Ph.D. DISSERTATION

Massive Connectivity with Compressive Sensing for Machine-Type Communications

사물 통신에서 압축 센싱을 이용한 대규모 연결 방법에 대한 연구

BY

BYEONG KOOK JEONG

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Massive Connectivity with Compressive Sensing for Machine-Type Communications

사물 통신에서 압축 센싱을 이용한 대규모 연결 방법에 대한 연구

BY

BYEONG KOOK JEONG

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Massive Connectivity with Compressive Sensing for Machine-Type Communications

사물 통신에서 압축 센싱을 이용한 대규모 연결 방법에 대한 연구

지도교수 이 광 복

이 논문을 공학박사 학위논문으로 제출함

2019년 2월

서울대학교 대학원

전기 정보 공학부

정 병 국

정병국의 공학박사 학위 논문을 인준함

2019년 2월

위 원 장: _____
부위원장: _____
위 원: _____
위 원: _____
위 원: _____

Abstract

Massive machine-type communication (mMTC) has become one of the most important requirements for next generation (5G) communication systems with the advent of the Internet-of-Things (IoT). In the mMTC scenarios, grant-free non-orthogonal multiple access (NOMA) on the transmission side and compressive sensing-based multi-user detection (CS-MUD) on the reception side are promising because many users sporadically transmit small data packets at low rates. In this dissertation, we propose a novel CS-MUD algorithm for active user and data detection for the mMTC systems. The proposed scheme consists of a MAP-based active user detector (MAP-AUD) and a MAP-based data detector (MAP-DD). By exchanging extrinsic information between MAP-AUD and MAP-DD, the proposed algorithm improves the performance of the active user detection and the reliability of the data detection. In addition, we extend the proposed algorithm to exploit group sparsity. By jointly processing the multiple received data with common activity, the proposed algorithm demonstrates dramatically improved performance. We show by numerical experiments that the proposed algorithm achieves a substantial performance gain over existing algorithms.

keywords: massive machine-type communications (mMTC), compressive sensing-based multi-user detection (CS-MUD), maximum *a posteriori* probability (MAP), active user detection (AUD), group sparsity

student number: 2016-30214

Contents

Abstract **i**

Contents **ii**

List of Tables **iv**

List of Figures **v**

1 Introduction **1**

2 MAP-based Active User and Data Detection **7**

2.1 System Model 7

2.2 MAP-based Active User and Data Detection 9

2.2.1 Activity Log-Likelihood Ratios 11

2.2.2 MAP-based Active User Detection 12

2.2.3 MAP-based Data Detection 16

2.2.4 Inversion of Covariance Matrices 20

2.2.5 Comments on Complexity 23

3 Group Sparsity-Aware Active User and Data Detection **28**

3.1 Extraction of Extrinsic User Activity Information 28

3.2 Modified Active User and Data Detection 32

4 Numerical Results **35**

4.1 Simulation Setup 35

4.2 Simulation Results 52

5 Conclusion **55**

Abstract (In Korean) **61**

Acknowlegement **63**

List of Tables

2.1 Summary of the proposed algorithm 19
2.2 Comparison of computational complexity 27

3.1 Summary of the extended algorithm 34

4.1 Simulation Parameters 36

List of Figures

1.1 (a) Illustration of the mMTC uplink transmission. 5

1.1 (b) Time diagram of the nMTC uplink transmission. 6

2.1 Iterative structure of the proposed algorithm. 10

2.2 Recursion-based covariance update. 22

3.1 Factor graph of the extended algorithm exploiting the group sparsity. . 30

3.2 Example of the evolution of activity LLR $\widetilde{L}_{A,n}$ for two active users ($\{2, 4\}$) out of the total $N(= 6)$ users and $K(= 4)$ symbols in a frame. 31

4.1 (a) AUD success probability as a function of the average SNR. 38

4.1 (b) Net SER as a function of the average SNR. 39

4.2 (a) AUD success probability when the group sparsity is exploited. . . 40

4.2 (b) Net SER when the group sparsity is exploited. 41

4.3 (a) AUD success probability when the interference covariance matrix is approximated to be diagonal. 42

4.3 (b) Net SER when the interference covariance matrix is approximated to be diagonal. 43

4.4 (a) AUD success probability when *a priori* user activities are incorrect. 44

4.4 (b) Net SER when *a priori* user activities are incorrect. 45

4.5 (a) AUD success probability for the various frame lengths. 46

4.5 (b) Net SER for the various frame lengths. 47

4.6 (a) AUD success probability for the various user activities. 48
4.6 (b) Net SER for the various user activities. 49
4.7 (a) AUD success probability for the various spreading factors. 50
4.7 (b) Net SER for the various spreading factors. 51

Chapter 1

Introduction

With the advent of the Internet-of-Things (IoT) era, machine-type communications have received a great deal of attention in recent years. In fact, we are witnessing a trend that numerous machine-type devices, such as mobile devices, machines, and sensors, are connected to the internet via wireless links [1]. In accordance with this trend, the International Telecommunication Union (ITU) defined massive machine-type communication (mMTC) as one of representative service categories for next generation (5G) wireless systems [2]. The mMTC focuses on the uplink communication of a large number of devices that *sporadically* transmit *short-sized* packets with *low transmission rates* to the base station (BS) [3]. In the mMTC perspective, the conventional multiple access mechanism in which the BS allocates orthogonal time and frequency resources to each user through complicated scheduling is not relevant since it will increase the signaling overhead and latency significantly [4, 5].

To overcome these shortcomings, *grant-free* non-orthogonal multiple access (NOMA) approaches have been proposed in recent years [6–9]. In the grant-free multiple access scheme, since the BS is not aware of the users transmitting information, an operation to distinguish *active* users from all possible potential users needs to be performed before data detection. When the number of active users is small, i.e., user activity is low, compressive sensing based multi-user detection (CS-MUD) is a good choice to solve the

problem at hand since it outperforms the classical MUD based on linear least-square (LS) and minimum mean square error (MMSE) detection [10]. Overall, CS-MUD can be classified into two categories: *convex optimization* based algorithms and *greedy* algorithms. The former formulates CS-MUD as an LS problem regularized by a sparsity promoting term, which is solved by convex optimization techniques [11–13]. The latter iteratively finds an active user and removes its vestige from the received signal in a greedy fashion [14–16]. Due to the computational benefit and competitive performance, greedy algorithms have been popularly used in the mMTC scenarios [17–21].

In finding out active users, most greedy algorithms rely on the correlation between the modified received vector (called residual) and the column vector (which corresponds to a user) of the channel matrix[1] as a decision statistic because the correlation is a simple yet effective statistic to test the user activity [21–24]. In [21], a group orthogonal matching pursuit (group OMP or GOMP) exploiting common sparsity caused by a frame structure has been proposed. It is similar to the simultaneous OMP (SOMP) [22] in that both enhance the detection performance by accumulating the correlation for a group of symbols. In [23], an iterative order recursive least square (IORLS) has been proposed. IORLS enhances GOMP by employing the whole symbols in a frame. In [24], a prior-information aided adaptive compressive sensing (PIA-ASP) have been proposed. PIA-ASP uses the temporal correlation between activities of adjacent symbols. Clearly, using the correlation is simple and easy, but the selection of a user having the maximum correlation may not be the right choice depending on the distribution of the channel matrix, the transmit data, and the noise. To address this problem, greedy algorithms called Bayesian pursuit algorithm (BPA) have been proposed [25,26]. Since BPA exploits the *a priori* distribution of the transmit data and the user activity, it performs better than the correlation-based greedy algorithms. However,

[1]The channel matrix (a.k.a. the *sensing matrix* and/or the *dictionary* in the CS literature) is the matrix which represents the relationship between the received vector and the transmit vector containing the data of all active and inactive users.

the performance depends heavily on the reliability of the *a priori* information.

In CS-MUD, the data detection is as important as the active user detection because the vestige of detected users has to be removed to form a residual signal. The commonly used data detection schemes are subspace projection methods such as LS and MMSE detection [21, 23–25, 27]. Using a finite alphabet constraint of the transmit data, the detection performance can be further improved [28–32]. In [28–31], sparsity-aware sphere detection (SA-SD) has been proposed. SA-SD performs close to the maximum likelihood (ML) detection but it requires considerable computational complexity caused by the combinatorial list search and the burdensome preprocessing (e.g., QR-decomposition). In [32], soft-feedback OMP (SF-OMP) has been proposed. SF-OMP improves the reliability of the data detection by the subspace projection followed by the sigmoid-like slicing. However, the performance of SF-OMP highly depends on the channel matrix structure.

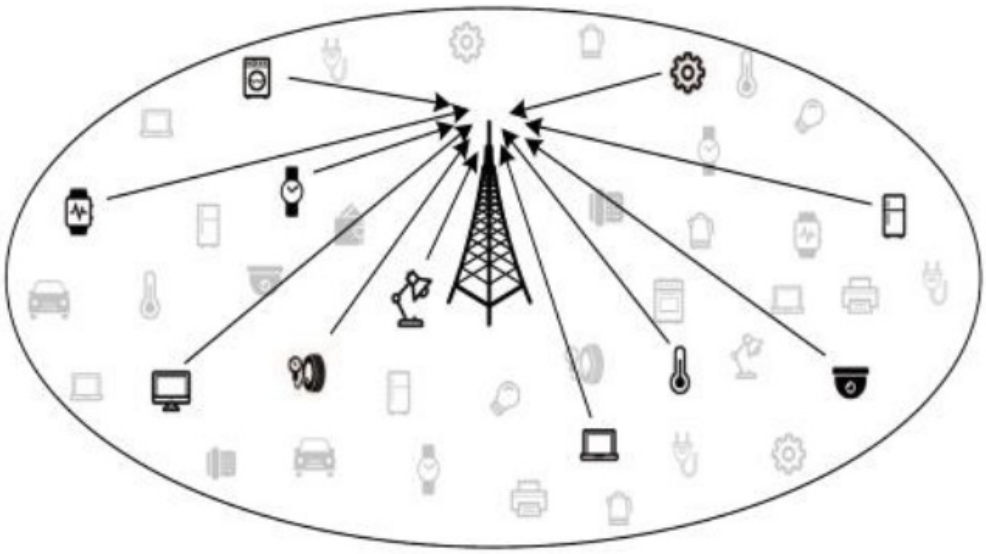
An aim of this dissertation is to propose a greedy algorithm that performs the identification of active users and the data detection simultaneously based on the maximum a posteriori probability (MAP) criterion. We exploit the finite alphabet constraint of the transmit data and the common sparsity inferred from the frame structure. The proposed algorithm is distinct from conventional approaches in that the *a posteriori* activity probability is used to detect the active user and the soft symbol information based on the *a posteriori* probability is used to detect the data. By exchanging *extrinsic* information between a MAP-based active user detector (MAP-AUD) and a MAP-based data detector (MAP-DD), the proposed algorithm improves the reliability of the *a posteriori* probabilities. Further, we aggregate the activity information of symbols in a frame to exploit the common activity. The activity information is derived from the soft symbol information. Using the aggregated activity information as modified *a priori* information, MAP-AUD and MAP-DD can enhance the reliability of the soft symbol information. In view of this, the overall algorithm can be thought as a *message-passing algorithm* [33] which employs the activity information as a message. We show from

numerical experiments that the proposed algorithm outperforms conventional greedy algorithms and in particular, performs close to an ideal detector with perfect knowledge on the user activity (which is called the Oracle detector) in the high signal-to-noise ratio (SNR) regime.

The rest of the dissertation is organized as follows[2]. The proposed MAP-AUD and MAP-DD are discussed in Chapter 2. The extension of the proposed algorithm to exploit group sparsity is discussed in Chapter 3. The numerical results are provided in Chapter 4. Lastly, this dissertation is concluded in Chapter 5.

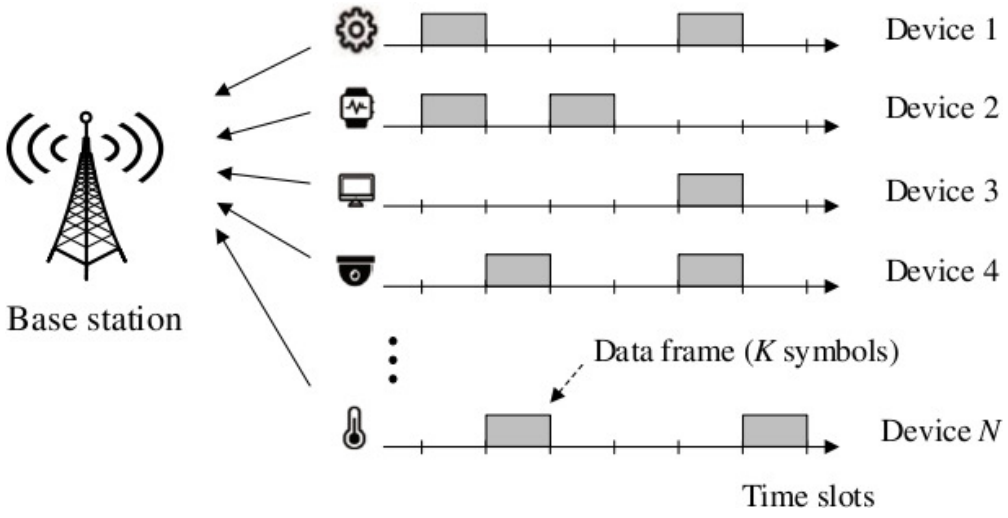
Notation: Boldface lower and upper-case characters represent column vectors and matrices, respectively. For a matrix $\mathbf{A}$, $\mathbf{A}^{-1}$, $\mathbf{A}^{\dagger}$, $\mathbf{A}^{T}$, $\mathbf{A}^{H}$, and $\mathbf{A}_{\mathcal{S}}$ are the inverse, pseudo-inverse, transpose, Hermitian transpose, and the sub-matrix with the columns in $\mathcal{S}$, respectively. For a vector $\mathbf{x}$, $\mathbf{x}_{\mathcal{S}}$ is the sub-vector with the elements in $\mathcal{S}$, $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^H\mathbf{A}\mathbf{x}$, and $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^H\mathbf{x}}$. For a complex number, $\mathcal{R}\{\cdot\}$ denotes the real part. For a set $\mathcal{A}$, $\overline{\mathcal{A}}$, $|\mathcal{A}|$ and $\mathcal{A}_j$ are the complementary set, cardinality, and j-th element, respectively. For a random variable, $\overline{(\cdot)}$ (or $E[\cdot]$) denotes the expectation and $\text{Cov}(\mathbf{x}) = E[\mathbf{x}\mathbf{x}^H] - E[\mathbf{x}]E[\mathbf{x}]^H$. As operators, $\otimes$ and $*$ denote the Kronecker product and convolution, respectively. Lastly, $\{x_k\}_{k=1:K}$ represents $\{x_1, x_2, \cdots, x_K\}$.

[2]The details of this dissertation can be also found in [34, 35].



(a)

Figure 1.1: (a) Illustration of the mMTC uplink transmission.



(b)

Figure 1.1: (b) Time diagram of the nMTC uplink transmission.

Chapter 2

MAP-based Active User and Data Detection

2.1 System Model

We consider the uplink transmission from N machine-type devices (which we call users in the sequel) to the BS. A user is inactive for most of the time and sporadically wakes up to monitor the physical or environmental condition and then transmit the information data to the BS, as illustrated in Fig. 1.1(a). We assume that each user and BS are synchronized, meaning that users switch activity and transmit the data on an identical time slot basis, as illustrated in Fig. 1.1(b).

