t-1)} + \rho (y^{(k)(t)} - \sum_{i=1}^{2N_t} y_i^{(k)(t)}), \forall k \quad (19)$$

The PJADMM algorithm is outlined in Algorithm 1. It is clear from (15)-(19) that at every iteration, t , the element $x_i^{(t)}$ depends on $\mathbf{y}_i^{(t)}, x_i^{(t-1)}, x_j^{(t-1)}, \forall j \neq i, \mathbf{y}_j^{(t-1)}, \forall j \neq i$ and $\lambda^{(k)(t-1)} \forall k$. Therefore, it depends on the symbols' values of the previous iteration and their contribution to the received vector, and not on the current iteration values. The same is also true for the updates of \mathbf{y}_i , and $\lambda^{(k)}$. This facilitates the implementation of the parallel processing at the receiver per every iteration, as shown in Algorithm 1

Note that we do not discuss the optimality of the proposed algorithm, in this paper, because the optimality of PJADMM was proven in [17] when the objective function is a sum of strictly convex separable functions, which is the case in our

Algorithm 1 Proximal Jacobian ADMM

- 1: **Input:** $\mathbf{H}, \mathbf{y}, \rho, \tau$
 - 2: **Output:** $x_i, \mathbf{y}_i, \forall i$
 - 3: **Initialization:** $x_i = 0, \mathbf{y}_i = \mathbf{0}, \forall i, \lambda^k = 0, \forall k$
 - 4: $\delta =$ convergence tolerance
 - 5: $T =$ Maximum number of iterations
 - 6: **Proximal Jacobian ADMM:**
 - 7: **repeat**
 - 8: $t \leftarrow t + 1$
 - 9: **Parallel Processing**
 - 10: use (15) to update $\mathbf{x}_i \forall i$
 - 11: use (18) to update $\mathbf{y}_i \forall i$
 - 12: use (19) to update $\lambda^{(k)}, \forall k$
 - 13: $V^{(t)} = \sum_{i=1}^{2N_t} \|\mathbf{y}_i - \mathbf{h}_i \mathbf{x}_i\|_2^2$
 - 14: **until** $|V^{(t)} - V^{(t-1)}| < \delta$ **OR** $t > T$
-

objective function.

Complexity analysis of the proposed algorithm

We evaluate the computational complexity of the proposed algorithm based on the number of multiplication operations needed to detect the transmitted symbol vector. We adopt the number of real-valued multiplications for the analysis of the computational complexity as in [13], and [11]. The overall computational complexity of the PJADMM based algorithm consists of two parts: the first part, which is independent of the number of iterations, is needed to be performed once such as the multiplications operations of $\sum_{k=1}^{2N_r} h_i^{(k)}$, and the second part, which is iteration dependent, is need to be repeated in every iteration, such as the multiplication operation of $\sum_{k=1}^{2N_r} h_i^{(k)} y_i^{(k)(t-1)}$. There is no computations required to perform initialization as the initial values of x_i and \mathbf{y} can be all zeros.

Based on (15)-(19), the iteration independent computations requires $4N_r$ real-valued multiplications, and the iteration dependent requires $T(14N_r + 2N_t)$ real-valued multiplications. As it is shown in the next section, the number of iteration, T , depends on the number of uplink user antennas, N_t . Therefore, based on the parallel implementation of PJ ADMM, the number of time units required to process the detection of one element \hat{x}_i is:

$$\# \text{of time units} = 4N_r + T(14N_r + 2N_t) \quad (20)$$

and since all elements of the received vector $(x_i, \forall i)$ are processed in parallel, the total processing time for detecting one received vector is the same as that needed for detecting one element.

IV. NUMERICAL RESULTS

To evaluate the performance of the proposed algorithm we implemented several simulation experiments for the uplink massive MIMO system in a block flat fading channel. We assume perfect knowledge of the channel state information at the receiver and uncoded QPSK modulations for demonstration, however the proposed algorithm can be extended for higher QAM modulations with coded cases. The aim of the

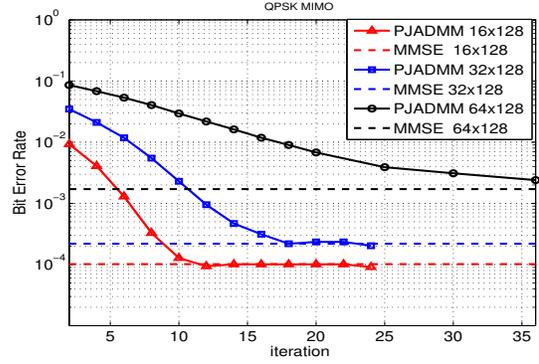


Fig. 1: BER performance versus # of Iterations of the PJADMM based Algorithm for different massive MIMO configurations at $\text{SNR} = 12$ dB

simulation in this section is to show that the bit error rate (BER) performance of the proposed algorithm can achieve the performance of the exact MMSE technique (which is the benchmark [20]) at various number of users with faster processing time.

