Sparsity-Exploiting Blind Receiver Algorithms for Unsourced Multiple Access in MIMO and Massive MIMO Channels

Jiaai Liu and Xiaodong Wang, *Fellow, IEEE*

***Abstract*— We propose new transmission schemes and receiver algorithms for unsourced multiple access (UMA) in MIMO and massive MIMO channels. Each active transmitter's information bits are first channel encoded. The coded bits are divided into sub-blocks and each sub-block is modulated and transmitted. For both MIMO and massive MIMO channels, the conventional nonlinear modulation can be employed where each sub-block of coded bits is mapped to a transmitted signal vector. For the massive MIMO channel, we propose a new hybrid modulation scheme to reduce the receiver complexity, where the first sub-block is nonlinearly modulated, and the subsequent sub-blocks are linearly modulated and spread by the first sub-block signal. We also propose sparsity-exploiting blind receiver algorithms. Specifically, for the MIMO case, we exploit the codeword sparsity inherent in the UMA system, and a channel clustering technique, to estimate the channel and the transmitted signal of each transmitter. For the massive MIMO, in addition to the codeword sparsity, we further exploit the channel sparsity and user sparsity in estimating the channel and transmitted signal of each transmitter. The proposed receiver algorithms for both MIMO and massive MIMO channels output either hard or soft estimates of the coded bits, and therefore single-user channel decoding of the information bits can be performed for each transmitter. Extensive simulation results are provided to demonstrate the performances of the proposed algorithms.**

***Index Terms*— Unsourced multiple access (UMA), MIMO, massive MIMO, codeword sparsity, channel sparsity, user sparsity, orthogonal matching-pursuit (OMP), matrix completion, clustering, soft demodulation.**

I. Introduction

WHILE the current cellular systems have been mainly designed for wireless services for human users, the notions of 5G and beyond incorporate fundamentally new services such as Internet-of-Things (IoT) and machine-type communications (MTC). A key requirement for future wireless systems that aim to support IoT and MTC is massive device connectivity, where a large number of devices communicate with a base-station (BS) in an uncoordinated manner. Such a paradigm of communication is known as the massive Unsourced Multiple Access (UMA) that is characterized by the following features: 1) uncoordinated or grant-free: no resource coordination or transmission scheduling is needed; 2) sporadic: only a small portion of the large number of transmitters in the network is active during any coherence time interval; 3) unsourced: the transmitters share a common codebook; and 4) small packet size and low data rate for each transmitter. Note that the unsourced attribute of the UMA system dictates that the transmitted signal from each transmitter depends only on the information it wishes to send, but not on the identity of the transmitter itself. This is in contrast to the traditional pilot-based systems where each transmitter employs a distinct pilot sequence for channel estimation. Hence in UMA systems the receiver has to be "blind." In MTC scenarios, using pilot symbols in the small data packet can incur a significant loss in spectral efficiency, which is one of the main motivation behind the pilot-free UMA systems.

A. Prior Work

Information theoretic aspects of UMA systems have been studied in [1] and [2], for the scenarios of infinite and finite data block sizes, respectively. In [3], a low-complexity UMA coding scheme, called T-fold ALOHA, is proposed based on a combination of compute-and-forward and coding for a binary adder channel. On the other hand, the design of practical UMA systems is primarily based on exploiting the inherent sparsity of the system, i.e., the number of active transmitters is far less than the size of the common codebook. However, the codebook size is usually extremely large and divide-and-conquer strategies are adopted where a codeword is split into multiple sub-blocks which effectively reduces the codebook size. For example, in [4], sub-blocks of a codeword are transmitted according to a Tanner graph and the decoding is conducted by successive interference cancellation (SIC). In [5]–[6], each sub-block of information data is encoded by adding parity bits that are dependent on previous sub-blocks and the decoding is facilitated by these parity bits.

On the receiver side, the decoding in UMA systems is typically cast as a compressed sensing (CS) problem and various sparse recovery algorithms such as LASSO, OMP and AMP

Manuscript received February 14, 2021; revised June 6, 2021 and August 9, 2021; accepted September 19, 2021. Date of publication September 29, 2021; date of current version December 16, 2021. This work was supported in part by the U.S. National Science Foundation (NSF) under grants CCF 1814803 and SHF 7995357, and in part by the U.S. Office of Naval Research (ONR) under grant N000142112155. The associate editor coordinating the review of this article and approving it for publication was L. Song. *(Corresponding author: Xiaodong Wang.)*

The authors are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: wangx@ee.columbia.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3116239.

Digital Object Identifier 10.1109/TCOMM.2021.3116239

can be employed to recover the transmitted codewords. However, existing works [3]–[6] do not consider the channel state information (CSI) and simply model the received signal as a noisy sum of all transmitted signals. The scheme in [7] adopts the transmission scheme in [5] and explores the statistical CSI, i.e., the channel covariance, in the massive MIMO scenario. In this paper, we propose new UMA decoding schemes that exploit the instantaneous CSI for both conventional MIMO and massive MIMO systems [8], [9].

B. Contributions

In this paper we propose new encoding and decoding techniques for UMA systems under both conventional MIMO block fading channels and millimeter-wave (mmWave) massive MIMO channels. In particular, we assume that the short data packet of each active transmitter is encoded using an arbitrary error-correction code and the coded bits are divided into sub-blocks and transmitted. We develop receiver algorithms that perform either hard or soft demodulation of the coded bits of each active transmitter. Then the information bits of each transmitter can be decoded using a single-user hard or soft channel decoder. Although similar to the prior works mentioned above, the data transmission is based on dividing each data packet into sub-blocks, a distinguishing feature of this work lies in the exploitation of the CSI and other types of signal sparsity, such as channel sparsity and user sparsity in mmWave massive MIMO systems. The main contributions of this paper are summarized as follows:

- For MIMO UMA systems, we propose a novel decoding algorithm that actively estimates the channel of each transmitter, which not only serves as a tag of the transmitter in order to properly assemble its sub-blocks at the receiver, but also is essential in performing soft demodulation. In our system, any channel code, such as classical algebraic codes and modern codes like LDPC or Polar codes, can be used to encode the information.
- For massive MIMO UMA systems, on the transmitter side, we consider two modulation schemes: the nonlinear modulation is the traditional way of mapping each sub-block of bits into a signal vector; and in the hybrid modulation, only the first sub-block is nonlinearly modulated, but each bit in all other sub-blocks are linearly modulated. Such a hybrid modulation can significantly reduce the receiver complexity. On the receiver side, to reduce the hardware cost, we assume that the number of RF chains is smaller than the number of antennas and each RF chain can employ a low-resolution analog-to-digital converter (ADC).
- For massive MIMO UMA systems, for both modulation schemes, we develop novel receiver algorithms that exploit the additional sparsities including the channel sparsity and the user sparsity. Here channel estimation is still the key step which is now facilitated by low-rank matrix completion.