The transmit data of an active user n consists of K symbols (which we call a frame in the sequel) so that the symbol vector is expressed as $\mathbf{d}_n = [d_{n,1}, d_{n,2}, \cdots, d_{n,K}]^T \in \mathbb{C}^K$. In particular, $\mathbf{d}_n = \mathbf{0}$ for the inactive user. Note that all elements of $\mathbf{d}_n$ have common symbol activity. Each symbol is spread by a user-specific spreading sequence vector $\mathbf{s}_n \in \mathbb{C}^M$ which is known at the BS and thus the transmit signal vector $\mathbf{m}_n$ in a frame is given by $\mathbf{m}_n = \mathbf{d}_n \otimes \mathbf{s}_n \in \mathbb{C}^{MK}$. We assume that each symbol is i.i.d. and uniformly drawn from a finite alphabet $\mathcal{A}$. We also assume that the user activity follows the i.i.d. *Bernoulli* distribution with an activity probability of p_n which is known at the BS. In many applications of mMTC such as smart metering, factory automation, surveillance, and health monitoring, the information is generated periodically [1] and

the BS exploits *a priori* knowledge on p_n estimated by statistics.

In this setup, the received signal vector $\mathbf{y}_k \in \mathbb{C}^M$ $(1 \leq k \leq K)$ can be expressed as

$$\begin{aligned}\mathbf{y}_k &= \sum_{n=1}^{N} (\mathbf{h}_n * \mathbf{s}_n) d_{n,k} + \mathbf{v}_k \\ &= \mathbf{A}\mathbf{x}_k + \mathbf{v}_k, \end{aligned} \tag{2.1}$$

where $\mathbf{h}_n \in \mathbb{C}^{\tau_n}$ is the fading channel between a user n and the BS with a length of τ_n, $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N] \in \mathbb{C}^{M \times N}$ is the channel matrix capturing the spreading sequences and the fading channels, $\mathbf{x}_k = [d_{1,k}, d_{2,k}, \cdots, d_{N,k}]^T$ is the k-th symbol vector containing all active and inactive user data, and $\mathbf{v}_k$ is the additive white Gaussian noise vector ($\mathbf{v}_k \sim \mathcal{CN}(\mathbf{0}, \sigma_v^2 \mathbf{I}_M)$). Note that the last $(\tau_n - 1)$ samples in $\mathbf{y}_k$ after the convolution in (2.1) are omitted based on the assumption that the inter-symbol interference (ISI) is negligible. This is because the data rate of mMTC is low [1] so that the symbol duration M is much longer than the multi-path delay profile τ_n (i.e., $M \gg \tau_n$). We assume that the channels between users and the BS are under the block-fading, meaning that the channel matrix $\mathbf{A}$ is invariant during the frame and the BS has the perfect knowledge of the channels.

In the mMTC scenarios, the number of users is in general much larger than the amount of resources being used for the transmission (i.e., $N \gg M$) and the data vector $\mathbf{x}_k$ is sparse because only a few users are active at a time. In this sense, the active user and symbol detection problem can be modeled as a sparse signal recovery problem using multiple received signal vectors $\{\mathbf{y}_k\}_{k=1:K}$. In this chapter, we first propose an algorithm based on a single received signal vector $\mathbf{y}_k$ $(1 \leq k \leq K)$ and then, in the next chapter, extend it into an algorithm based on multiple received signal vectors $\{\mathbf{y}_k\}_{k=1:K}$.

2.2 MAP-based Active User and Data Detection

In this section, we propose a MAP-based active user and symbol detection algorithm using a single received signal vector $\mathbf{y}$ ($= \mathbf{y}_k$). For notational simplicity, we skip the subscript k indicating the symbol index in this section. Fig. 2.1 depicts the iterative structure of the proposed algorithm.

In essence, the proposed algorithm consists of two parts: MAP-AUD and MAP-DD. First, using the *a priori* user activity information L_A of all users and the soft symbol information L_{E_2} of detected users in the previous iterations as input, MAP-AUD finds the user n^* having the largest *a posteriori* user activity probability and then computes the soft symbol information L_{E_1} of the user n^*. To be specific, L_{E_2} is used to compute the soft symbols for the users detected in the previous iterations. These soft symbols are removed from the received vector $\mathbf{y}$ in the soft interference cancellation block (soft IC). Next, using L_{E_1} delivered from MAP-AUD, MAP-DD refines L_{E_2} of all detected users. The refined L_{E_2} is then fed back to MAP-AUD, completing one cycle of the iteration.

MAP-AUD and MAP-DD improve the quality of the active user and symbol detection by exchanging *extrinsic* information which serves as the *a priori* information to each other. We henceforth use the subscripts '1' and '2' to denote MAP-AUD and MAP-DD, respectively. According to this rule, L_{A_1} (L_{A_2}) and L_{E_1} (L_{E_2}) represent *a priori* soft symbol information input to MAP-AUD (MAP-DD) and extrinsic soft symbol information generated by MAP-AUD (MAP-DD), respectively.

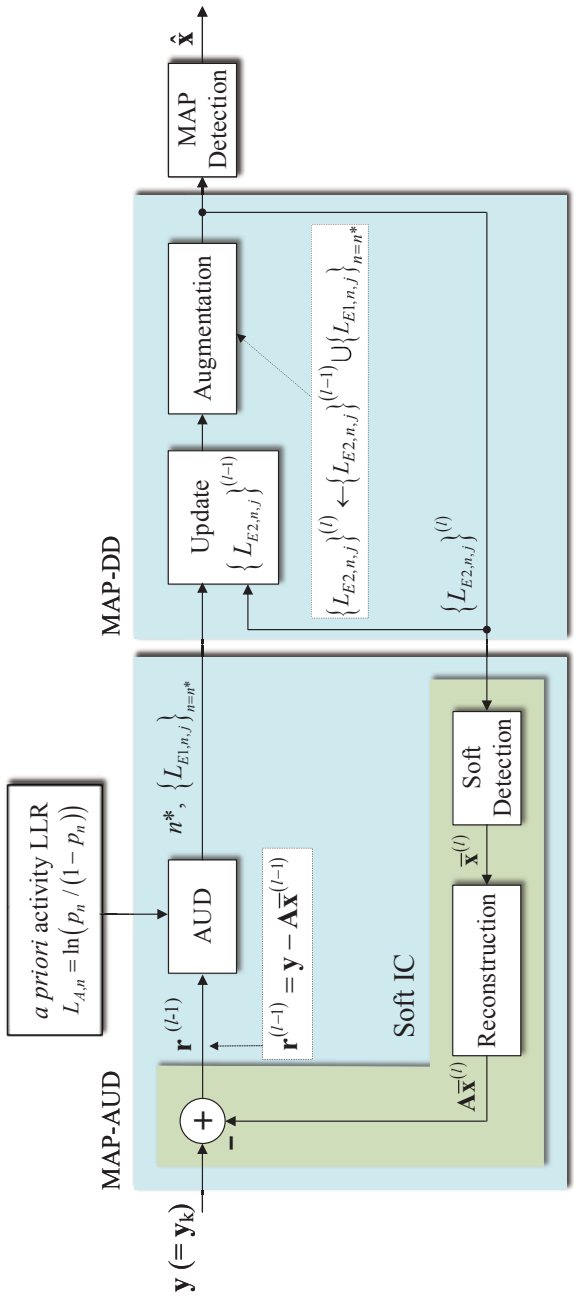


Figure 2.1: Iterative structure of the proposed algorithm.

2.2.1 Activity Log-Likelihood Ratios

In many iterative algorithms, the log-likelihood ratio (LLR) is used to extract *extrinsic* information from *a posteriori* information. In the proposed method, we use two types of activity LLRs: *user activity LLR* and *symbol-element activity LLR*. The user activity LLR refers to the level of user activity and the symbol-element activity LLR indicates the level on what element of the alphabet $\mathcal{A}$ is active. Note that the user activity is equivalent to the symbol activity.

The *a posteriori* user activity LLR L_n is defined as

$$L_n(\mathbf{y}) = \ln \frac{P\left(x_n \in \mathcal{A} \mid \mathbf{y}\right)}{P\left(x_n = 0 \mid \mathbf{y}\right)} = L_{E,n}(\mathbf{y}) + L_{A,n} \tag{2.2}$$

where x_n is the n-th element of the transmit symbol vector $\mathbf{x}$ in (2.1),

$$L_{E,n}(\mathbf{y}) = \ln \frac{P\left(\mathbf{y} \mid x_n \in \mathcal{A}\right)}{P\left(\mathbf{y} \mid x_n = 0\right)}, \tag{2.3}$$

and

$$L_{A,n} = \ln \frac{P(x_n \in \mathcal{A})}{P(x_n = 0)}. \tag{2.4}$$

In a similar manner, the *a posteriori* symbol-element activity LLR $L_{n,j}$ is defined as

$$L_{n,j}(\mathbf{y}) = \ln \frac{P\left(x_n = \mathcal{A}_j \mid \mathbf{y}\right)}{P\left(x_n = 0 \mid \mathbf{y}\right)} = L_{E,n,j}(\mathbf{y}) + L_{A,n,j} \tag{2.5}$$

where

$$L_{E,n,j}(\mathbf{y}) = \ln \frac{P\left(\mathbf{y} \mid x_n \in \mathcal{A}_j\right)}{P\left(\mathbf{y} \mid x_n = 0\right)} \tag{2.6}$$

and

$$L_{A,n,j} = \ln \frac{P\left(x_n \in \mathcal{A}_j\right)}{P\left(x_n = 0\right)}. \tag{2.7}$$

In (2.2) and (2.5), the subscripts E and A represent *extrinsic* and *a priori*, respectively.

From (2.2) and (2.5), it is clear that

$$\begin{aligned} L_n(\mathbf{y}) &= \ln \frac{P\left(x_n \in \mathcal{A} \mid \mathbf{y}\right)}{P\left(x_n = 0 \mid \mathbf{y}\right)} \\ &= \ln \sum_{\mathcal{A}_j \in \mathcal{A}} \frac{P\left(x_n \in \mathcal{A}_j \mid \mathbf{y}\right)}{P\left(x_n = 0 \mid \mathbf{y}\right)} \\ &= \ln \sum_{\mathcal{A}_j \in \mathcal{A}} \exp\left(L_{n,j}(\mathbf{y})\right). \end{aligned} \tag{2.8}$$

Similarly,

$$L_{A,n} = \ln \sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{A,n,j}\right). \tag{2.9}$$

In particular, if all elements of an alphabet $\mathcal{A}$ are equally probable, we have

$$L_{A,n,j} = L_{A,n} - \ln|\mathcal{A}|. \tag{2.10}$$

Noting that $P(x_n = 0) + \sum_{\mathcal{A}_j \in \mathcal{A}} P(x_n = \mathcal{A}_j) = 1$, we have

$$\begin{aligned} P\left(x_n = \mathcal{A}_j\right) &= \frac{\exp\left(L_{A,n,j}\right)}{1 + \sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{A,n,j}\right)} \\ P\left(x_n = 0\right) &= \frac{1}{1 + \sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{A,n,j}\right)}. \end{aligned} \tag{2.11}$$

In particular, if $P(x_n = 0) \ll 1$, we have

$$P\left(x_n = \mathcal{A}_j\right) \approx \frac{\exp\left(L_{A,n,j}\right)}{\sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{A,n,j}\right)}. \tag{2.12}$$

Note that $\{L_{A,n,j}\}_{j=1:|\mathcal{A}|}$ and $\{L_{E,n,j}\}_{j=1:|\mathcal{A}|}$ has one-to-one correspondence with $\{P(x_n = \mathcal{A}_j)\}_{j=1:|\mathcal{A}|} \cup \{P(x_n = 0)\}$ and $\{P(\mathbf{y}|x_n = \mathcal{A}_j)\}_{j=1:|\mathcal{A}|} \cup \{P(\mathbf{y}|x_n = 0)\}$. In this sense, $\{L_{A,n,j}\}_{j=1:|\mathcal{A}|}$ and $\{L_{E,n,j}\}_{j=1:|\mathcal{A}|}$ can be considered as the soft symbol information.

2.2.2 MAP-based Active User Detection

The goal of MAP-AUD is to find the user having the largest *a posteriori* user activity probability among undetected users. Let $\mathcal{S}^{(l-1)}$ be the support[1] of the $(l-1)$-th

[1]Support is an index set of non-zero elements which corresponds to detected users.

iteration, then the index of the user maximizing *a posteriori* user activity probability is

$$\begin{aligned} n^* &= \underset{n \in \overline{\mathcal{S}}^{(l-1)}}{\arg\max}\, L_n(\mathbf{y}) \\ &\overset{(a)}{=} \underset{n \in \overline{\mathcal{S}}^{(l-1)}}{\arg\max} \left(\ln \sum_{\mathcal{A}_j \in \mathcal{A}} \exp\left(L_{E_1,n,j}(\mathbf{y}) + L_{A,n,j}\right) \right) \\ &\overset{(b)}{=} \underset{n \in \overline{\mathcal{S}}^{(l-1)}}{\arg\max} \left(\ln \sum_{\mathcal{A}_j \in \mathcal{A}} \exp\left(L_{E_1,n,j}(\mathbf{y})\right) + L_{A,n} \right) \\ &\overset{(c)}{\approx} \underset{n \in \overline{\mathcal{S}}^{(l-1)}}{\arg\max} \left(\max_j L_{E_1,n,j}(\mathbf{y}) + L_{A,n} \right) \end{aligned} \tag{2.13}$$

where (a) is from (2.5) and (2.8), (b) is from (2.10), and (c) is from max-log approximation (i.e., $\ln \sum_j \exp(L_j) \approx \max_j L_j$). From (2.13), it is clear that we need both $L_{A,n}$ and $L_{E_1,n,j}$ to find n^*.

Using the *a priori* user activity probability p_n, we have

$$L_{A,n} = \ln \frac{p_n}{1 - p_n}. \tag{2.14}$$

To exploit the soft symbol information of the previously detected users delivered from MAP-DD, we modify $L_{E_1,n,j}$ in (2.6) as follows:

$$\begin{aligned} L_{E_1,n,j}(\mathbf{y}) &= \ln \frac{P(\mathbf{y} \mid x_n = \mathcal{A}_j)}{P(\mathbf{y} \mid x_n = 0)} \\ &= \ln \frac{E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[P\left(\mathbf{y} \mid x_n = \mathcal{A}_j, \mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\right]}{E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[P\left(\mathbf{y} \mid x_n = 0, \mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\right]} \\ &\overset{(a)}{\approx} \ln \frac{E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[\exp\left(-\left\|\mathbf{y} - \sum_{i \in \mathcal{S}^{(l-1)}} x_i \mathbf{a}_i - \mathcal{A}_j \mathbf{a}_n\right\|^2_{\mathbf{C}_n^{(l)-1}}\right)\right]}{E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[\exp\left(-\left\|\mathbf{y} - \sum_{i \in \mathcal{S}^{(l-1)}} x_i \mathbf{a}_i\right\|^2_{\mathbf{C}_n^{(l)-1}}\right)\right]} \end{aligned} \tag{2.15}$$

where (a) follows from the Gaussian approximation of the interference-plus-noise vector and

$$\mathbf{C}_n^{(l)} = \mathrm{Cov}\left(\sum_{i \neq n, i \in \overline{\mathcal{S}}^{(l-1)}} x_i \mathbf{a}_i + \mathbf{v} \right). \tag{2.16}$$