Regarding a comparison with other state of the art work, it is important to note that, all previous works such as [10], [12], [13], [16], assume the MMSE matrix ($\mathbf{W} = \mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I}_{N_t}$) is a diagonal dominant matrix. This can be an acceptable assumption when the number of BS antennas is an order of magnitude more than the number of users. Therefore, the proposed algorithm in this paper is a versatile in the sense that it can provide the same performance of the exact MMSE performance at any number of users, in addition, to providing fast processing time.

In the first simulation experiment, we studied the BER performance of our algorithm compared to the MMSE at various number of iterations and at fixed SNR value. The number of iterations T changes with a step size 2. The BER of various massive MIMO configurations are studied at $\text{SNR} = 12$ dB. Fig. 1 shows BER performance versus maximum number of iterations for the following MIMO configurations $16 \times 128, 32 \times 128, 64 \times 128$. The MMSE performance is also shown in the same figure to examine the number of iterations required by the algorithm to reach MMSE performance. It can be seen from the figure that the larger the ratio between N_r and N_t the smaller the number of iterations required. The 16×128 configuration requires 12, while the 64×128 configuration requires 40 iterations to reach MMSE performance.

Next, we use the number of iterations obtained in the pervious simulation to study the BER performance of our proposed algorithm compared to the exact MMSE performance at various SNR values. Similar to the configuration above, the number of BS antennas is kept at 128, while the number of single antenna users can vary such that $N_t \leq N_r$, as depicted in Fig. 2. It can be clearly observed that the performance of our proposed algorithm matches that of the AltMin technique [14] and the exact MMSE. Note, although the AltMin algorithm requires less iterations in each MIMO configuration,

V. CONCLUSION AND FUTURE WORK

This paper presented a novel reformulation for the ML problem in such a way that it suits PJADMM optimization tool. The PJADMM technique is used to solve the new formulated problem in an iterative manner such that in every iteration all variables are updated in parallel and in a closed form expression, without a need for any matrix inversion. This provides much faster processing time compared to the exact MMSE and the other iterative-based techniques, especially when the number of single antenna users is close to the number of base station (BS) antennas.

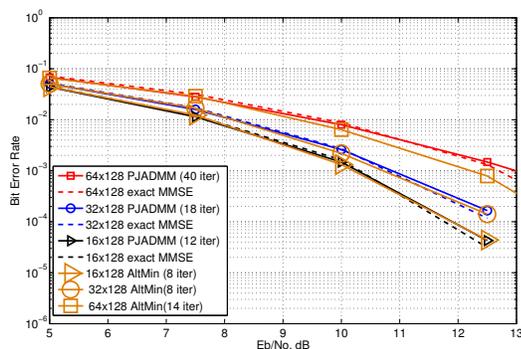


Fig. 2: BER performance of PJADMM based Algorithm at various massive MIMO configurations

the proposed algorithm has the advantage of parallel detection implementations, i.e. it can provide lower latency.

The next simulation experiment is used to emphasize the processing time of the proposed algorithm with its parallel implementation compared to the MMSE bench mark and also compared to a previous iterative techniques such as [14]. The comparison is made in terms of the number of time units required to process a one received vector. We define the number of time units as the number of computations required, which is basically a function of the number of real-valued multiplication operations and also the number of iterations. The comparison is also based on the fact that all techniques have the same BER performance. Note, the aim of this paper is not to compare the performance of the proposed algorithm with other techniques such as those in [11]-[13] because we prove that it achieves the exact MMSE performance as depicted in Fig. 2. However, we aim to provide a parallel implementation of the detection process using PJADMM that was not done in [11]-[13]. Table I, shows clearly how the proposed algorithm requires much less processing time compared to both the exact MMSE technique and the AltMin technique that has a successive symbol detection implementation. At a smaller number of users (i.e. the ratio between N_r and N_t is large) such as 16, the PJADMM based algorithm can provide a processing time that is two times faster than the exact MMSE and 10 times faster than AltMin. As N_t increases with fixed N_r , the advantage of the proposed algorithm is of orders of magnitude faster. For example, at $N_t = 64$, it performs 18 times faster than AltMin and 28 times faster than exact MMSE.