The remainder of this paper is organized as follows. In Section II we describe the transmission schemes and the signal models for the proposed MIMO UMA and massive MIMO UMA systems. In Sections III and IV we develop the receiver algorithms for the MIMO UMA and the massive MIMO UMA systems, respectively. Simulation results are provided in Section V. Finally concluding remarks are given in Section VI.

II. System Descriptions

A. Transmitted Signals

Assume that there are totally K_{tot} transmitters in the network, and only K ($K \ll K_{tot}$) of them are transmitting data in any channel coherence time interval T using a common codebook. In particular, for a given coherence interval, we index the K active transmitters as $k = 1, 2, \ldots, K$. Each transmitter transmits M bits of information using a common codebook $\mathcal{C}$. The encoding process at each transmitter is illustrated in Fig. 1. Let $\boldsymbol{d}_k \in \{0,1\}^M$ denote the information bit vector of transmitter k which is first encoded into a (possibly interleaved) code bit vector $\boldsymbol{b}_k \in \{0,1\}^{M_c}$ using an error-correction code with rate r. (The purpose of interleaving the coded bits is to mitigate possible bursty channel errors.) Hence $M_c = M/r$. Note that any classical or modern channel code, such as algebraic codes, convolutional codes, LDPC codes, or Polar codes, can be employed to encode $\boldsymbol{d}_k$ into $\boldsymbol{b}_k$.

The code bit vector $\boldsymbol{b}_k$ is then mapped to the transmitted signal $\boldsymbol{x}_k \in \mathbb{C}^T$ through a modulation process. In order to reduce the complexity of the encoder and decoder, block processing is employed, such that the total coherence interval T is divided into J sub-intervals of length $M_s = T/J$, and in the j-th sub-interval, transmitter k transmits a signal $\boldsymbol{x}_k(j) \in \mathbb{C}^{M_s}$. Specifically, the code bit vector $\boldsymbol{b}_k$ is divided into J blocks $\boldsymbol{b}_k = [\boldsymbol{b}_k(1)^T, \ldots, \boldsymbol{b}_k(J)^T]^T$, where $\boldsymbol{b}_k(j) \in \{0,1\}^{M_b}$ with $M_b = M_c/J$. Each code bit block $\boldsymbol{b}_k(j)$ is then mapped to a symbol vector $\boldsymbol{x}_k(j) \in \mathbb{C}^{M_s}$. In this paper, we consider two types of UMA systems: MIMO UMA, where a regular MIMO receiver is employed at the BS, and massive MIMO UMA, where a massive MIMO receiver is employed at the BS. The modulation processes of mapping $\boldsymbol{b}_k(j)$ to $\boldsymbol{x}_k(j)$ for the two UMA systems are as follows.

1) MIMO UMA Modulations: Using a sensing matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N] \in \mathbb{C}^{M_s \times N}$ with $N = 2^{M_b}$, $\boldsymbol{b}_k(j)$ is mapped to the $n_k(j)$-th column of $\boldsymbol{C}$, where $n_k(j)$ is the integer whose binary expansion is given by $\boldsymbol{b}_k(j)$. Examples of sensing matrices include the Gaussian matrix, binary BCH matrix, etc. [10]. Define $\boldsymbol{\gamma}_k(j)$ as an $N \times 1$ vector with the $n_k(j)$-th entry being 1 and all other entries being 0. Hence there is a one-to-one mapping between the code bit vector $\boldsymbol{b}_k(j)$ and the indicator vector $\boldsymbol{\gamma}_k(j)$. Then the transmitted signal by transmitter k during the j-th sub-interval can be written as

$$\boldsymbol{x}_k(j) = \boldsymbol{C}\boldsymbol{\gamma}_k(j), \quad j = 1, \ldots, J, \ k = 1, \ldots, K. \tag{1}$$

Clearly this is a nonlinear modulation scheme. Since $K \ll N$, we assume that $\{n_k(j), k = 1, \ldots, K\}$ are different for each k in the j-th block. This can be justified as follows. The probability that two transmitters transmit the same information packet of size M bits is very low (e.g., $M = 70$ in our simulations. For two different information packets, after

channel encoding, the probability that sub-blocks of the two coded packets are the same is also very low. To prevent the two information packets from being identical, one can embed a few bits of ID information so that information packets from different transmitters are always different and so are their corresponding coded packets. Hence we assume that the coded bit sequences from different transmitters in each sub-block are different.

2) Massive MIMO UMA Modulations: For the massive MIMO UMA system, we consider two modulation schemes. The first one is the same nonlinear modulation scheme given by (1). The second modulation scheme is a hybrid one, where the first code bit block of each transmitter is modulated the same way as described above, i.e., $\boldsymbol{x}_k(1) = \boldsymbol{C}\boldsymbol{\gamma}_k(1), k = 1, \ldots, K$. Then for the remaining code bit blocks, linear modulation with symbol spreading is employed. Specifically, the i-th bit of $\boldsymbol{b}_k(j)$, denoted by $[\boldsymbol{b}_k(j)]_i, i = 1, \ldots, M_b$, is spread by a sequence $\boldsymbol{s}_{k,i} \in \mathbb{C}^{\frac{M_s}{M_b}}$. For each $j = 2, \ldots, J$, the transmitted signal is given by

$$\boldsymbol{x}_k(j) = \left[[\tilde{\boldsymbol{b}}_k(j)]_1 \boldsymbol{s}_{k,1}^T, \ldots, [\tilde{\boldsymbol{b}}_k(j)]_{M_b} \boldsymbol{s}_{k,M_b}^T\right]^T \in \mathbb{C}^{M_s}, \quad (2)$$