Since the direct computation of (2.15) is intractable due to the large number of combinations in $\mathbf{x}_{\mathcal{S}^{(l-1)}}$, we instead use the approximation that $E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}[\exp(\cdot)] \approx \exp(E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}[\cdot])$. This approximation is accurate when the user indices chosen in the previous iterations are perfect and the symbol detection errors are also negligible (see the end of this section). Under these assumptions, (2.15) can be approximated as

$$\begin{aligned} L_{E_1,n,j}(\mathbf{y}) &\approx \ln \frac{\exp\left(E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[-\left\|\mathbf{y}-\sum_{i\in\mathcal{S}^{(l-1)}} x_i\mathbf{a}_i - \mathcal{A}_j\mathbf{a}_n\right\|^2_{\mathbf{C}_n^{(l)-1}}\right]\right)}{\exp\left(E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[-\left\|\mathbf{y}-\sum_{i\in\mathcal{S}^{(l-1)}} x_i\mathbf{a}_i\right\|^2_{\mathbf{C}_n^{(l)-1}}\right]\right)} \\ &= \mathcal{R}\left\{\left(2\mathcal{A}_j\underbrace{\left(\mathbf{y}-\sum_{i\in\mathcal{S}^{(l-1)}}\overline{x}_i\mathbf{a}_i\right)}_{=\mathbf{r}^{(l-1)}} - |\mathcal{A}_j|^2\mathbf{a}_n\right)^H \mathbf{C}_n^{(l)-1}\mathbf{a}_n\right\} \end{aligned} \tag{2.17}$$

where $\mathbf{r}^{(l-1)}$ is the residual vector from the previous iteration (see Fig. 2.1). By denoting the *a priori* LLR of x_i ($i \in \mathcal{S}^{(l-1)}$) as $L_{A_1,i,j}$, the soft symbol $\overline{x}_i$ in (2.17) can be expressed as

$$\overline{x}_i = \sum_{\mathcal{A}_j\in\mathcal{A}} P(x_i=\mathcal{A}_j)\,\mathcal{A}_j \overset{(a)}{\approx} \frac{\sum_{\mathcal{A}_j\in\mathcal{A}}\exp(L_{A_1,i,j})\,\mathcal{A}_j}{\sum_{\mathcal{A}_j\in\mathcal{A}}\exp(L_{A_1,i,j})} \tag{2.18}$$

where (a) is from (2.12) because x_i is highly likely to be active (i.e., $P(x_i=0) \ll 1$) based on the assumption that the user indices chosen in the previous iterations are perfect. In (2.18), $L_{A_1,i,j}$ consists of $L_{A,i,j}$ derived from p_i and $L_{E_2,i,j}$ delivered from MAP-AUD. Since p_i is the user activity information, it does not contain any information about what alphabet element in $\mathcal{A}$ the symbol x_i is generated from. Therefore, by applying the equi-probable alphabet assumption, from (2.10), we have

$$L_{A_1,i,j} = L_{A,i,j} + L_{E_2,i,j} = L_{A,i} - \ln|\mathcal{A}| + L_{E_2,i,j}. \tag{2.19}$$

Combining (2.18) and (2.19), we have

$$\begin{aligned} \overline{x}_i &\approx \frac{\sum_{\mathcal{A}_j\in\mathcal{A}}\exp(L_{A,i}-\ln|\mathcal{A}|+L_{E_2,i,j})\,\mathcal{A}_j}{\sum_{\mathcal{A}_j\in\mathcal{A}}\exp(L_{A,i}-\ln|\mathcal{A}|+L_{E_2,i,j})} \\ &= \frac{\sum_{\mathcal{A}_j\in\mathcal{A}}\exp(L_{E_2,i,j})\,\mathcal{A}_j}{\sum_{\mathcal{A}_j\in\mathcal{A}}\exp(L_{E_2,i,j})} \end{aligned} \tag{2.20}$$

where $i \in \mathcal{S}^{(l-1)}$. Note that $\overline{x}_i$ depends only on the extrinsic LLR $L_{E_2,i,j}$ delivered from MAP-DD.

In (2.14) and (2.17), we obtained the statistics $L_{A,n}$ and $L_{E_1,n,j}$ to identify the active user index n^* in (2.13). Once the active user index n^* is found, the support set is updated as $\mathcal{S}^{(l)} = \mathcal{S}^{(l-1)} \cup \{n^*\}$ and the *extrinsic* LLR $\{L_{E_1,n,j}\}_{n=n^*,j=1:|\mathcal{A}|}$ is delivered to MAP-DD.

Proof of the Approximation $E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}[\exp(\cdot)] \approx \exp(E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}[\cdot])$

Recall the assumption that the identified user indices of the previous iterations are perfect and the symbol detection errors are also negligible. Hence, for all user indices $n \in \mathcal{S}^{(l-1)}$, $P(x_n = 0) \ll 1$ and there exists an alphabet element $\mathcal{A}_{j^*}$ which has the dominant probability; $P(x_n = \mathcal{A}_{j^*}) \approx 1$ and $P(x_n = \mathcal{A}_j) \ll 1$ for all $j \in \mathcal{A} - \{j^*\}$. We take such j^* for each user. Subsequently, because $0 \leq \exp\left(-\|f\left(\mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right) \leq 1$ (i.e., lower/upper-bounded), we have

$$\begin{aligned} &E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[\exp\left(-\|f\left(\mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right)\right] \\ &\quad= \sum_{\mathbf{x}_{\mathcal{S}^{(l-1)}} \in \Omega} \exp\left(-\|f\left(\mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right) P\left(\mathbf{x}_{\mathcal{S}^{(l-1)}}\right) \\ &\quad\approx \exp\left(-\|f\left(\mathbf{x}^*_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right) \end{aligned} \tag{2.21}$$

where $f(\mathbf{x}_{\mathcal{S}^{(l-1)}})$ is an arbitrary affine function of $\mathbf{x}_{\mathcal{S}^{(l-1)}}$ (see (2.17)), Ω is the set of all possible combinations of $\mathbf{x}_{\mathcal{S}^{(l-1)}}$, and $\mathbf{x}^*_{\mathcal{S}^{(l-1)}} = \left[\mathcal{A}_{j^*_1}, \cdots, \mathcal{A}_{j^*_{l-1}}\right]^T$. Similarly, we have

$$\begin{aligned} &\exp\left(E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[-\|f\left(\mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right]\right) \\ &\quad\approx \exp\left(-\|f\left(\mathbf{x}^*_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right). \end{aligned} \tag{2.22}$$

Combining (2.21) and (2.22), we have the desired approximation as

$$\begin{aligned} &E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[\exp\left(-\|f\left(\mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right)\right] \\ &\quad\approx \exp\left(E_{\mathbf{x}_{\mathcal{S}^{(l-1)}}}\left[-\|f\left(\mathbf{x}_{\mathcal{S}^{(l-1)}}\right)\|^2_{\mathbf{C}^{-1}}\right]\right). \end{aligned} \tag{2.23}$$

2.2.3 MAP-based Data Detection

The goal of MAP-DD is to obtain the extrinsic LLR $L_{E_2,n,j}$ of all user indices in $\mathcal{S}^{(l)}$. This information will be fed back to MAP-AUD for the next iteration and will also be used for the symbol detection when the iterative processing is completed. In a nutshell, MAP-DD consists of two processes: *update* and *augmentation*. The update process refines $L_{E_2,n,j}$ of all user indices detected in the previous iterations and the augmentation process then adds $L_{E_1,n^*,j}$ of the newly detected user index into the updated $L_{E_2,n,j}$[2]. After MAP-AUD, the received signal vector $\mathbf{y}$ can be decomposed as

$$\mathbf{y} = \sum_{i \in \mathcal{S}^{(l-1)}} x_i \mathbf{a}_i + x_{n^*} \mathbf{a}_{n^*} + \sum_{i \in \overline{\mathcal{S}}^{(l)}} x_i \mathbf{a}_i + \mathbf{v} \tag{2.24}$$

where $x_{n^*}\mathbf{a}_{n^*} + \sum_{i \in \overline{\mathcal{S}}^{(l)}} x_i \mathbf{a}_i$ is the interference in the previous iteration. Since $L_{E_1,n^*,j}$, which corresponds to the soft symbol information on x_{n^*}, is available thanks to MAP-AUD, we can refine $L_{E_2,n,j}$ by excluding $x_{n^*}\mathbf{a}_{n^*}$ in (2.24) from the interference.

In MAP-DD, we exploit the soft symbol information on x_n, which is $L_{E_2,n,j}$ obtained in the previous iteration and $L_{E_1,n^*,j}$ delivered from MAP-AUD. Similar to MAP-AUD, we modify $L_{E_2,n,j}$ in (2.6) as follows:

$$\begin{aligned} L_{E_2,n,j}(\mathbf{y}) &= \ln \frac{P(\mathbf{y} \mid x_n = \mathcal{A}_j)}{P(\mathbf{y} \mid x_n = 0)} \\ &= \ln \frac{E_{\mathbf{x}_{\mathcal{T}_n^{(l)}}}\left[P\left(\mathbf{y} | x_n = \mathcal{A}_j, \mathbf{x}_{\mathcal{T}_n^{(l)}}\right)\right]}{E_{\mathbf{x}_{\mathcal{T}_n^{(l)}}}\left[P\left(\mathbf{y} | x_n = 0, \mathbf{x}_{\mathcal{T}_n^{(l)}}\right)\right]} \\ &\overset{(a)}{\approx} \ln \frac{E_{\mathbf{x}_{\mathcal{T}_n^{(l)}}}\left[\exp\left(-\left\|\mathbf{y} - \sum_{i \in \mathcal{T}_n^{(l)}} x_i \mathbf{a}_i - \mathcal{A}_j \mathbf{a}_n\right\|^2_{\mathbf{\Gamma}^{(l)-1}}\right)\right]}{E_{\mathbf{x}_{\mathcal{T}_n^{(l)}}}\left[\exp\left(-\left\|\mathbf{y} - \sum_{i \in \mathcal{T}_n^{(l)}} x_i \mathbf{a}_i\right\|^2_{\mathbf{\Gamma}^{(l)-1}}\right)\right]} \end{aligned} \tag{2.25}$$

[2]In the first iteration, the update process is skipped because $\mathcal{S}^{(0)}$ is empty.

where $\mathcal{T}_n^{(l)} = \mathcal{S}^{(l)} - \{n\}$ and (a) is from the Gaussian approximation of the interference-plus-noise, and

$$\mathbf{\Gamma}^{(l)} = \mathrm{Cov}\left(\sum_{i \in \overline{\mathcal{S}}^{(l)}} x_i \mathbf{a}_i + \mathbf{v}\right). \tag{2.26}$$

Note that the newly detected component $x_{n^*}\mathbf{a}_{n^*}$ is removed from the received signal vector $\mathbf{y}$ and hence does not contribute to the interference-plus-noise covariance matrix $\mathbf{\Gamma}^{(l)}$. Similar to MAP-AUD, we assume that the identified user indices are perfect and the symbol detection errors are also negligible. In this setting, (2.25) can be rewritten as

$$\begin{aligned} & L_{E_2,n,j}\left(\mathbf{y}\right) \\ & \approx \ln \frac{\exp\left(E_{\mathbf{x}_{\mathcal{T}_n^{(l)}}}\left[-\left\|\mathbf{y} - \sum_{i \in \mathcal{T}_n^{(l)}} x_i \mathbf{a}_i - \mathcal{A}_j \mathbf{a}_n\right\|^2_{\mathbf{\Gamma}^{(l)-1}}\right]\right)}{\exp\left(E_{\mathbf{x}_{\mathcal{T}_n^{(l)}}}\left[-\left\|\mathbf{y} - \sum_{i \in \mathcal{T}_n^{(l)}} x_i \mathbf{a}_i\right\|^2_{\mathbf{\Gamma}^{(l)-1}}\right]\right)} \\ & = \mathcal{R}\left\{\left(2\mathcal{A}_j\left(\mathbf{y} - \sum_{i \in \mathcal{T}_n^{(l)}} \overline{x}_i \mathbf{a}_i\right) - \left|\mathcal{A}_j\right|^2 \mathbf{a}_n\right)^H \mathbf{\Gamma}^{(l)-1} \mathbf{a}_n\right\}. \end{aligned} \tag{2.27}$$

By denoting the *a priori* soft symbol information on x_i as $L_{A_2,i,j}$, the soft symbol x_i in (2.27) can be expressed as

$$\overline{x}_i = \sum_{\mathcal{A}_j \in \mathcal{A}} P\left(x_i = \mathcal{A}_j\right) \mathcal{A}_j \approx \frac{\sum_{\mathcal{A}_j \in \mathcal{A}} \exp\left(L_{A_2,i,j}\right) \mathcal{A}_j}{\sum_{\mathcal{A}_j \in \mathcal{A}} \exp\left(L_{A_2,i,j}\right)} \tag{2.28}$$

where $L_{A_2,i,j} = L_{E_1,n^*,j}$ (delivered from MAP-AUD) if $i = n^*$ and $L_{A_2,i,j} = L_{E_2,i,j}$ (obtained in the previous iteration) otherwise. This update process is applied to all user indices $n \in \mathcal{S}^{(l-1)}$. After finishing the update process, $L_{E_1,n^*,j}$ is added into the updated LLR set.

The augmented $\{L_{E_2,n,j}\}_{n \in \mathcal{S}^{(l)}, j=1:|\mathcal{A}|}$ is then fed back to MAP-AUD for the next iteration. The iteration lasts until all active users are detected. Specifically, an iteration

stops when the magnitude of the residual vector $\|\mathbf{r}^{(l)}\|_2$ is smaller than the predefined threshold. After the final (L-th) iteration, symbol detection is performed by finding the alphabet index j^* maximizing $L_{E_2,n,j}$ for each user index $n \in \mathcal{S}^{(L)}$. In Table 2.1, we summarize the proposed algorithm.

Table 2.1: Summary of the proposed algorithm

Input: $\mathbf{A}$ (channel matrix), $\mathbf{y}$ (received vector),

σ_v^2 (AWGN variance), $\{p_n\}$ (user activity probability)

Output: $\widehat{\mathbf{x}}$ (estimated symbol vector), $\mathcal{S}$ (support set)

Subscript: n (user index), j (alphabet element index)

Step 1: (Initialization)

$\mathcal{S}^{(0)} = \varnothing$, $\{L_{E_2,n,j}\} = \varnothing$, $l = 1$.

Step 2: (MAP-AUD) Find n^* and deliver $\{L_{E_1,n^*,j}\}$ to MAP-DD.

$L_{A,n} = \ln p_n / (1 - p_n)$.

$$L_{E_1,n,j} = \mathcal{R}\left\{\left(2\mathcal{A}_j\left(\mathbf{y} - \sum_{i\in\mathcal{S}^{(l-1)}} \overline{x}_i \mathbf{a}_i\right) - |\mathcal{A}_j|^2 \mathbf{a}_n\right)^H \mathbf{C}_n^{(l)-1}\mathbf{a}_n\right\}$$

where $\overline{x}_i = \dfrac{\sum_{\mathcal{A}_j\in\mathcal{A}} \exp\left(L_{E_2,i,j}\right)\mathcal{A}_j}{\sum_{\mathcal{A}_j\in\mathcal{A}} \exp\left(L_{E_2,i,j}\right)}$.

$n^* = \arg\max_{n\in\overline{\mathcal{S}}^{(l-1)}} \left(\max_j L_{E_1,n,j} + L_{A,n}\right)$.

Step 3: (MAP-DD) Refine $\{L_{E_2,n,j}\}_{n\in\mathcal{S}^{(l-1)}}$.

/* At the first iteration, skip this step. */

$\mathcal{T}_n^{(l)} = \mathcal{S}^{(l-1)} \cup \{n^*\} - \{n\}$.

$$L_{E_2,n,j} = \mathcal{R}\left\{\left(2\mathcal{A}_j\left(\mathbf{y} - \sum_{i\in\mathcal{T}_n^{(l)}} \overline{x}_i \mathbf{a}_i\right) - \left|\mathcal{A}_j\right|^2 \mathbf{a}_n\right)^H \mathbf{\Gamma}^{(l)-1}\mathbf{a}_n\right\}$$

where $\overline{x}_i = \begin{cases} \dfrac{\sum_{\mathcal{A}_j\in\mathcal{A}} \exp\left(L_{E_1,i,j}\right)\mathcal{A}_j}{\sum_{\mathcal{A}_j\in\mathcal{A}} \exp\left(L_{E_1,i,j}\right)} & \text{if } i = n^*, \\ \dfrac{\sum_{\mathcal{A}_j\in\mathcal{A}} \exp\left(L_{E_2,i,j}\right)\mathcal{A}_j}{\sum_{\mathcal{A}_j\in\mathcal{A}} \exp\left(L_{E_2,i,j}\right)} & \text{otherwise.} \end{cases}$

Step 4: (Augmentation) Add n^* into $\mathcal{S}^{(l-1)}$.