TABLE I: The number of time units ($\times 10^4$) required to process one received vector at $\text{SNR} = 12$ dB

	$N_t=16$	$N_t=32$	$N_t=64$	$N_t=128$
MMSE	5.7	31.1	219.5	1697
AltMin[14]	20	40.9	140.9	281.8
PJADMM	2.24	3.39	7.73	10.3

REFERENCES

- [1] F. Boccardi, R. W. Heath Jr, A. Lozano, T. L. Marzetta, and P. Popovski, "Five Disruptive Technology Directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.
- [2] T. Van Chien and E. Björnson, "Massive mimo communications," in *5G Mobile Communications*. Springer, 2017, pp. 77–116.
- [3] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk and scale," *arXiv preprint arXiv:1801.01270*, 2018.
- [4] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, "5g urllc: Design challenges and system concepts," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2018, pp. 1–6.
- [5] A. Anand and G. de Veciana, "Resource allocation and harq optimization for urllc traffic in 5g wireless networks," *arXiv preprint arXiv:1804.09201*, 2018.
- [6] A. Shapin, K. Kittichokechar, N. Andgart, M. Sundberg, and G. Wikström, "Physical layer performance for low latency and high reliability in 5g," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2018, pp. 1–6.
- [7] E. Baştuğ, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *arXiv preprint arXiv:1405.5974*, 2014.
- [8] L. Tian, C. Yan, W. Li, Z. Yuan, W. Cao, and Y. Yuan, "On uplink non-orthogonal multiple access for 5g: opportunities and challenges," *China Communications*, vol. 14, no. 12, pp. 142–152, 2017.
- [9] S. A. Ashraf, Y.-P. E. Wang, S. Eldessoki, B. Holfeld, D. Parruca, M. Serror, and J. Gross, "From radio design to system evaluations for ultra-reliable and low-latency communication," in *European Wireless Conference 2017 (EW 2017)*, 2017.
- [10] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale mimo detection for 3gpp lte: Algorithms and fpga implementations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 916–929, 2014.
- [11] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, "Conjugate gradient-based soft-output detection and precoding in massive mimo systems," in *Global communications conference (GLOBECOM), 2014 IEEE*. IEEE, 2014, pp. 3696–3701.
- [12] F. Jiang, C. Li, and Z. Gong, "A low complexity soft-output data detection scheme based on jacobi method for massive mimo uplink transmission," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–5.
- [13] X. Qin, Z. Yan, and G. He, "A near-optimal detection scheme based on joint steepest descent and jacobi method for uplink massive mimo systems," *IEEE Communications Letters*, vol. 20, no. 2, pp. 276–279, 2016.
- [14] A. Elgabli, A. Elghariani, V. Aggarwal, and M. Bell, "A low complexity detection algorithm based on alternating minimization," *arXiv preprint arXiv:1809.02119*, 2018.
- [15] A. Elghariani and M. Zoltowski, "Low complexity detection algorithms in large-scale mimo systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1689–1702, 2016.
- [16] F. Jiang, C. Li, and Z. Gong, "Low complexity and fast processing algorithms for v2i massive mimo uplink detection," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5054–5068, 2018.
- [17] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block admm with o(1/k) convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

- [18] A. Elgabli, A. Elghariani, V. Aggarwal, and M. R. Bell, "A low-complexity detection algorithm for uplink massive mimo systems based on alternating minimization," *IEEE wireless communication letters*, 2019.
- [19] A. Elgabli, A. Elghariani, A. O. Al-Abbasi, and M. Bell, "Two-stage lasso admm signal detection algorithm for large scale mimo," in *Signals, Systems, and Computers, 2017 51st Asilomar Conference on*. IEEE, 2017, pp. 1660–1664.
- [20] L. Dai, X. Gao, X. Su, S. Han, I. Chih-Lin, and Z. Wang, "Low-complexity soft-output signal detection based on gauss–seidel method for uplink multiuser large-scale mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4839–4845, 2015.