where $\tilde{\boldsymbol{b}}_k(j) = 2\boldsymbol{b}_k(j) - \mathbf{1} \in \{+1, -1\}^{M_b}$ with $\mathbf{1}$ being an all-one vector. Note that the rate of these blocks is the same as that of the first block, which is M_b code bits per sub-interval of duration $M_s = T/J$ per transmitter. The advantage of employing linear modulation is its simple demodulation process. However, since no pilot symbols are used, one block of nonlinear modulation is needed to enable the fully blind receiver, as will be seen in Sec. IV. Moreover, a blind receiver inherently has an ambiguity in associating signals to transmitters, and hence the spreading sequence $\boldsymbol{s}_k = [\boldsymbol{s}_{k,i}^T, \ldots, \boldsymbol{s}_{k,M_b}^T]^T \in \mathbb{C}^{M_s}$ should be chosen as data dependent rather than transmitter dependant. To that end, for each transmitter k, we use the transmitted signal in the first block as the spreading sequence for subsequent blocks, i.e.,

$$\boldsymbol{s}_k = \boldsymbol{x}_k(1), \quad (3)$$

$$\text{or } \boldsymbol{s}_{k,i} = \boldsymbol{x}_k(1)\left(\frac{(i-1)M_s}{M_b} + 1 : \frac{iM_s}{M_b}\right), \quad i = 1, \ldots, M_b. \quad (4)$$

To avoid two transmitters using the same spreading sequence, we assume that the transmitted signals in the first block by different transmitters are different, which can be realized by, e.g., including the ID information using the first few bits in each information bit vector $\boldsymbol{d}_k$. The receiver first estimates $\boldsymbol{x}_k(1)$ in the first block, based on which it then despreads the subsequent blocks and demodulates the corresponding code bits.

B. MIMO Channel Models

We consider an uplink system where each transmitter has a single transmit antenna, and the base station employs N_a receive antennas. The $N_a \times K$ channel matrix is denoted as $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K]$, where $\boldsymbol{h}_k$ is the channel response vector between the k-th transmitter and the base station. For MIMO UMA, $\boldsymbol{H}$ is an unknown matrix without additional structure.

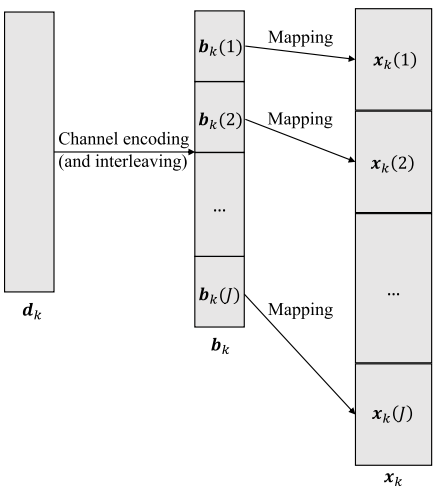


Fig. 1. The block encoding scheme.

On the other hand, for massive MIMO UMA, we assume that the system operates in the mmWave band and the channel vector $\boldsymbol{h}_k$ during the coherence interval T is described by the geometric wide-band model [11]

$$\boldsymbol{h}_k = \sum_{l=1}^{L} \alpha_{k,\ell} \boldsymbol{a}_R(\phi_{k,\ell}) \in \mathbb{C}^{N_a \times 1}, \quad (5)$$

where L is the number of scatterers, $\alpha_{k,\ell} \sim \mathcal{CN}(0, \sigma_{k,\ell}^2)$ is the complex gain of the ℓ-th path of the k-th transmitter with average power gain $\sigma_{k,\ell}^2$. Assuming that the receive antenna arrays are installed horizontally, let $\phi_{k,\ell} = \sin(\theta_{k,\ell}) \in [-1, 1]$ denote the direction of arrival (DoA) of the ℓ-th path of transmitter k, where $\theta_{k,\ell}$ is the physical azimuth. The normalized receive antenna array response at the base station is given by

$$\boldsymbol{a}_R(\phi) = \frac{1}{\sqrt{N_a}}\left[1, e^{j\frac{2\pi}{\lambda}d\phi}, \ldots, e^{j(N_a-1)\frac{2\pi}{\lambda}d\phi}\right]^T \in \mathbb{C}^{N_a \times 1}, \quad (6)$$

where λ is the wavelength and d ($d \geq \lambda/2$) is the inter-antenna element spacing.

Define a channel dictionary matrix

$$\tilde{\boldsymbol{A}}_R = \left[\boldsymbol{a}_R(\tilde{\phi}_1), \boldsymbol{a}_R(\tilde{\phi}_2), \ldots, \boldsymbol{a}_R(\tilde{\phi}_{\tilde{N}})\right] \in \mathbb{C}^{N_a \times \tilde{N}}, \quad (7)$$

where $\{\tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_{\tilde{N}}\}$ denotes a set of $\tilde{N}$ points uniformly spaced in $[-1, 1]$. We set the value of $\tilde{N}$ sufficiently large ($\tilde{N} \gg L$) such that the real channel angles $\{\phi_{k,\ell}\}$ can be well approximated by elements in the dictionary. Under this assumption, the channel vector in (5) can be rewritten as

$$\boldsymbol{h}_k = \tilde{\boldsymbol{A}}_R \tilde{\boldsymbol{\alpha}}_k, \quad (8)$$

where $\tilde{\boldsymbol{\alpha}}_k \in \mathbb{C}^{\tilde{N} \times 1}$ is an L-sparse vector, i.e., it has only L non-zero elements corresponding to $\{\alpha_{k,\ell}, \ell = 1, \ldots, L\}$ in (5).