$\mathcal{S}^{(l)} = \mathcal{S}^{(l-1)} \cup \{n^*\}$.

$\left\{L_{E_2,n,j}\right\}_{n\in\mathcal{S}^{(l)}} = \left\{L_{E_2,n,j}\right\}_{n\in\mathcal{S}^{(l-1)}} \cup \left\{L_{E_1,n,j}\right\}_{n=n^*}$.

Step 5: (Iteration) Repeat until stopping criteria are met.

$l = l + 1$ and then go to **step 2**.

Step 6: (Final results)

$$\widehat{x}_n = \begin{cases} \mathcal{A}_{j_n^*} \text{ where } j_n^* = \arg\max_j L_{E_2,n,j} & \text{if } n \in \mathcal{S}^{(L)}, \\ 0, & \text{otherwise.} \end{cases}$$

2.2.4 Inversion of Covariance Matrices

In the computation of $L_{E_1,n,j}$ in (2.17) and $L_{E_2,n,j}$ in (2.27), we need to compute the inverse covariance matrices $\mathbf{C}_n^{(l)-1}$ and $\mathbf{\Gamma}^{(l)-1}$. This operation is computationally burdensome because $\mathbf{C}_n^{(l)-1}$ and $\mathbf{\Gamma}^{(l)-1}$ should be computed for every user in every iteration. To reduce the computational complexity associated with the covariance matrix inversion, we exploit a recursion-based approach in this work. The key idea of this approach is to compute $\mathbf{C}_n^{(l)-1}\mathbf{a}_n$ and $\mathbf{\Gamma}^{(l)-1}\mathbf{a}_n$ instead of $\mathbf{C}_n^{(l)-1}$ and $\mathbf{\Gamma}^{(l)-1}$ (see (2.17) and (2.27)).

First, it is clear from (2.16) and (2.26) that

$$\begin{aligned}\mathbf{C}_n^{(l)} &= \mathrm{Cov}\left(\sum_{i\neq n, i\in\overline{\mathcal{S}}^{(l-1)}} x_i\mathbf{a}_i + \mathbf{v}\right)\\ &= \mathrm{Cov}\left(\sum_{i\in\overline{\mathcal{S}}^{(l-1)}} x_i\mathbf{a}_i - x_n\mathbf{a}_n + \mathbf{v}\right)\\ &= \mathbf{\Gamma}^{(l-1)} - \beta_n\mathbf{a}_n\mathbf{a}_n^H \end{aligned} \tag{2.29}$$

where $\beta_n = \overline{|x_n|^2} = (1/|\mathcal{A}|)\sum_{\mathcal{A}_j\in\mathcal{A}} p_n|\mathcal{A}_j|^2$. It is also clear from (2.26) that $\mathbf{\Gamma}^{(l)}$ satisfies a recursive equation as

$$\begin{aligned}\mathbf{\Gamma}^{(l)} &= \mathrm{Cov}\left(\sum_{i\in\overline{\mathcal{S}}^{(l)}} x_i\mathbf{a}_i + \mathbf{v}\right)\\ &= \mathrm{Cov}\left(\sum_{i\in\overline{\mathcal{S}}^{(l-1)}} x_i\mathbf{a}_i - x_{n^*}\mathbf{a}_{n^*} + \mathbf{v}\right)\\ &= \mathbf{\Gamma}^{(l-1)} - \beta_{n^*}\mathbf{a}_{n^*}\mathbf{a}_{n^*}^H. \end{aligned} \tag{2.30}$$

and

$$\mathbf{\Gamma}^{(0)} = \sigma_v^2\mathbf{I}_M + \mathbf{A}\mathbf{B}\mathbf{A}^H \tag{2.31}$$

where $\mathbf{B} = \mathrm{diag}([\beta_1, \beta_2, \cdots, \beta_N]^T)$. Applying the matrix inversion lemma[3] to (2.29)

[3] $\left(\mathbf{X} + \tau\mathbf{a}\mathbf{a}^H\right)^{-1}\mathbf{b} = \mathbf{X}^{-1}\mathbf{b} - \left(\frac{\tau\mathbf{a}^H\mathbf{X}\mathbf{b}}{1+\tau\mathbf{a}^H\mathbf{X}\mathbf{a}}\right)\mathbf{X}^{-1}\mathbf{a}$.

and (2.30), we have

$$\mathbf{C}_n^{(l)-1}\mathbf{a}_n = \left(\frac{1}{1-\beta_n\,\mathbf{a}_n^H\,\mathbf{\Gamma}^{(l-1)-1}\,\mathbf{a}_n}\right)\mathbf{\Gamma}^{(l-1)-1}\mathbf{a}_n \tag{2.32}$$

and

$$\mathbf{\Gamma}^{(l)-1}\mathbf{a}_n = \mathbf{\Gamma}^{(l-1)-1}\mathbf{a}_n + \left(\frac{\beta_{n^*}\mathbf{a}_{n^*}^H\mathbf{\Gamma}^{(l-1)-1}\mathbf{a}_n}{1-\beta_{n^*}\mathbf{a}_{n^*}^H\mathbf{\Gamma}^{(l-1)-1}\mathbf{a}_{n^*}}\right)\mathbf{\Gamma}^{(l-1)-1}\mathbf{a}_{n^*}. \tag{2.33}$$

Clearly, matrix inversion is unnecessary in the computation of $\mathbf{C}_n^{(l)-1}\mathbf{a}_n$ and $\mathbf{\Gamma}^{(l)-1}\mathbf{a}_n$ except for the first iteration. This recursion-based covariance update approach is depicted in Fig. 2.2 where (a) and (b) mean (2.32) and (2.33), respectively.

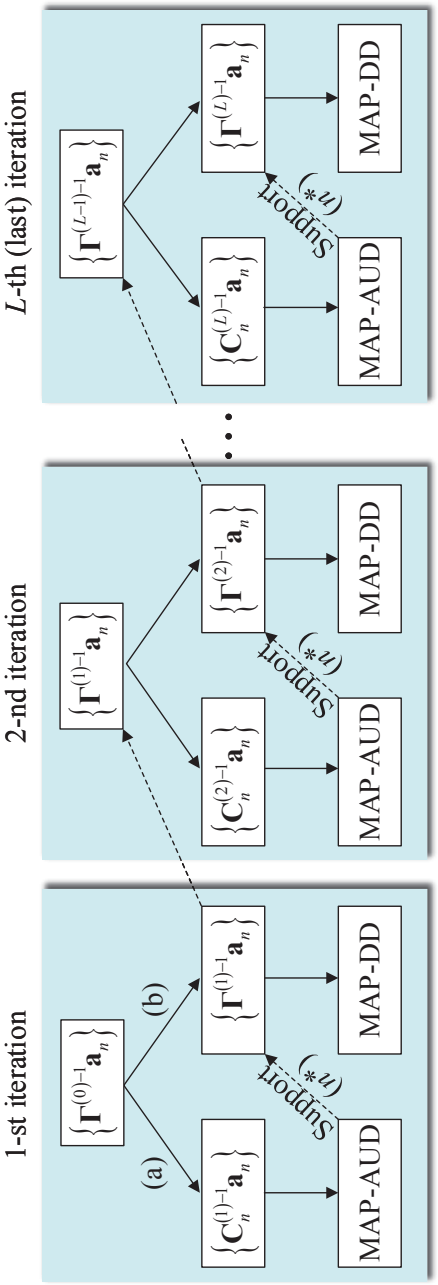


Figure 2.2: Recursion-based covariance update.

In case that the user-specific spreading sequence $\mathbf{s}_n$ is randomly generated and independent from each other, the covariance matrix $\mathbf{\Gamma}^{(l)}$ can be approximated as

$$\begin{aligned}\mathbf{\Gamma}^{(l)} &= \sum_{i\in\overline{\mathcal{S}}^{(l)}} \beta_i \mathbf{a}_i \mathbf{a}_i^H + \sigma_v^2 \mathbf{I}_M \\ &\approx \text{diag}\left(\sum_{i\in\overline{\mathcal{S}}^{(l)}} \beta_i \mathbf{a}_i \mathbf{a}_i^H\right) + \sigma_v^2 \mathbf{I}_M \\ &= \sum_{i\in\overline{\mathcal{S}}^{(l)}} \beta_i \,\text{diag}\left(\mathbf{a}_i \mathbf{a}_i^H\right) + \sigma_v^2 \mathbf{I}_M. \end{aligned} \tag{2.34}$$

We can explain the validity of the approximation in (2.34) with an example. First, recall that $\mathbf{a}_i = \mathbf{h}_i * \mathbf{s}_i$ (see (2.1)) and let $\mathbf{h}_i$ be $[h_{i,0}, h_{i,1}]^T$ as an example, then $a_{i,j}a_{i,j+1}^* = |h_{i,1}|^2 s_{i,j-1}s_{i,j}^* + h_{i,0}h_{i,1}^*|s_{i,j}|^2 + h_{i,0}^* h_{i,1} s_{i,j-1}s_{i,j+1}^* + |h_{i,0}|^2 s_{i,j}s_{i,j+1}^*$ where $a_{i,j}$, $h_{i,j}$, and $s_{i,j}$ denote the j-th element of $\mathbf{a}_i$, $\mathbf{h}_i$, and $\mathbf{s}_i$, respectively. Since the channels $\{\mathbf{h}_n\}_{n=1:N}$ are independent from each other, the summation ($\sum_{i\in\overline{\mathcal{S}}^{(l)}}$) of the second and third terms ($h_{i,0}h_{i,1}^*|s_{i,j}|^2$ and $h_{i,0}^* h_{i,1} s_{i,j-1}s_{i,j+1}^*$) will vanish if the number of the summed items is large enough. The other terms will also vanish after the summation under the assumption that $\mathbf{s}_n$ is randomly generated and independent from each other. In this way, we can easily show that $\mathbf{\Gamma}^{(l)}$ can be approximated as a diagonal matrix.

Using this diagonal approximation, we have

$$\begin{aligned}\mathbf{\Gamma}^{(0)} &= \sigma_v^2 \mathbf{I}_M + \sum_{i=1}^{N} \beta_i \text{diag}\left(\mathbf{a}_i \mathbf{a}_i^H\right), \\ \mathbf{C}_n^{(l)} &= \mathbf{\Gamma}^{(l-1)} - \beta_n \,\text{diag}\left(\mathbf{a}_n \mathbf{a}_n^H\right), \\ \mathbf{\Gamma}^{(l)} &= \mathbf{C}_{n^*}^{(l)}. \end{aligned} \tag{2.35}$$

Since $\mathbf{C}_n^{(l)}$ and $\mathbf{\Gamma}^{(l)}$ are diagonal, we can easily compute $\mathbf{C}_n^{(l)-1}$ and $\mathbf{\Gamma}^{(l)-1}$.

2.2.5 Comments on Complexity

In this section, we analyze the computational complexity of the proposed algorithm. We use the well-known OMP algorithm as a reference [14]. In analyzing the com-

plexity, we count the complex floating point operation (FLOP) such as addition and multiplication.

First, we analyze the complexity of OMP. OMP finds an active user by choosing a user having maximum correlation between the residual vector $\mathbf{r}^{(l-1)}$ and each column vectors of the channel matrix $\mathbf{A}$ in the l-th iteration, i.e., $\{\mathbf{r}^{(l-1)H}\mathbf{a}_n\}_{n\in\overline{\mathcal{S}}^{(l-1)}}$. Since $\mathbf{r}^{(l-1)} \in \mathbb{C}^M$, the complexity of this identification step is

$$\mathcal{C}_{\mathrm{I}} = \sum_{l=1}^{L} 2M(N-l+1) \approx 2LMN \tag{2.36}$$

where L is the total number of iterations. After the identification step, OMP estimates the data of all detected users, $\widehat{\mathbf{x}}_{\mathcal{S}^{(l)}}$, by projecting the received vector $\mathbf{y}$ to the subspace spanned by the column vectors corresponding to the detected users, i.e., $\widehat{\mathbf{x}}_{\mathcal{S}^{(l)}} = \mathbf{A}^{\dagger}_{\mathcal{S}^{(l)}}\mathbf{y} = (\mathbf{A}^{H}_{\mathcal{S}^{(l)}}\mathbf{A}_{\mathcal{S}^{(l)}})^{-1}\mathbf{A}^{H}_{\mathcal{S}^{(l)}}\mathbf{y}$. We approximate the complexity of the inversion of a matrix $\mathbf{X} \in \mathbb{C}^{l\times l}$ as $l^3/3$ [36]. Since $\mathbf{A}_{\mathcal{S}^{(l)}} \in \mathbb{C}^{M\times l}$, the complexity of this projection step is

$$\mathcal{C}_{\mathrm{P}} = \sum_{l=1}^{L}\left\{Ml(l+1) + \frac{l^3}{3} + 2Ml + 2l^2\right\} \approx \frac{1}{3}L^3M + \frac{1}{12}L^4. \tag{2.37}$$

Lastly, the residual vector $\mathbf{r}^{(l-1)}$ is updated to $\mathbf{r}^{(l)} = \mathbf{y} - \mathbf{A}_{\mathcal{S}^{(l)}}\mathbf{x}_{\mathcal{S}^{(l)}}$. The complexity of this update step is

$$\mathcal{C}_{\mathrm{U}} = \sum_{l=1}^{L}\{2Ml + M\} \approx L^2M. \tag{2.38}$$

Combining (2.36), (2.37), and (2.38), the total complexity of OMP is

$$\mathcal{C}_{\mathrm{OMP}} \approx 2LMN + \frac{1}{3}L^3M + L^2M + \frac{1}{12}L^4. \tag{2.39}$$

Now, we analyze the complexity of the proposed algorithm. First, we count FLOPs for $\{\mathbf{\Gamma}^{(0)-1}\mathbf{a}_n\}_{n=1:N}$. Using $\mathbf{\Gamma}^{(0)} = \sigma_v^2\mathbf{I}_M + \mathbf{A}\mathbf{B}\mathbf{A}^{\mathbf{H}} = \sigma_v^2\mathbf{I}_M + (\mathbf{A}\mathbf{B}^{\frac{\mathbf{1}}{\mathbf{2}}})(\mathbf{A}\mathbf{B}^{\frac{\mathbf{1}}{\mathbf{2}}})^{\mathbf{H}}$

(see (2.31)), the complexity of $\{\mathbf{\Gamma}^{(0)-1}\mathbf{a}_n\}_{n=1:N}$ is

$$\begin{aligned}\mathcal{C}_{\text{pre1}} &= \underbrace{M(M+1)N + MN + M}_{\mathbf{\Gamma}^{(0)}} + \underbrace{\frac{1}{3}M^3}_{\mathbf{\Gamma}^{(0)-1}} + \underbrace{2M^2N}_{\{\mathbf{\Gamma}^{(0)-1}\mathbf{a}_n\}} \\ &\approx 3M^2N + \frac{1}{3}M^3. \end{aligned} \tag{2.40}$$

From (2.32) and (2.33), the complexity of $\{\mathbf{C}_n^{(l)-1}\mathbf{a}_n\}_{n\in\overline{\mathcal{S}}^{(l)}}$ and $\{\mathbf{\Gamma}^{(l)-1}\mathbf{a}_n\}_{n\in\overline{\mathcal{S}}^{(l)}}$ is

$$\mathcal{C}_{\text{pre2}} = \sum_{l=1}^{L}(7M+6)(N-l) \approx 7LMN. \tag{2.41}$$

In the proposed algorithm, the (modified) correlation is $\widetilde{\mathbf{r}}_{n,j}^{(l-1)H}\mathbf{C}_n^{(l)-1}\mathbf{a}_n$ where $\widetilde{\mathbf{r}}_{n,j}^{(l-1)} = 2\mathcal{A}_j\mathbf{r}^{(l-1)} - |\mathcal{A}_j|^2\mathbf{a}_n$ (see (2.17)). The complexity of $\{\widetilde{\mathbf{r}}_{n,j}^{H}\mathbf{C}_n^{(l)-1}\mathbf{a}_n\}_{n\in\overline{\mathcal{S}}^{(l-1)},j=1:|\mathcal{A}|}$ is

$$\mathcal{C}_{\text{MAP-AUD}} \approx \sum_{l=1}^{L} 2|\mathcal{A}|M(N-l) \approx 2|\mathcal{A}|LMN \tag{2.42}$$

where we ignore the complexity of $\widetilde{\mathbf{r}}_{n,j}$ for simplicity since it is not a main factor. Since the operation of MAP-DD is similar to that of MAP-AUD (see (2.27)), we can easily show that

$$\mathcal{C}_{\text{MAP-DD}} = \sum_{l=1}^{L} 2|\mathcal{A}|Ml \approx |\mathcal{A}|L^2M. \tag{2.43}$$

Lastly, the complexity of the conversion from a LLR to a soft symbol (see (2.20) and (2.28)) is

$$\mathcal{C}_{\text{LLR}} = \sum_{l=1}^{L} 3|\mathcal{A}|l \approx \frac{3}{2}|\mathcal{A}|L^2. \tag{2.44}$$

Combining the complexity from (2.40) to (2.44), the total complexity of the proposed algorithm is

$$\mathcal{C}_{\text{prop}} \approx 3M^2N + (7+2|\mathcal{A}|)LMN + \frac{1}{3}M^3 + |\mathcal{A}|L^2M + \frac{3}{2}|\mathcal{A}|L^2. \tag{2.45}$$

Finally, we analyze the complexity when the diagonal approximation is applied to the covariance matrices. In this case, the complexity of $\{\mathbf{C}_n^{(l)}\}_{n\in\overline{\mathcal{S}}^{(l)}}$ and $\mathbf{\Gamma}^{(l)}$ is (see (2.35))

$$\mathcal{C}_{\text{diag-pre1}} = \underbrace{2MN + M}_{\mathbf{\Gamma}^{(0)}} + \sum_{l=1}^{L} M(N-l) \approx LMN. \tag{2.46}$$

Using the diagonal property of the covariance matrices, $\mathcal{C}_{\text{MAP-AUD}}$ in (2.42) and $\mathcal{C}_{\text{MAP-DD}}$ in (2.43) are

$$\begin{aligned} \mathcal{C}_{\text{diag-MAP-AUD}} &\approx (2|\mathcal{A}|+1)LMN \\ \mathcal{C}_{\text{diag-MAP-DD}} &\approx \frac{1}{2}(2|\mathcal{A}|+1)L^2M. \end{aligned} \tag{2.47}$$

Combining (2.46), (2.47), and (2.44), we have

$$\mathcal{C}_{\text{diag-prop}} \approx (2|\mathcal{A}|+2)LMN + \frac{1}{2}(2|\mathcal{A}|+1)L^2M + \frac{3}{2}|\mathcal{A}|L^2. \tag{2.48}$$

From (2.39), (2.45), and (2.47), we can observe that the complexity of the proposed algorithm is higher than OMP mainly because it depends on the alphabet size $|\mathcal{A}|$. However, since the low order modulation schemes (e.g., BPSK, QPSK) are used in the typical mMTC systems, the complexity of the proposed algorithm does not increase significantly compared to OMP. In Table 2.2, we summarize the complexity of OMP and proposed algorithms for various parameter settings (N, M, L, and modulation order) under the assumption that the user activity is about 10%.

Table 2.2: Comparison of computational complexity

Mod. (N, M, L)	$\mathcal{C}_{\textbf{OMP}}$	$\mathcal{C}_{\textbf{prop}}$	$\mathcal{C}_{\textbf{diag-prop}}$
BPSK $(64, 16, 8)$	2.05×10^4	1.42×10^5	5.19×10^4
BPSK $(64, 32, 8)$	4.06×10^4	3.92×10^5	1.04×10^5
QPSK $(64, 16, 8)$	2.05×10^4	1.78×10^5	8.69×10^4
QPSK $(64, 32, 8)$	4.06×10^4	4.62×10^5	1.73×10^5

Chapter 3

Group Sparsity-Aware Active User and Data Detection

In this chapter, we describe an active user and symbol detection algorithm operating on multiple received vectors. Since the user activity does not change in a frame, all symbols in the frame have common activity such that the estimated user activity information in an arbitrary received vector can be used as *a priori* activity information in the remaining received vectors. For each received vector, the user activity information of the detected users can be obtained from the symbol-element activity LLR $L_{E_2,n,j}$ computed by MAP-DD. Employing the message-passing framework, we can deliver the activity information to the remaining received vectors. To improve the quality of the activity information, we need to obtain the activity information of undetected users as well. We first explain how to extract the activity information of all detected and undetected users and then move on to the discussion of the extended MAP-AUD/DD exploiting the common activity.

3.1 Extraction of Extrinsic User Activity Information

First, after processing the received symbol vector $\mathbf{y}$, we obtain the extrinsic symbol-element activity LLR $L_{E_2,n,j}$ for the detected users ($n \in \mathcal{S}$). Since the extrinsic LLR serves as the *a priori* LLR of the remaining vectors, similar to (2.9), the extrinsic user

activity LLR $L_{E,n}$ is calculated as

$$L_{E,n} = \ln \sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{E_2,n,j}\right). \tag{3.1}$$

Next, noting that MAP-AUD computes the extrinsic symbol-element activity LLR $L_{E_1,n,j}$ for all user indices in $\overline{\mathcal{S}}$ (see (2.13)), we can obtain the activity information of undetected users by performing an additional iteration (except for MAP-DD). From (2.17), we have

$$L_{E_1,n,j}\left(\mathbf{y}\right) = \mathcal{R}\left\{\left(2\mathcal{A}_j\left(\mathbf{y} - \sum_{i\in\mathcal{S}^{(L)}} \overline{x}_i \mathbf{a}_i\right) - |\mathcal{A}_j|^2 \mathbf{a}_n\right)^H \mathbf{C}_n^{(L+1)-1} \mathbf{a}_n\right\} \tag{3.2}$$

where L denotes the last iteration index. Similar to (3.1), $L_{E,n}$ for the undetected users ($n \in \overline{\mathcal{S}}$) is

$$L_{E,n} = \ln \sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{E_1,n,j}\right). \tag{3.3}$$

From (3.1) and (3.3), $L_{E,n}$ for all detected and undetected users can be obtained.

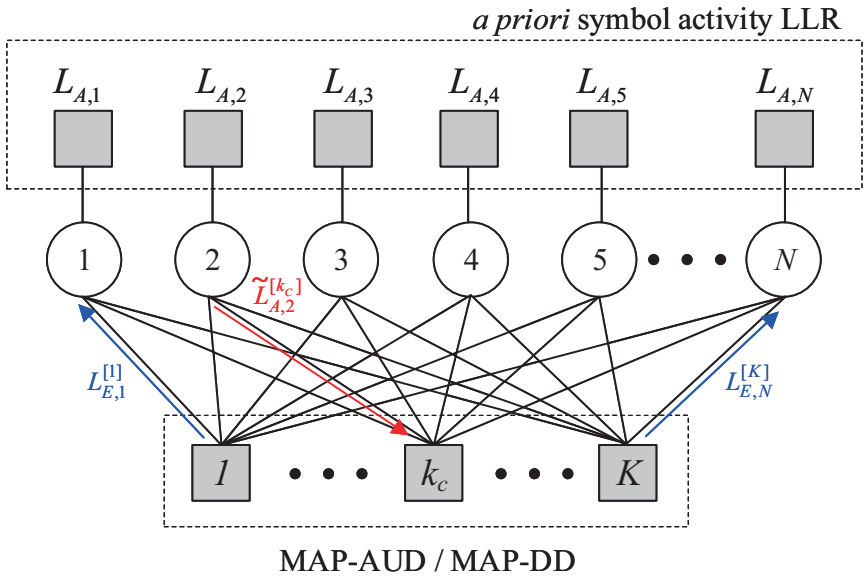


Figure 3.1: Factor graph of the extended algorithm exploiting the group sparsity.

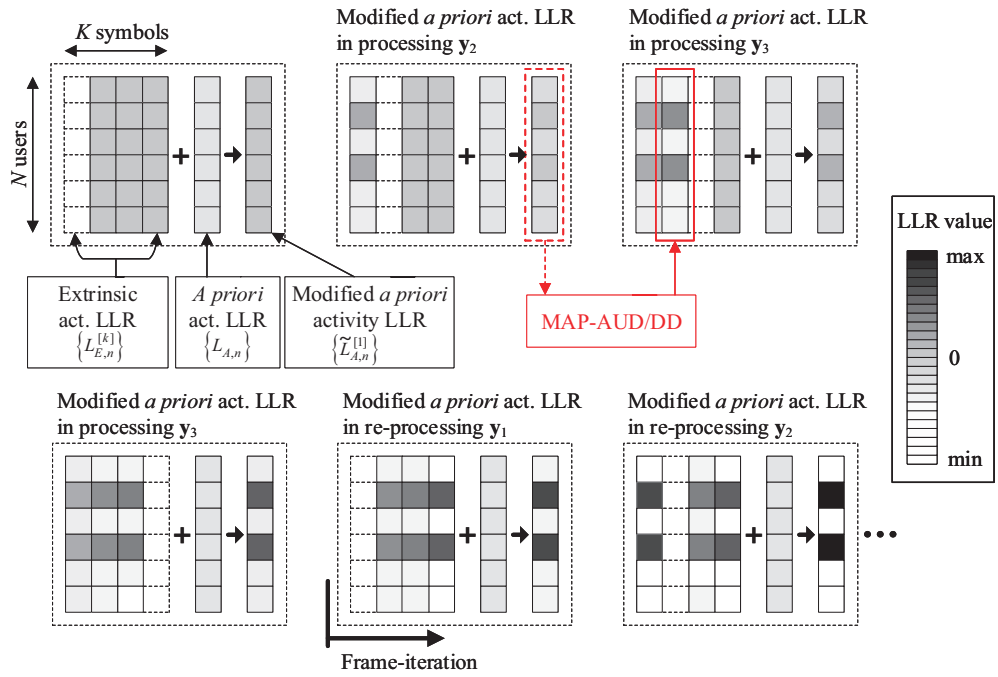


Figure 3.2: Example of the evolution of activity LLR $\widetilde{L}_{A,n}$ for two active users ($\{2, 4\}$) out of the total $N(= 6)$ users and $K(= 4)$ symbols in a frame.

3.2 Modified Active User and Data Detection

In this section, we extend the proposed algorithm in a way to exploit group sparsity. We use an additional superscript $[k]$ to indicate the symbol index of activity LLRs. The extrinsic user activity LLR $L_{E,n}^{[k]}$ is aggregated through the message-passing framework as illustrated by a factor graph in Fig. 3.1. In this factor graph, circle and square nodes represent variable and function nodes, respectively. The user activity LLR is used as a message and MAP-AUD/DD is performed in the function nodes. Accordingly, the modified *a priori* user activity LLR of a symbol k_c is expressed as

$$\widetilde{L}_{A,n}^{[k_c]} = w \sum_{k \neq k_c} L_{E,n}^{[k]} + L_{A,n} \tag{3.4}$$

where w is a weighting factor of $L_{E,n}^{[k]}$ learned from the group sparsity against $L_{A,n}$ obtained from the *a priori* user activity probability p_n (see (2.14)). In this work, we set w to $1/(K-1)$, meaning that the weight of the average of $L_{E,n}^{[k]}$ is identical to that of $L_{A,n}$. The extended algorithm employs $\widetilde{L}_{A,n}^{[k_c]}$ instead of $L_{A,n}$. Note that $\widetilde{L}_{A,n}^{[k_c]}$ evolves through a frame.

Fig. 3.2 illustrates the evolution of $\widetilde{L}_{A,n}^{[k]}$ when the symbol index k varies from 1 to K. The extrinsic user activity LLR $L_{E,n}^{[k]}$ and *a priori* user activity LLR $L_{A,n}$ are initialized with 0 and $\ln p_n/(1-p_n)$, respectively. First, when the received vector $\mathbf{y}_1$ is used as an input to MAP-AUD/DD, only $L_{A,n}$ is used as the *a priori* LLR ($\widetilde{L}_{A,n}^{[1]}$). Next, when $\mathbf{y}_2$ is used as the input, the output of MAP-AUD/DD, $L_{E,n}^{[1]}$, after processing $\mathbf{y}_1$, as well as $L_{A,n}$ are used as the *a priori* LLR ($\widetilde{L}_{A,n}^{[2]}$). In general, when $\mathbf{y}_{k_c}$ is used as the input, $\{L_{E,n}^{[k]}\}_{k=1:(k_c-1)}$ as well as $L_{A,n}$ are used as the *a priori* information ($\widetilde{L}_{A,n}^{[k_c]}$). As the number of received vectors k increases, $\widetilde{L}_{A,n}^{[k]}$ becomes more reliable because more extrinsic information serves as the *a priori* information and, more importantly, because only extrinsic information is aggregated. Moreover, noting that we can fully exploit the group sparsity only after receiving the entire frame (because $\{L_{E,n}^{[k]}\}$ is partially filled), we can further improve the performance by the frame iteration. In the frame iteration, MAP-AUD/DD is performed for the whole frame once again using the

updated $\{L_{E,n}^{[k]}\}_{k=1:K}$ as *a priori* information. In Table 3.1, the extended algorithm is summarized.

Table 3.1: Summary of the extended algorithm

Input: $\{p_n\}$ (user activity probability)

Output: $\{\widehat{\mathbf{x}}_k\}$ (estimated symbol vector)

Subscript: n (user index), j (alphabet element index),

$k(k_c)$ ((current) symbol index)

Step 1: (Initialization)

$k_c = 1$, $L_{E,n}^{[k]} = 0 \quad (1 \le n \le N, 1 \le k \le K)$

Step 2: (Symbol reception) /* Receive the symbol vector $\mathbf{y}_{k_c}$. */

Step 3: (Modified *a priori* activity LLR)

$\widetilde{L}_{A,n}^{[k_c]} = L_{A,n} + \frac{1}{(K-1)} \sum_{k \neq k_c} L_{E,n}^{[k]}$ /* (3.4) */

Step 4: (MAP-AUD/DD) /* Perform MAP-AUD/DD using $\widetilde{L}_{A,n}^{[k_c]}$. */

Obtain $L_{E_2,n,j}^{[k_c]}$ $(\forall n \in \mathcal{S}_{k_c})$ from MAP-DD. /* (2.27) */

Obtain $\widehat{\mathbf{x}}_{k_c}$ from MAP-DD.

Obtain $L_{E_1,n,j}^{[k_c]}$ $(\forall n \in \overline{\mathcal{S}}_{k_c})$ from additional MAP-AUD. /* (3.2) */

Step 5: (Extrinsic activity LLR) /* Obtain $L_{E,n}^{[k_c]}$. */

$$L_{E,n}^{[k_c]} = \begin{cases} \ln \sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{E_2,n,j}^{[k_c]}\right), & \text{for } n \in \mathcal{S}_{k_c} \\ \ln \sum_{j=1}^{|\mathcal{A}|} \exp\left(L_{E_1,n,j}^{[k_c]}\right) & \text{for } n \in \overline{\mathcal{S}}_{k_c} \end{cases}$$

Step 6: (Iteration) /* Perform MAP-AUD/DD if $k_c \le K$. */

/* In the frame iteration, skip **step 2**. */

$k_c = k_c + 1$ and then go to **step 2** or **3**.

Step 7: (Frame iteration) /* Repeat until stopping conditions are met. */

$k_c = 1$ and then go to **step 3**.

Chapter 4

Numerical Results

4.1 Simulation Setup

We simulate the uplink of an underdetermined mMTC system in which the number of users N is much larger than the spreading factor M ($N \gg M$). We generate each spreading sequence vector $\mathbf{s}_n$ by an i.i.d. complex Gaussian random vector ($\mathbf{s}_n \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_M)$) and then scale it as $\|\mathbf{s}_n\|_2 = 1$. We consider the frequency-flat Rayleigh fading channels between users and the BS generated by an i.i.d. complex Gaussian variable $\mathcal{CN}(0, 1)$. Thus, the average symbol SNR is set to $1/\sigma_v^2$. In this work, we terminate the iteration when $\|\mathbf{r}^{(l)}\|_2 \leq 10^{-3}$. In Table 4.1, the simulation parameters are summarized.