C. Received Signals

The signal arriving at the N_a base station receive antennas during the j-th sub-interval is given by

$$\begin{aligned}\boldsymbol{Y}(j) &= \sum_{k=1}^{K} \boldsymbol{h}_k \boldsymbol{x}_k(j)^T + \boldsymbol{Z}(j) \\ &= \boldsymbol{H}\boldsymbol{X}(j) + \boldsymbol{Z}(j) \in \mathbb{C}^{N_a \times M_s},\ j = 1, \ldots, J, \quad (9)\end{aligned}$$

where $\boldsymbol{Z}(j) \in \mathbb{C}^{N_a \times M_s}$ is the additive white Gaussian noise (AWGN) matrix, i.e., its elements are i.i.d. $\mathcal{CN}(\boldsymbol{0}, \sigma^2)$,

$$\begin{aligned}\boldsymbol{H} &= [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K] \in \mathbb{C}^{N_a \times K} \\ &= \tilde{\boldsymbol{A}}_R [\tilde{\boldsymbol{\alpha}}_1, \ldots, \tilde{\boldsymbol{\alpha}}_K] \text{ for geometric model}, \quad (10)\end{aligned}$$

$$\text{and } \boldsymbol{X}(j) = [\boldsymbol{x}_1(j), \ldots, \boldsymbol{x}_K(j)]^T \in \mathbb{C}^{K \times M_s}, \quad (11)$$

where for nonlinear modulation, $\boldsymbol{x}_k(j)$ is given by (1); and for hybrid modulation, $\boldsymbol{x}_k(1)$ is given by (1) with $j = 1$, and $\boldsymbol{x}_k(j)$ is given by (2) for $j = 2, \ldots, J$. Denote $\boldsymbol{X} = [\boldsymbol{X}(1), \ldots, \boldsymbol{X}(J)]$, $\boldsymbol{Y} = [\boldsymbol{Y}(1), \ldots, \boldsymbol{Y}(J)]$ and $\boldsymbol{Z} = [\boldsymbol{Z}(1), \ldots, \boldsymbol{Z}(J)]$. Then (9) can be rewritten as

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{Z} \in \mathbb{C}^{N_a \times T}. \quad (12)$$

When there are a large number of receive antennas at the base station, i.e., massive MIMO, typically an analog structure with N_r ($N_r < N_a$) RF chains is employed to combine the incoming signal in the RF band. Define a connection matrix $\boldsymbol{E} \in \{0, 1\}^{N_a \times T}$ such that $\boldsymbol{E}(i, t) = 1$, if the i-th antenna is connected to an RF chain at time t, and $\boldsymbol{E}(i, t) = 0$ otherwise. Hence each column of $\boldsymbol{E}$ has exactly N_r 1's. If the analog combining is realized through antenna selection, i.e., at each time, N_r out of N_a antennas are randomly connected to RF chains, then the observed received signal during the entire coherence time interval is given by [12]

$$\boldsymbol{Y}_o = \boldsymbol{E} \circ \boldsymbol{Y} = \boldsymbol{E} \circ (\boldsymbol{H}\boldsymbol{X} + \boldsymbol{Z}) \in \mathbb{C}^{N_a \times T}, \quad (13)$$

where $\circ$ denotes the Hadamard product.

On the other hand, if the RF combing is realized through DFT phase shifters, the observed received signal can be written as [12]

$$\tilde{\boldsymbol{Y}}_o = \boldsymbol{E} \circ (\boldsymbol{F}\boldsymbol{Y}) = \boldsymbol{E} \circ (\tilde{\boldsymbol{H}}\boldsymbol{X} + \tilde{\boldsymbol{Z}}) \in \mathbb{C}^{N_a \times T}, \quad (14)$$

where $\tilde{\boldsymbol{H}} = \boldsymbol{F}\boldsymbol{H}$, $\tilde{\boldsymbol{Z}} = \boldsymbol{F}\boldsymbol{Z}$, and $\boldsymbol{F}$ is the $N_a \times N_a$ DFT matrix with

$$\boldsymbol{F}(m, n) = \frac{1}{\sqrt{N_a}} e^{-j\frac{2mn\pi}{N_a}}, \quad m, n = 0, \ldots, N_a - 1. \quad (15)$$

Note that since $\boldsymbol{F}$ is unitary, $\tilde{\boldsymbol{Z}}$ still contains i.i.d. $\mathcal{CN}(\boldsymbol{0}, \sigma^2)$ samples.

D. Overview of Proposed Algorithms

Our goal is to decode the information bit vectors $\boldsymbol{d}_k, k = 1, \ldots, K$, based on the observed signal $\boldsymbol{Y}$ in (12) in MIMO UMA, or $\boldsymbol{Y}_o$ in (13) ($\tilde{\boldsymbol{Y}}_o$ in (14)) in massive MIMO UMA, and the prior knowledge of the common code book (i.e., the channel code and the modulation scheme). Note that in stark contrast to conventional communication systems, in the system described above no pilot symbols are needed to estimate the unknown channels $\boldsymbol{H}$. In fact, $\boldsymbol{H}$ will be estimated by exploiting various signal sparsity properties in this system. In particular, in the next two sections, three types of signal sparsities will be exploited leading to decoding algorithms for both MIMO UMA and massive MIMO UMA. The following is a brief overview.

1) For MIMO UMA, we exploit the *codeword sparsity*. That is, the number of codewords in the sensing matrix $\boldsymbol{C} \in \mathbb{C}^{M_s \times N}$ is much larger than the number of active transmitters, i.e., $N = 2^{M_b} \gg K$. The MIMO UMA decoding algorithm (Alg. 1) exploits such codeword sparsity and a clustering technique to decode all transmitted codewords $\{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K\}$ based on the received signal $\boldsymbol{Y}$ in (12).
2) For massive MIMO UMA, in addition to codeword sparsity, we also exploit the *channel sparsity* exhibited by the mmWave channel model given in (8). That is, the vector $\tilde{\boldsymbol{\alpha}}_k \in \mathbb{C}^{\tilde{N}}$ satisfies $\|\tilde{\boldsymbol{\alpha}}_k\|_0 = L \ll \tilde{N}$. Moreover, the third type of sparsity is *user sparsity* in massive MIMO systems. That is, the number of active transmitters is much less than the number of receive antennas and the coherence time interval, i.e., $K \ll \min\{N_a, T\}$. Then the signal component $\boldsymbol{H}\boldsymbol{X}$ of the received signal matrix $\boldsymbol{Y}$ in (12) is a low-rank matrix. The massive MIMO UMA decoding algorithm (Alg. 2) exploits all three types of sparsities to decode all transmitted codewords $\{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K\}$ based on the partially observed received signal $\boldsymbol{Y}_o$ in (13) ($\tilde{\boldsymbol{Y}}_o$ in (14)).