Table 4.1: Simulation Parameters

Parameter	**Parameter Value**
N: User number	32
M: Spreading factor	8
K: Frame length	64
$\mathcal{A}$: Symbol alphabet	$\{-1, 1\}$ (BPSK)
Simulation iteration	≥ 10000

As a reference, we use OMP [14] and SOMP [22] which is the group version of OMP. As a conventional technique exploiting the *a priori* information on user activities, we use the fast Bayesian pursuit algorithm (FBPA) (with searching parameter $D = 5$) [25] and SF-OMP [32]. In particular, we use the genie-aided SF-OMP (GA-SF-OMP), which has perfect knowledge of the variance of interferences (corresponding to undetected user signals). Note that GA-SF-OMP is actually unrealistic and represents the performance upper-bound of SF-OMP. In addition, we use GA-SF-SOMP which is a combination of SOMP and GA-SF-OMP as a reference for exploiting the group sparsity. Lastly, we use MMSE and Oracle MMSE/ML to represent the lower and upper bound of the performance, respectively. Since the references (except for MMSE/ML) need a normalized channel matrix, we slightly modify the system model as

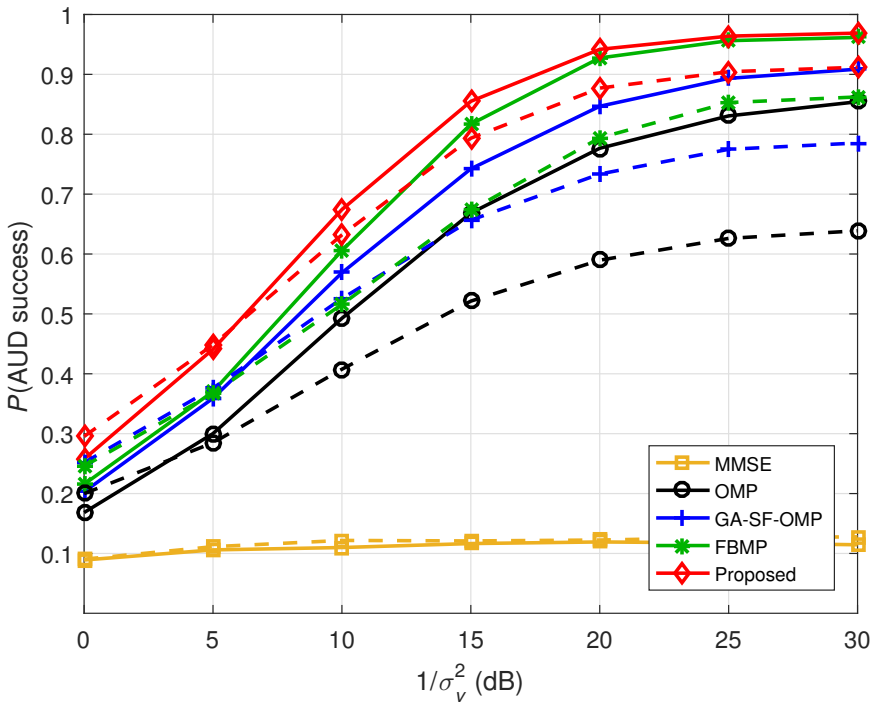
$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} = \widetilde{\mathbf{A}}\mathbf{D}\mathbf{x} + \mathbf{n} = \widetilde{\mathbf{A}}\widetilde{\mathbf{x}} + \mathbf{n} \tag{4.1}$$

where $\widetilde{\mathbf{A}}$ is the column-normalized channel matrix, $\mathbf{D}$ is the diagonal matrix having the l_2-norms of the column vectors as diagonal elements, and $\widetilde{\mathbf{x}} = \mathbf{D}\mathbf{x}$. In the simulation of the references, $\widetilde{\mathbf{x}}$ is initially estimated and then scaled to $\mathbf{x}$. Note that our algorithm does not require this modification at all.

As a performance measure, we use the successful AUD probability and the net SER as well. The net SER is the symbol error rate of active users and is defined as

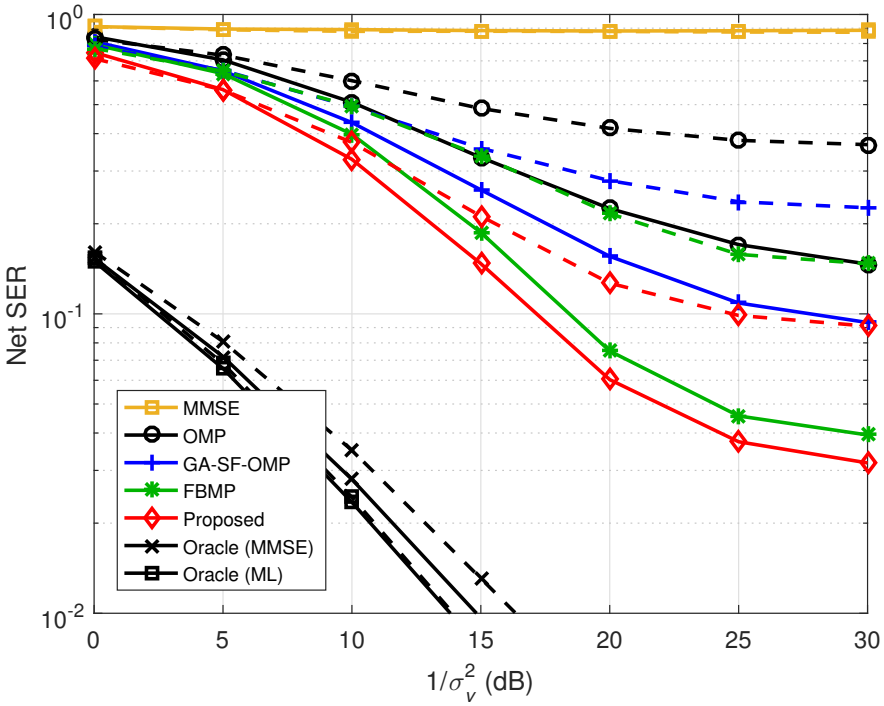
$$\text{Net SER} = 1 - P\left(\text{AUD success } \cap \text{ symbol detection success}\right).$$

We set the activities of all users to be equal (i.e., $p_n = p$ for all users) for simplicity. Note that the proposed algorithm can also be applied to scenarios having different user activities.



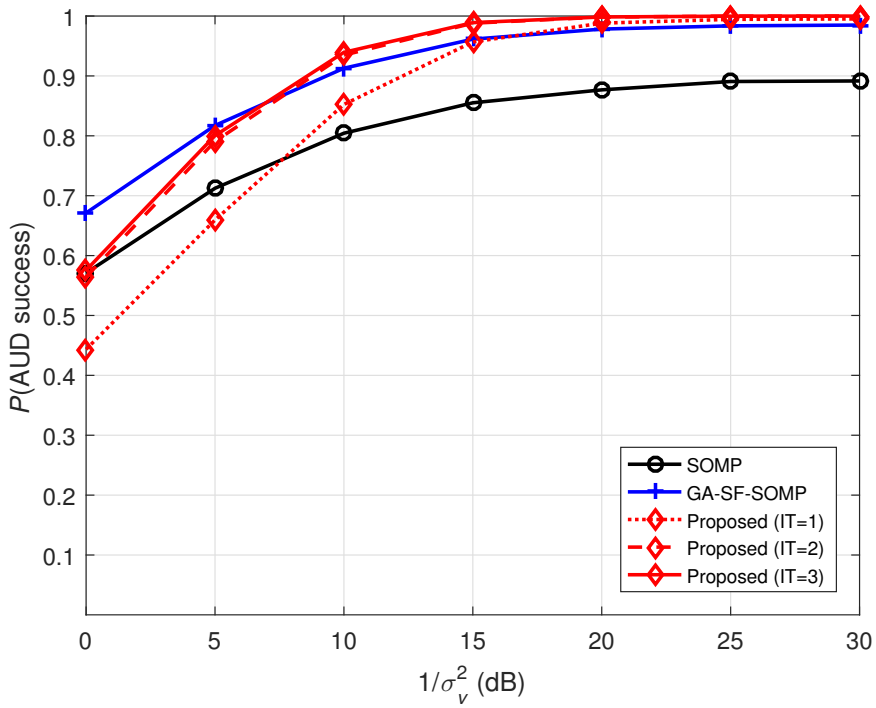
(a)

Figure 4.1: (a) AUD success probability as a function of the average SNR.



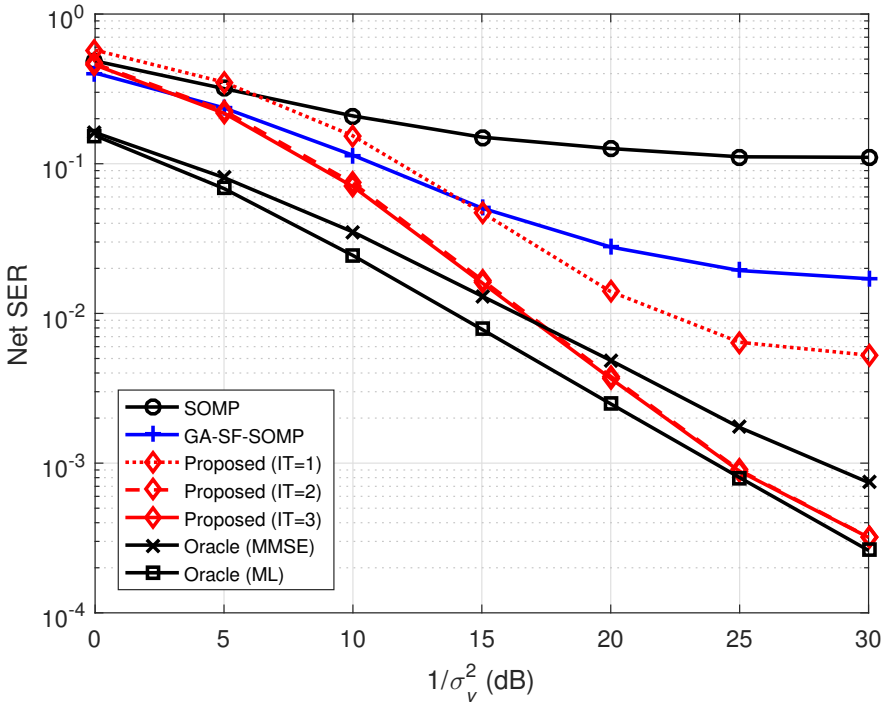
(b)

Figure 4.1: (b) Net SER as a function of the average SNR.



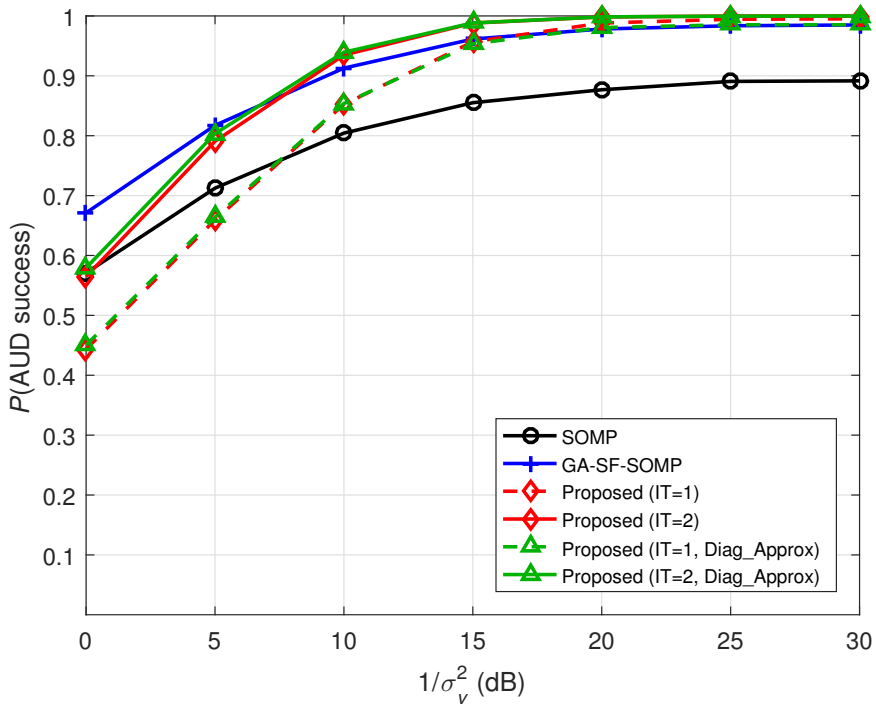
(a)

Figure 4.2: (a) AUD success probability when the group sparsity is exploited.



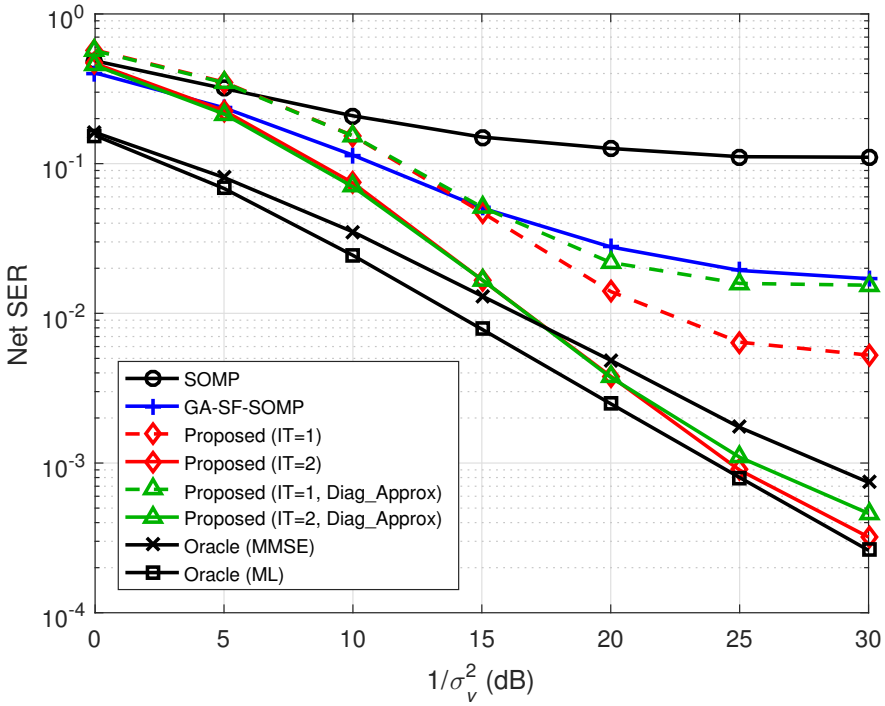
(b)

Figure 4.2: (b) Net SER when the group sparsity is exploited.