III. MIMO UMA Decoding Based on Codeword Sparsity

In this section, we consider the problem of estimating $\boldsymbol{X}(j)$ from $\boldsymbol{Y}(j)$ in (9), without assuming any structure on the unknown channel matrix $\boldsymbol{H}$. Note that since no pilot symbols are used, any estimate has an inherent permutation ambiguity. That is, for each $j = 1, \ldots, J$, any algorithm can only provide an estimate of a column-permuted channel matrix $\boldsymbol{H}$ and the corresponding row-permuted codeword matrix $\boldsymbol{X}$, i.e., $\hat{\boldsymbol{H}}(j)\boldsymbol{\Pi}_j$ and $\boldsymbol{\Pi}_j^T \hat{\boldsymbol{X}}(j)$, where $\boldsymbol{\Pi}_j$ is a $K \times K$ permutation matrix that satisfies $\boldsymbol{\Pi}_j \boldsymbol{\Pi}_j^T = \boldsymbol{I}$. In order to estimate the permutations $\boldsymbol{\Pi}_j, j = 1, \ldots, J$, we make use of the fact that for each true channel $\boldsymbol{h}_k$, each $\hat{\boldsymbol{H}}(j)\boldsymbol{\Pi}_j$ contains one column that corresponds to an estimate of $\boldsymbol{h}_k$; and therefore these J columns corresponding to $\boldsymbol{h}_k$ form a cluster in $\mathbb{C}^{N_a}$ that is centered around $\boldsymbol{h}_k$. Then by applying a clustering algorithm we can identify the permutations $\{\boldsymbol{\Pi}_j, j = 1, \ldots, J\}$ and obtain the final channel estimate $\hat{\boldsymbol{H}}$.

A. Block CS Decoding

To proceed, we substitute (11) into (9), then the received signal at the j-th sub-interval is given by

$$\begin{aligned}\boldsymbol{Y}(j)^T &= \boldsymbol{X}(j)^T \boldsymbol{H}^T + \boldsymbol{Z}(j)^T \in \mathbb{C}^{M_s \times N_a} \\ &= \underbrace{\boldsymbol{C}\,[\boldsymbol{\gamma}_1(j) \cdots \boldsymbol{\gamma}_K(j)] \begin{bmatrix} \boldsymbol{h}_1^T \\ \vdots \\ \boldsymbol{h}_K^T \end{bmatrix}}_{\boldsymbol{\Gamma}(j) \in \mathbb{C}^{N \times N_a}} + \boldsymbol{Z}(j)^T\end{aligned}$$

$$= \boldsymbol{C}\boldsymbol{\Gamma}(j) + \boldsymbol{Z}(j)^T. \quad (16)$$

Note that $\boldsymbol{\Gamma}(j)$ is row sparse, i.e., it has K non-zero rows corresponding to $\boldsymbol{h}_k^T, k = 1, \ldots, K$. Hence by estimating the row-sparse $\boldsymbol{\Gamma}(j)$, we obtain the estimates of both the channel $\boldsymbol{h}_k$ and the transmitted signal $\boldsymbol{x}_k(j) = \boldsymbol{C}\boldsymbol{\gamma}_k(j)$ of all active transmitters $k = 1, \ldots, K$. We use the Simultaneous Orthogonal Matching Pursuit (S-OMP) algorithm [13] to estimate $\boldsymbol{\Gamma}(j)$, as summarized in Alg. 1(a).

In each iteration, it identifies one column of $\boldsymbol{C}$ that has the largest contribution (inner product) to the current residual $\boldsymbol{R}$, and adds the column index n^* to the index set $\mathcal{I}$. It then computes the corresponding coefficients $\boldsymbol{\Sigma}$ of the columns indexed by $\mathcal{I}$ in $\boldsymbol{C}$, i.e., $\boldsymbol{C}(:, \mathcal{I})$, and updates the residual $\boldsymbol{R}$. Note that $\boldsymbol{\Sigma}$ corresponds to an estimate of $\boldsymbol{\Gamma}(j)$ in (16). Hence after the last iteration, the indices $\{\hat{n}_1(j), \ldots, \hat{n}_K(j)\}$ corresponding to the rows of $\boldsymbol{\Sigma}$ with the largest norms are selected as the estimates of the transmitted codeword indices, and the corresponding channel estimates are $\{\hat{\boldsymbol{g}}_1(j), \ldots, \hat{\boldsymbol{g}}_K(j)\}$.

The computational complexity of Alg. 1(a) is dominated by the pseudo-inverse operations, which is $\mathcal{O}\left(M_s^3\right)$ in each iteration. Hence the complexity of Alg. 1(a) is $\mathcal{O}\left(NM_s^3\right)$.

Algorithm 1(a): S-OMP for Estimating $\boldsymbol{\Gamma}(j)$ in (16)

Input: Received signal in the j-th block $\boldsymbol{Y}(j) \in \mathbb{C}^{N_a \times M_s}$, sensing matrix $\boldsymbol{C} \in \mathbb{C}^{M_s \times N}$

Output: Estimated channels $\{\hat{\boldsymbol{g}}_1(j), \ldots, \hat{\boldsymbol{g}}_K(j)\}$, indices of transmitted codewords $\{\hat{n}_1(j), \ldots, \hat{n}_K(j)\}$

Initialization: $\boldsymbol{R} = \boldsymbol{Y}(j)^T, \mathcal{I} = \emptyset$;

for $t = 1, 2, \ldots, N$ **do**

$n^* = \text{argmax}_{n \in \{1, \ldots, N\}} \frac{\|\boldsymbol{R}^H \boldsymbol{C}(:,n)\|_2}{\|\boldsymbol{C}(:,n)\|_2}$;

$\mathcal{I} \leftarrow \mathcal{I} \cup \{n^*\}$;

$\boldsymbol{\Sigma} = \boldsymbol{C}(:, \mathcal{I})^{\dagger} \boldsymbol{Y}(j)^T$, where $\dagger$ is the pseudo-inverse operator ;

$\boldsymbol{R} = \boldsymbol{Y}(j)^T - \boldsymbol{C}(:, \mathcal{I})\boldsymbol{\Sigma}$.

end

Find K rows of $\boldsymbol{\Sigma}$ with the largest ℓ_2 norm: $\{\hat{\boldsymbol{g}}_1(j)^T = \boldsymbol{\Sigma}(\hat{n}_1(j), :), \ldots, \hat{\boldsymbol{g}}_K(j)^T = \boldsymbol{\Sigma}(\hat{n}_K(j), :)\}$ and their corresponding indices $\{\hat{n}_1(j), \ldots, \hat{n}_K(j)\}$.

B. Resolving Permutation Ambiguity by Channel Clustering

After running the S-OMP algorithm for J blocks, we obtain J permuted versions of estimates of true channels $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K$, $\{\hat{\boldsymbol{g}}_1(j), \ldots, \hat{\boldsymbol{g}}_K(j), j = 1, \ldots, J\}$. Let the transmitter ordering be the one corresponding to $j = 1$, i.e.,

$$\hat{\boldsymbol{h}}_k(1) = \hat{\boldsymbol{g}}_k(1), \quad (17)$$

$$\hat{\boldsymbol{x}}_k(1) = \boldsymbol{C}(:, \hat{n}_k(1)), \quad k = 1, \ldots, K. \quad (18)$$

Let $\pi_j = [\pi_j(1), \ldots, \pi_j(K)]$ denote the permutation corresponding to block $j \in \{2, \ldots, J\}$ with respect to block $j = 1$. Then we can write

$$\hat{\boldsymbol{h}}_k(j) = \hat{\boldsymbol{g}}_{\pi_j(k)}(j) \quad (19)$$

$$\hat{\boldsymbol{x}}_k(j) = \boldsymbol{C}(:, \hat{n}_{\pi_j(k)}(j)), \quad k = 1, \ldots, K; \; j = 2, \ldots, J. \quad (20)$$

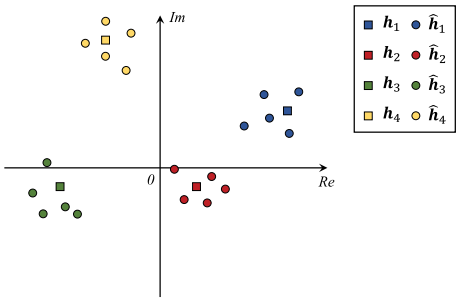


Fig. 2. An example of channel clustering when $N_a = 1, K = 4, J = 5$.

Hence we must identify the permutations $\{\pi_j, j = 2, \ldots, J\}$ in order to form the estimate of the transmitted signal $\boldsymbol{x}_k = \left[\boldsymbol{x}_k(1)^T, \ldots, \boldsymbol{x}_k(J)^T\right]^T$, i.e.,

$$\hat{\boldsymbol{x}}_k = \left[\hat{\boldsymbol{x}}_k(1)^T, \hat{\boldsymbol{x}}_{\pi_2(k)}(2)^T, \ldots, \hat{\boldsymbol{x}}_{\pi_J(k)}(J)^T\right], \quad k = 1, \ldots, K. \quad (21)$$

Then $\hat{\boldsymbol{x}}_k$ will be used to decode the transmitted information bit vector $\boldsymbol{d}_k$. In order to identify the permutations $\{\pi_j, j = 2, \ldots, J\}$, we note that for a given k, $\{\hat{\boldsymbol{g}}_k(1), \hat{\boldsymbol{g}}_{\pi_j(k)}(j), j = 2, \ldots, J\}$ is a set of J estimates of the same channel vector $\boldsymbol{h}_k$, and therefore it forms a cluster in the N_a-dimensional space $\mathbb{C}^{N_a}$ centered at $\boldsymbol{h}_k$. Fig. 2 shows an example of the true and estimated channels when $N_a = 1, K = 4, J = 5$, where each square represents a true channel and each circle represents an estimated channel; and different colors represent different transmitters. It is seen that there are $K = 4$ clusters; and each cluster contains $J = 5$ points corresponding to the estimated channels, $\{\hat{\boldsymbol{h}}_k(1), \ldots, \hat{\boldsymbol{h}}_k(5)\}$ that are centered at the true channel $\boldsymbol{h}_k$. Consequently the JK_a channel estimates $\{\hat{\boldsymbol{g}}_k(j), k = 1, \ldots, K, j = 1, \ldots, J\}$ output by Alg. 1(a) form K clusters in $\mathbb{C}^{N_a}$, each consisting of J points centered at $\boldsymbol{h}_k$. Then by applying a clustering algorithm, summarized as Alg. 1(b), we can identify the permutations.

In Alg. 1(b), let columns of $\hat{\boldsymbol{H}}$ denote the centers of the K clusters, which are initialized by the estimated channels of the first block $\hat{\boldsymbol{g}}_k(1), k = 1, \ldots, K$. At the j-th outer iteration, $j = 2, \ldots, J$, it computes a $K \times K$ distance matrix $\boldsymbol{D}$ with each element denoting the Euclidean distance between any one of the current K cluster centers and any one of the K estimated channels $\hat{\boldsymbol{g}}_k(j)$ in the j-th block. Then in the k-th inner iteration, $k = 1, \ldots, K$, it picks one element $\hat{\boldsymbol{g}}_{q^*}(j)$ in $\{\hat{\boldsymbol{g}}_1(j), \ldots, \hat{\boldsymbol{g}}_K(j)\}$ that has not been assigned to a cluster and has the minimum distance to its cluster center $\boldsymbol{H}(:, p^*)$, and assigns $\pi_j(p^*) = q^*$. It then updates the center of cluster p^* by

$$\hat{\boldsymbol{H}}(:, p^*) = ((j-1)\hat{\boldsymbol{H}}(:, p^*) + \hat{\boldsymbol{g}}_{q^*}(j))/j, \quad (22)$$

which is the mean of j estimated channel vectors in one cluster. It also sets the p^*-th row and the q^*-th column of $\boldsymbol{D}$ to infinity to avoid them being selected again. Note that Alg. 1(b) returns both the permutations $\{\pi_j, j = 2, \ldots, J\}$ and the final channel estimate $\hat{\boldsymbol{H}} = \left[\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_K\right]$, where $\hat{\boldsymbol{h}}_k$

is the mean of the estimates of $\boldsymbol{h}_k$ in J blocks, i.e.,

$$\hat{\boldsymbol{h}}_k = \frac{1}{J}\left(\hat{\boldsymbol{g}}_k(1) + \sum_{j=2}^{J} \hat{\boldsymbol{g}}_{\pi_j(k)}(j)\right). \tag{23}$$