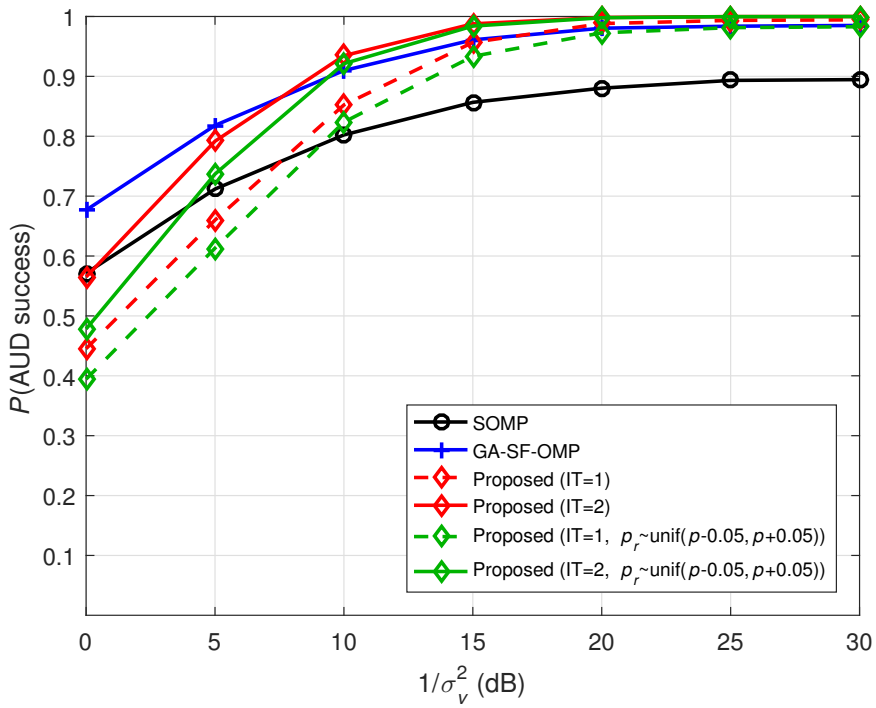
(a)

Figure 4.3: (a) AUD success probability when the interference covariance matrix is approximated to be diagonal.



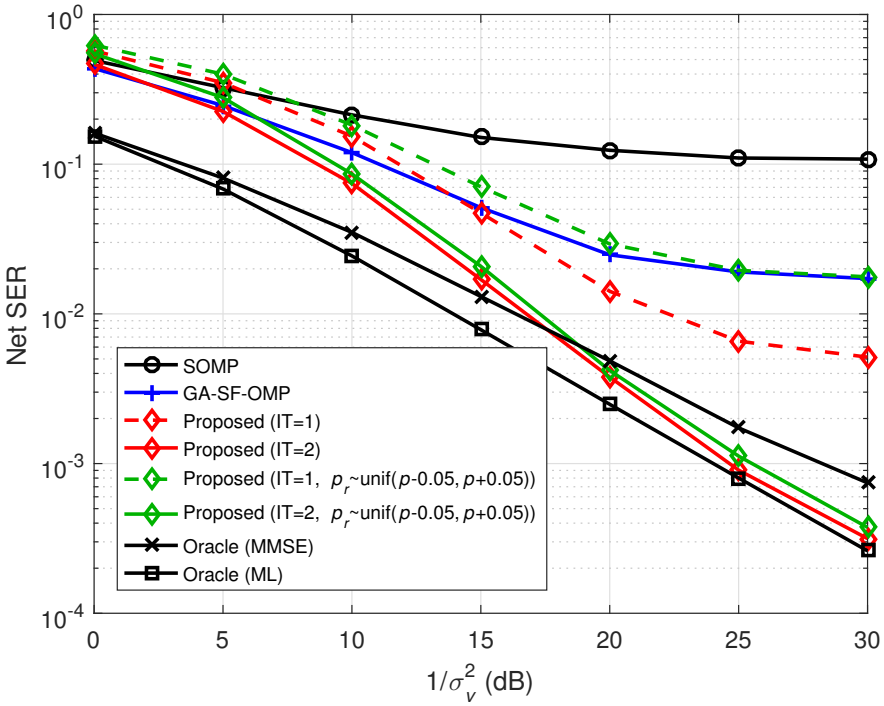
(b)

Figure 4.3: (b) Net SER when the interference covariance matrix is approximated to be diagonal.



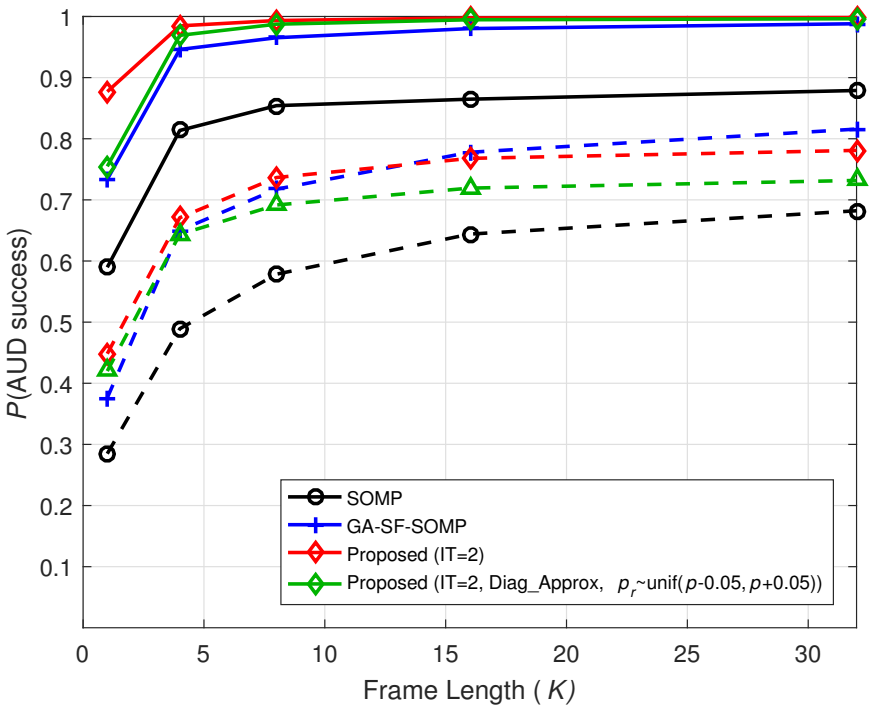
(a)

Figure 4.4: (a) AUD success probability when *a priori* user activities are incorrect.



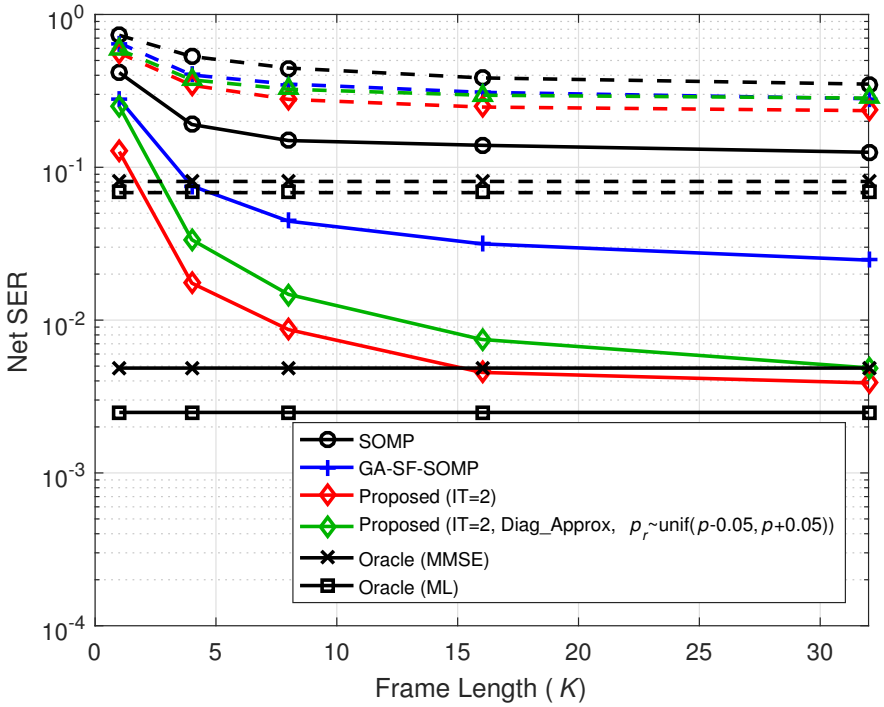
(b)

Figure 4.4: (b) Net SER when *a priori* user activities are incorrect.



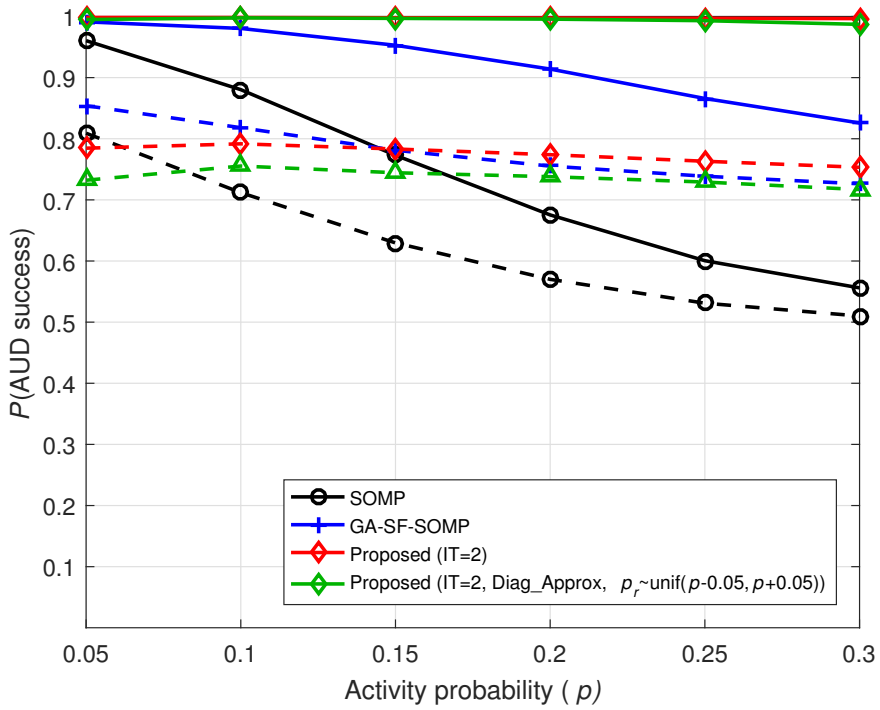
(a)

Figure 4.5: (a) AUD success probability for the various frame lengths.



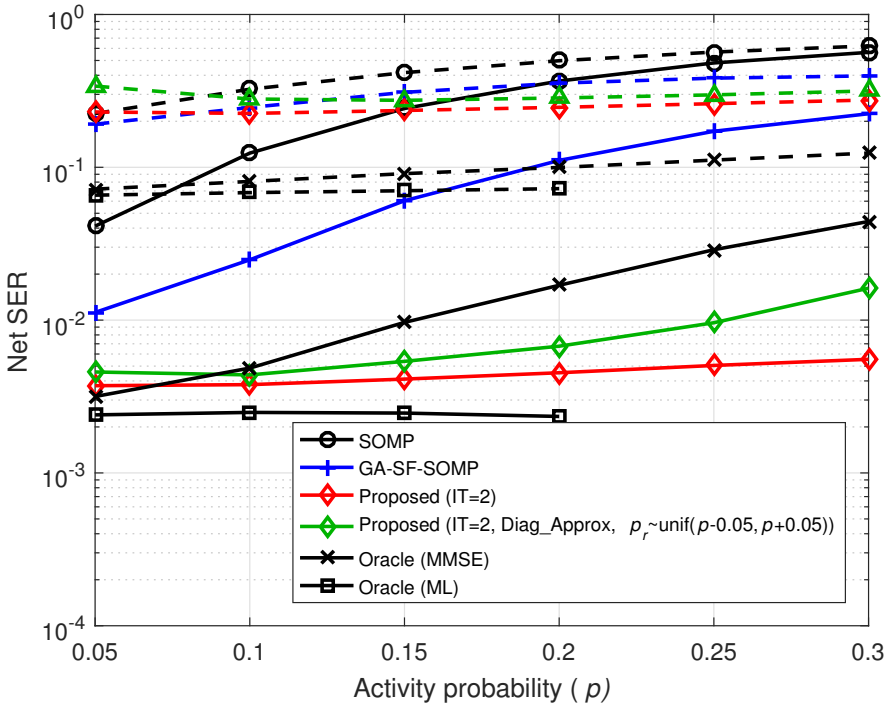
(b)

Figure 4.5: (b) Net SER for the various frame lengths.



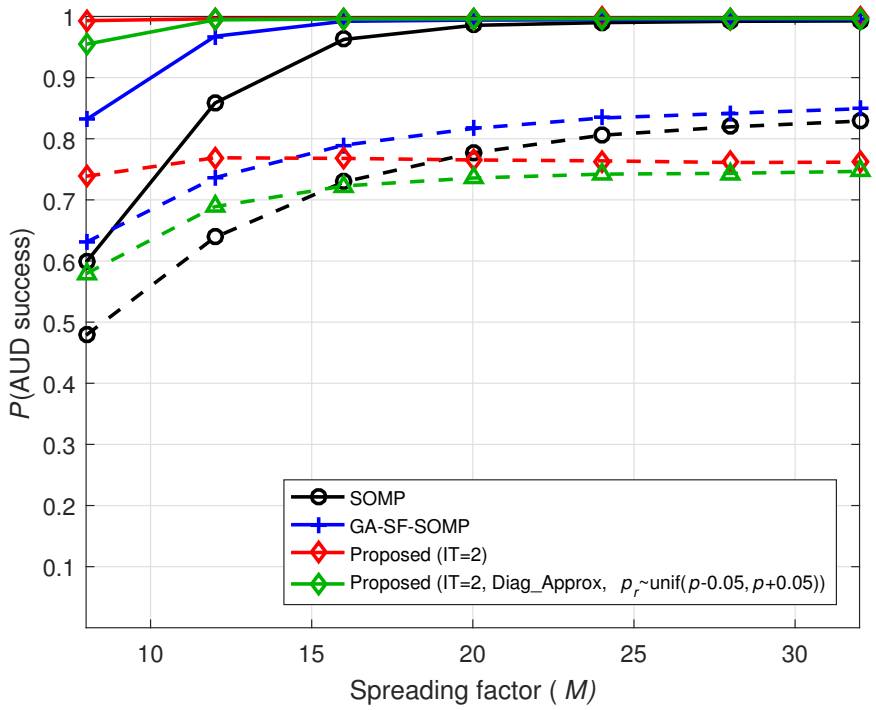
(a)

Figure 4.6: (a) AUD success probability for the various user activities.



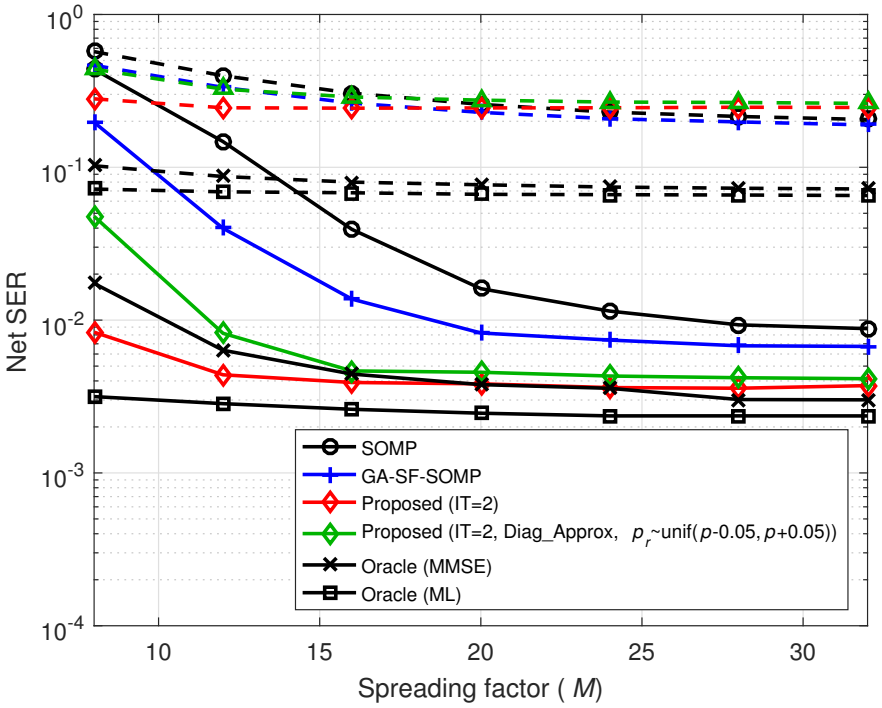
(b)

Figure 4.6: (b) Net SER for the various user activities.