Algorithm 1(b): Identifying Permutations $\{\pi_j, j = 2, \ldots, J\}$ by Clustering

Input: JK channel estimates from Algorithm 1(a) $\{\hat{\boldsymbol{g}}_k(j), k = 1, \ldots, K; j = 1, \ldots, J\}$
Output: Permutations $\pi_j, j = 2, \ldots, J$, $\hat{\boldsymbol{H}} = \left[\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_K\right]$
Initialization: $\boldsymbol{D} = \boldsymbol{0}_{K\times K}$, $\hat{\boldsymbol{H}} = \boldsymbol{0}_{N_a\times K}$, $\hat{\boldsymbol{H}}(:,k) = \hat{\boldsymbol{g}}_k(1)$ for $k = 1, 2, \ldots, K$;
for $j = 2, \ldots, J$ **do**
  $\boldsymbol{D}(k_1, k_2) = \|\hat{\boldsymbol{H}}(:,k_1) - \hat{\boldsymbol{g}}_{k_2}(j)\|_2$ for all $k_1, k_2 = 1, \ldots, K$;
  for $k = 1, \ldots, K$ **do**
    $p^*, q^* = \text{argmin}_{p,q=1,\ldots,K} \boldsymbol{D}(p,q)$;
    $\pi_j(p^*) = q^*$;
    Update the center of cluster p^* according to (22);
    $\boldsymbol{D}(p^*,:) = \infty$, and $\boldsymbol{D}(:,q^*) = \infty$;
  end
end

The computational complexity of Alg. 1(b) is $\mathcal{O}\left(JK^2\ N_a\right)$. Given the permutations $\{\pi_j, j = 2, \ldots, J\}$, the estimate of the transmitted signal $\hat{\boldsymbol{x}}_k$ is then given by (18) and (20)-(21), based on which demodulation and decoding can be performed to obtain the transmitted data $\boldsymbol{d}_k$. On the other hand, the estimated channel $\hat{\boldsymbol{H}}$ will be used by the soft demodulator in (27)-(28).

C. Single-User Demodulation and Decoding

For each transmitter k, based on the recovered transmitted signal $\hat{\boldsymbol{x}}_k$ in (21), we can then perform either hard or soft demodulation and decoding to obtain an estimate of its information bit vector $\boldsymbol{d}_k$. In particular, for hard decision, we first perform demapping to obtain the estimated code bits $\hat{\boldsymbol{b}}_k$ as follows:

$$\hat{\boldsymbol{b}}_k(1) = \text{ binary expansion of } \hat{n}_k(1), \tag{24}$$

$$\hat{\boldsymbol{b}}_k(j) = \text{ binary expansion of } \hat{n}_{\pi_j(k)}(j),\ j = 2, \ldots, J, \tag{25}$$

$$\hat{\boldsymbol{b}}_k = \left[\hat{\boldsymbol{b}}_k(1)^T, \ldots, \hat{\boldsymbol{b}}_k(J)^T\right]^T \in \{0,1\}^{M_c}. \tag{26}$$

Note that the complexity of the above hard demodulation is linear in terms of the total number of code bits, i.e., $\mathcal{O}(M_b)$ *per block per transmitter*. Then we perform hard channel decoding using $\hat{\boldsymbol{b}}_k$ to obtain the estimated information bit vector $\hat{\boldsymbol{d}}_k$.

On the other hand, for soft decision, we need the log-likelihood ratio (LLR) of each bit of $\boldsymbol{b}_k$ and then decode $\boldsymbol{d}_k$ using a soft channel decoder. To compute the LLR of each code bit, we first form the noisy received signal from the k-th transmitter by subtracting from the received signal $\boldsymbol{Y}(j)$ the signal components from all other transmitters, i.e.,

$$\begin{aligned}\boldsymbol{Y}_k(j) &= \boldsymbol{Y}(j) - \sum_{k'\neq k} \hat{\boldsymbol{h}}_{k'}\hat{\boldsymbol{x}}_{k'}(j)^T \\ &\approx \hat{\boldsymbol{h}}_k \boldsymbol{x}_k(j)^T + \boldsymbol{Z}(j) \in \mathbb{C}^{N_a\times M_s}.\end{aligned} \tag{27}$$

Then the probability that the transmitted signal $\boldsymbol{x}_k(j)$ equals to the n-th column of the codebook $\boldsymbol{C}$ can be written as

$$\begin{aligned}&P\big(\boldsymbol{x}_k(j) = \boldsymbol{C}(:,n) \mid \boldsymbol{Y}_k(j)\big) \\ &\qquad \propto \exp\left(\underbrace{-\frac{\|\boldsymbol{Y}_k(j) - \hat{\boldsymbol{h}}_k\boldsymbol{C}(:,n)^T\|_{\mathcal{F}}^2}{\sigma^2}}_{\beta_{k,j,n}}\right), \\ &\qquad n = 1, \ldots, N = 2^{M_b},\end{aligned} \tag{28}$$

where $\|\cdot\|_{\mathcal{F}}^2$ denotes the Frobenius norm. Denote $[\boldsymbol{b}]_i$ as the i-th element of the vector $\boldsymbol{b}$, and $\text{bin}(n,i)$ as the i-th bit of the binary expansion of integer n. Then the LLR of the i-th bit of $\boldsymbol{b}_k(j)$ can be computed as

$$\begin{aligned}\text{LLR}_k(j,i) &\triangleq \log\frac{P\big([\boldsymbol{b}_k(j)]_i = 0 \mid \boldsymbol{Y}_k(j)\big)}{P\big([\boldsymbol{b}_k(j)]_i = 1 \mid \boldsymbol{Y}_k(j)\big)} \\ &= \log\frac{\sum_{n:\text{bin}(n,i)=0}\exp\left(\beta_{k,j,n}\right)}{\sum_{n:\text{bin}(n,i)=1}\exp\left(\beta_{k,j,n}\right)} \\ &\approx \max_{n:\text{bin}(n,i)=0}\beta_{k,j,n} - \max_{n:\text{bin}(n,i)=1}\beta_{k,j,n}, \\ &\qquad i = 1, \ldots, M_b,\ j = 1, \ldots, J.\end{aligned} \tag{29}$$