(a)

Figure 4.7: (a) AUD success probability for the various spreading factors.



(b)

Figure 4.7: (b) Net SER for the various spreading factors.

4.2 Simulation Results

Fig. 4.1 shows the AUD success probability and the net SER when the group sparsity is not exploited. In this figure, the dashed and solid lines indicate the performance when $p = 0.1$ and $p = 0.05$, respectively. We clearly observe that the proposed algorithm performs best among all algorithms under test and MMSE shows very poor performance since the system is underdetermined. Overall, algorithms exploiting the *a priori* distribution outperform algorithms without exploiting it. FBPA shows quite a good performance. However, since FBPA assumes that the transmit symbol vector $\mathbf{x}$ is a *Bernoulli-Gaussian* mixture which is not necessarily correct, there is substantial performance gap from the proposed algorithm. Though the proposed algorithm performs best, the net SER gap between Oracle MMSE/ML and the proposed algorithm is quite large even in the high SNR regime. This is because the AUD is still not perfect. We can deduce from this observation that the gap can be bridged if we enhance the AUD performance by exploiting the group sparsity.

Fig. 4.2 shows the AUD and net SER performance when p is set to 0.1 and the group sparsity is exploited, where IT denotes the number of the frame iterations. Note that we do not include the results of FBPA and MMSE since they do not exploit the group sparsity. We observe that as the SNR increases, the AUD success probability of the proposed algorithm becomes close to one and the net SER approaches the Oracle ML performance. We see that two frame iterations are sufficient in achieving near optimal performance. In the low SNR regime, GA-SF-SOMP performs slightly better than the proposed algorithm. Note that GA-SF-SOMP has the perfect knowledge of the variance of interferences in each iteration, which is not possible in practice.

Fig. 4.3 shows the influence of the diagonal approximation to the computation of the covariance matrices (see Section 2.2.4). In this simulation, p is set to 0.1. We observe that the approximation does not degrade the performance in the low SNR regime. This is because the noise dominates the distortion caused by the approximation in this regime. In the high SNR regime, we see a high net SER floor caused by the

approximation. However, this performance loss is significantly reduced after two frame iterations.

Fig. 4.4 shows the tolerance of the proposed algorithm for the accuracy of p. In this simulation, p is 0.1, and each user is active with the activity probability of p on the transmission side but p_r distributed between $p-0.05$ and $p+0.05$ uniformly at random is used as the *a priori* activity probability on the reception side. Therefore, p_r can be different for each user. We observe that the performance loss caused by the inaccurate *a priori* information is reduced as the number of frame iterations increases. This is because the effect of the extrinsic activity information L_E learned from the group sparsity on the performance is higher than the *a priori* activity information L_A derived from the (erroneous) p_r after the frame iteration. Recall that the modified *a priori* $\widetilde{L}_A$ (composed of L_E and L_A) is used when the group sparsity is exploited.

Fig. 4.5 shows the performance for various frame lengths, where the dashed and solid lines indicate the performance at 5 dB and 20 dB SNR, respectively. In this simulation, we set p to 0.1 and then investigate the performance at the transition SNR (5 dB) and the AUD-saturated SNR (20 dB). We observe that the performance improves with the frame length. However, the AUD performance of SOMP is not perfect even at the 20 dB SNR. The AUD performance of GA-SF-SOMP also saturates at 0.98. This is because SOMP and GA-SF-SOMP use the correlation between the received vector and the column vectors of the channel matrix as an activity decision statistic. This statistic is not a good choice when the column vectors are highly correlated. At the 5 dB SNR, as the frame length increases, the AUD performance of GA-SF-SOMP becomes slightly better than that of the proposed algorithm. However, the net SER performance of GA-SF-SOMP is still worse.

Fig. 4.6 shows the performance when the user activity probability p varies from 0.05 to 0.3, where the dashed and solid lines indicate the performance at 5 dB and 20 dB SNR, respectively and the net SER of the oracle ML in case that $p > 0.2$ is not shown due to infeasible simulation time. We observe that as p increases, the perfor-

mance of SOMP and GA-SF-SOMP is degraded but the performance of the proposed algorithm is not affected much. When p is higher than 0.2, the diagonal approximation and the erroneous p_r deteriorate the net SER at the 20 dB SNR. This is because the interference becomes the dominant factor as more users are active. However, the performance is still much better than SOMP, GA-SF-SOMP, and even Oracle MMSE. We also observe that the performance of the proposed algorithm is close to that of the Oracle ML performance at the 20 dB SNR.

Fig. 4.7 shows the performance for the various spreading factors, where the dashed and solid lines indicate the performance at 5 dB and 20 dB SNR, respectively. In this simulation, p is set to 0.1, the number of users N is set to 64, and the spreading factor M varies from 8 to 32. We observe that the performance is improved with M because the system becomes less underdetermined. To accommodate the massive connectivity of mMTC, the performance of small M/N is important. In this case, the proposed algorithm outperforms SOMP and GA-SF-SOMP by a large margin.

Chapter 5

Conclusion

In this dissertation, we proposed a MAP-based active user and symbol detection algorithm and the extended version exploiting group sparsity, and demonstrated its performance in the mMTC scenarios. Our work is motivated by the observation that most greedy algorithms use the correlation between the modified received vector and the column vectors of the channel matrix to determine the user activity, but this correlation may not be a good decision statistic because it does not reflect the distributions of such factors as users, channels, and noise. In this work, we instead used the *a posteriori* activity probability as a decision statistic. By exploiting the finite alphabet constraint, we jointly detected the active user and its soft symbol. The soft symbol information is refined and then used as the *a priori* information for the detection of the other active users. After completing the iterations, the soft symbol information is converted into the activity information. By aggregating the activity information of the multiple received signal vectors having the common activity, we could achieve the substantial improvement in the AUD performance. In this sense, our scheme is distinct from the conventional approaches that employs the accumulated correlation of the multiple vectors to enhance the AUD performance. From numerical experiments, we demonstrated that the proposed algorithm achieves significant gain in terms of the AUD success probability and the net SER over conventional greedy algorithms.

Bibliography

[1] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 120–128, Feb. 2017.

[2] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, Y. Selen, and J. Skold, "5G wireless access: Requirements and realization," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 42–47, Dec. 2014.

[3] F. Ghavimi and H. H. Chen, "M2M communications in 3GPP LTE/LTE-A networks: Architectures, service requirements, challenges, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 525-549, Second Quarter 2015.

[4] T. Taleb and A. Kunz, "Machine type communications in 3GPP networks: Potential, challenges, and solutions," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 178–184, Mar. 2012.

[5] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: Issues and approaches," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, June 2013.

[6] L. Dai, B. Wang, Y. Yuan, S. Han, C. l. I, and Z. Wang, "Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74-81, Sept. 2015.

[7] Y. Du, B. Dong, Z. Chen, J. Fang, and L. Yang, "Shuffled multiuser detection schemes for uplink sparse code multiple access systems," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1231–1234, June 2016.

[8] Z. Ding, L. Dai, and H. V. Poor, "MIMO-NOMA design for small packet transmission in the Internet of Things," *IEEE Access*, vol. 4, pp. 1393–1405, Apr. 2016.

[9] Z. Zhang, X. Wang, Y. Zhang, and Y. Chen, "Grant-free rateless multiple access: A novel massive access scheme for Internet of Things," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 2019–2022, Oct. 2016.

[10] B. Shim and B. Song, "Multiuser detection via compressive sensing," *IEEE Commun. Lett.*, vol. 16, no. 7, pp. 972–974, July 2012.

[11] D. L. Donoho and J. Tanner, "Sparse nonnegative solution of underdetermined linear equations by linear programming," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 27, pp. 9446–9451, Mar. 2005.

[12] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Mar. 2006.

[13] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l_1-regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[14] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[15] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.

[16] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6202–6216, Dec. 2012.

[17] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sept. 2016.

[18] H. F. Schepker and A. Dekorsy, "Sparse multi-user detection for CDMA transmission using greedy algorithms," in *Proc. 8th Int. Symp. Wireless Commun. Syst.*, Aachen, Germany, Nov. 2011, pp. 291–295.

[19] S. Park, H. Seo, and B. Shim, "Joint active user detection and channel estimation for massive machine-type communications," in *Proc. IEEE 18th Workshop Signal Process. Adv. Wireless Commun.*, Sapporo, Japan, July 2017, pp. 1–5.

[20] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed sensing for wireless communications: Useful tips and tricks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1527–1550, Third Quarter 2017.

[21] C. Bockelmann, H. F. Schepker, and A. Dekorsy, "Compressive sensing based multi-user detection for machine-to-machine communication," *Trans. Emerg. Telecommun. Technol.*, vol. 24, no. 4, pp. 389–400, Apr. 2013.

[22] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process.*, vol. 86, no. 3, pp. 572–588, Mar. 2006.

[23] A. T. Abebe, and C. G. Kang, "Iterative order recursive least square estimation for exploiting frame-wise sparsity in compressive sensing-based MTC," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1018–1021, May 2016.

[24] Y. Du, B. Dong, Z. Chen, X. Wang, Z. Liu, P. Gao, and S. Li, "Efficient multi-user detection for uplink grant-free NOMA: Prior-information aided adaptive compressive sensing perspective," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 12, pp. 2812–2828, Dec. 2017.

[25] P. Schniter, L. C. Potter, and J. Ziniel, "Fast bayesian matching pursuit," in *Proc. Inform. Theory Appl. Workshop*, San Diego, USA, Jan. 2008, pp. 326–333.

[26] C. Herzet and A. Dremeau, "Bayesian pursuit algorithms," in *Proc. 18th Eur. Signal Process. Conf.*, Aalborg, Denmark, Aug. 2010, pp. 1474–1478.

[27] S. Sparrer and R. Fischer, "MMSE-based version of OMP for recovery of discrete-valued sparse signals," *Electron. Lett.*, vol. 52, no. 1, pp. 75–77, Jan. 2015.

[28] S. Barik and H. Vikalo, "Sparsity-aware sphere decoding: Algorithms and complexity analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 9, pp. 2212-2225, May 2014.

[29] J. Ahn, B. Shim, and K. B. Lee, "Sparsity-aware ordered successive interference cancellation for massive machine-type communications," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 134–137, Feb. 2018.

[30] G. Chen, J. Dai, K. Niu, and C. Dong, "Sparsity-inspired sphere decoding (SI-SD): A novel blind detection algorithm for uplink grant-free sparse code multiple access," *IEEE Access*, vol. 5, pp. 19983–19993, Sept. 2017.

[31] B. Shim, S. Kwon, and B. Song, "Sparse detection with integer constraint using multipath matching pursuit," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1851–1854, Oct. 2014.

[32] S. Sparrer and R. Fischer, "Soft-feedback OMP for the recovery of discrete-valued sparse signals," in *Proc. 23rd Eur. Signal Process. Conf.*, Nice, France, Aug. 2015, pp. 1461–1465.

[33] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[34] B. K. Jeong, B. Shim, and K. B. Lee, "A compressive sensing-based active user and symbol detection technique for massive machine-type communications," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, Calgary, Canada, Apr. 2018, pp. 6623–6627.

[35] B. K. Jeong, B. Shim, and K. B. Lee, "MAP-based active user and data detection for massive machine-type communications," *IEEE Trans. Veh. Tech.*, vol. 67, no. 9, pp. 8481–8494, Sept. 2018.

[36] R. W. Farebrother, *Linear Least Squares Computations*. New York, NY, USA: Marcel Dekker, 1988.

초 록

사물 인터넷 (Internet of Things, IoT) 시대의 도래와 함께, 대규모 사물 통신 (massive machine-type communications, mMTC)은 차세대 무선 통신 표준의 주요 요구 사항들 중의 하나가 되었다. 대규모 사물 통신 환경에서는 많은 수의 사물 기기(machine-type device)들이 대부분 비활성 상태로 데이터를 전송하지 않다가 가끔씩 활성 상태로 전환되어 작은 크기의 데이터를 전송한다. 그러므로, 기지국 (base station, BS)으로부터의 스케쥴링을 기반으로 직교(orthogonal) 시간/주파수 자원을 할당받은 후 데이터 송수신이 이루어지는 기존의 통신 방식은, 실제 전송하려는 데이터 대비 많은 부가적인 제어 정보를 필요로 하고 또한 데이터의 지연을 유발시키므로 대규모 사물 통신에 적합하지 않다. 대신, 전송단에서는 스케쥴링 없이, 즉 기지국으로 부터의 승인 없이(grant-free), 비직교 자원에 다중 접속하고(non-orthogonal multiple access, NOMA), 수신단에서는 다중 사용자 검출(multi-user detection, MUD)을 이용하여 데이터의 충돌을 복조해 내는 방식이 대규모 사물 통신에 적합하다. 이 때, 사물 기기들이 전송하는 데이터의 희소 특성을 감안하면, 압축 센싱 기반의 다중 사용자 검출 방법(compressive sensing-based multi-user detection, CS-MUD)이 일반적인 다중 사용자 검출 방법보다 더 좋은 성능을 발휘할 수 있다.

본 논문에서는, 기존 방식보다 더 좋은 성능을 가진 새로운 압축 센싱 기반의 다중 사용자 검출 방법을 제안한다. 부연 설명하면, 가장 큰 사후 활성 확률을 가진 사용자를 찾고(maximum *a posteriori* probability-based active user detection, MAP-AUD), 역시 사후 확률 관점에서 가장 확률이 높은 데이터를 추정한다(maximum *a posteriori* probability-based data detection, MAP-DD). 이 때, MAP-AUD와 MAP-

DD 블록은 서로 외재적 정보(extrinsic information)만을 주고 받는데, 이 외재적 정보는 상대방의 사전 정보가 되고, 이 사전 정보를 이용하여 다시 사후 확률 관점에서 최적의 해를 구한다. 이러한 반복 수행을 통해 각 블록은 검출의 정확도와 성능을 높여 나간다.

이 방법은 패킷 단위로 확장될 수 있다. 각각의 사물 기기들이 전송하려는 데이터는 여러개의 심볼로 구성된 패킷이며, 한 패킷 내의 각각의 심볼은 공통된 활성도를 가지게 된다. 여기서, 이 공통된 활성도를 이용하면, 활성 사물 기기와 이들의 전송 데이터 추정의 정확도를 높일 수 있다. 하지만, 이는 공동 최적화(joint optimization) 문제로 매우 복잡한 연산을 필요로 한다. 본 논문에서는, 패킷 내의 임의의 하나의 심볼에서 추정된 사물 기기의 활성도는 다른 심볼의 활성도를 추정하는데 사전 정보(*a priori* information)로 이용될 수 있다는 점에 착안하여, 복잡한 공동 최적화 문제를 비교적 연산량이 적은 부분 최적화(subproblem optimization) 문제로 단순화 시키고, 이들 간에 메시지 전달 (massage-passing) 기법을 통해 공동 최적화의 해에 근접한 해를 구하는 방법을 제안한다. 이 때, 부분 최적화 문제의 해법이 바로 앞에 설명한 MAP-AUD/MAP-DD 방법이다.

마지막으로, 모의 실험을 통해 제안하는 방법이 기존 방법과 비교했을 때 매우 크게 성능이 향상됨을 보였다. 특히, 제안하는 방법은 전체 사용자 수 대비 이용 가능한 자원이 적을 때일수록 더 큰 성능 향상이 있는데, 이는 차세대 무선 통신에서 사물 통신이 고려하는 단위 면적당 사물 기기의 수(10^6 개/km^2)를 고려했을 때, 제안하는 방법이 대규모 사물 통신에 아주 효용성이 큼을 보여준다.

주요어: 대규모 사물 통신, 압축 센싱, 비직교 다중 접속, 다중 사용자 검출
학번: 2016-30214

ACKNOWLEGEMENT