And the LLRs for all code bits in $\boldsymbol{b}_k$ are

$$\begin{aligned}\text{LLR}(\boldsymbol{b}_k) = [&\text{LLR}_k(1,1), \ldots, \text{LLR}_k(1,M_b), \\ &\text{LLR}_k(2,1), \ldots, \text{LLR}_k(J,M_b)]^T \in \mathbb{R}^{M_c},\end{aligned} \tag{30}$$

based on which a soft channel decoder can then compute the LLRs of the information bits $\text{LLR}(\boldsymbol{d}_k)$. Note that the above soft demodulator has a computational complexity of $\mathcal{O}\left(2^{M_b}\right) = \mathcal{O}(N)$ *per block per transmitter*.

Finally the MIMO UMA decoding algorithm is summarized in Alg. 1. Note that its total computational complexity consists of three parts: the complexity of decoding all J blocks using Alg. 1(a), which is $\mathcal{O}\left(J2^{M_b}M_s^3\right)$; the complexity of identifying the permutations using Alg. 1(b), which is $\mathcal{O}\left(JK^2N_a\right)$; and the complexity of single-user demodulation for all K transmitters, which is $\mathcal{O}(JKM_b)$ for hard demodulation and $\mathcal{O}\left(JK2^{M_b}\right)$ for soft demodulation.

D. Discussions

Our approach to resolving different permutations over codeword blocks resulted from the CS decoding is through simple channel matching, and Alg. 1(b) has a complexity of $\mathcal{O}(JK^2N_a)$. Moreover, since the entire transmitted signal $\boldsymbol{x}_k$ is recovered prior to decoding data, single-user decoding can be performed for each active transmitter k independently.

In contrast, in existing works [5], [6], [14], the effect of channel is ignored; in fact, a unit scalar channel gain is assumed for all active transmitters, i.e., $h_k = 1, k = 1, \ldots, K$. Then in order to decode the data of all active transmitters

Algorithm 1: MIMO UMA Decoding Algorithm Based on Codeword Sparsity

Input: Received signals in a coherence time interval $\boldsymbol{Y}$, sensing matrix $\boldsymbol{C}$
Output: Decoded information bits of all active transmitters $\hat{\boldsymbol{d}}_k$ or $\mathrm{LLR}(\boldsymbol{d}_k)$, $k = 1, \ldots, K$
Run Alg. 1(a) to obtain $\{\hat{\boldsymbol{g}}_1(j), \ldots, \hat{\boldsymbol{g}}_K(j)\}$, $\{\hat{n}_1(j), \ldots, \hat{n}_K(j)\}, j = 1, \ldots, J$;
Run Alg. 1(b) to obtain $\pi_j, j = 2, \ldots, J$ and $\hat{\boldsymbol{H}}$;
for $k = 1, \ldots, K$ **do**
  Hard decoder: perform hard demodulation according to (24)-(26) to obtain $\hat{\boldsymbol{b}}_k$, and then perform hard decoding to get $\hat{\boldsymbol{d}}_k$;
  Soft decoder: perform soft demodulation according to (20), (27)-(30) to obtain $\mathrm{LLR}(\boldsymbol{b}_k)$, and then perform soft decoding to get $\mathrm{LLR}(\boldsymbol{d}_k)$.
end

in the presence of different permutations of the transmitted signals, joint decoding must be performed that has a very high complexity. In [5], the channel code is the simple parity-check code. Each $\boldsymbol{d}_k$ is first divided into J blocks, i.e., $\boldsymbol{d}_k = \left[\boldsymbol{d}_k(1)^T, \ldots, \boldsymbol{d}_k(J)^T\right]^T$, where $\boldsymbol{d}_k(j) \in \{0,1\}^{m_j}$, $\sum_{j=1}^{J} m_j = M$. Then parity check bits $\boldsymbol{p}_k(j) \in \{0,1\}^{\ell_j}$ are formed based on all the data bits prior to the j-th block and appended to $\boldsymbol{d}_k(j)$ to obtain $\boldsymbol{b}_k(j) = [\boldsymbol{d}_k(j)^T, \boldsymbol{p}_k(j)^T]^T \in \{0,1\}^{M_b}, j = 2, \ldots, J$, where $M_b = m_j + \ell_j, j = 1, \ldots, J$. Finally the M_b coded bits for each block are mapped to an M_s-dimensional symbol vector with a sensing matrix $\boldsymbol{C} \in \mathbb{C}^{M_s \times N}$, where $N = 2^{M_b}$. The CS decoder outputs K decoded binary vectors for each block. To resolve the permutation ambiguity, multi-user joint decoding is employed, where a tree with J stages is built to search for K valid paths that satisfy the parity check constrains. The number of parity check computations can be up to K^J during the search process. On the other hand, if general channel codes such as turbo codes or LDPC codes are employed, then effectively all possible permutations have to be enumerated, and for each possibility, we need to perform channel decoding for all transmitters, resulting in a prohibitively high complexity of $\mathcal{O}((K!)^J)$.

IV. Massive MIMO UMA Decoding Based on Codeword, Channel and User Sparsities

A. Exploiting User Sparsity by Matrix Completion

In this section, we consider the massive MIMO scenario discussed in Sec. II. Recall the received signal given by (9)-(12). In the massive MIMO scenario, we assume that $K \ll N_a$, and $K \ll T$. Hence the signal component of the signal $\boldsymbol{Y}$ in (12) is low-rank, i.e., $\mathrm{rank}(\boldsymbol{HX}) \leq K$. In hybrid massive MIMO systems, where the number of RF chains is less than the number of antennas, i.e., $N_r < N_a$, such a low-rank structure can be exploited to recover the missing entries of the observed signal $\boldsymbol{Y}_o$ in (13) ($\tilde{\boldsymbol{Y}}_o$ in (14)) [12]. That is, given $\boldsymbol{Y}_o$, $\boldsymbol{E}$ and σ^2, we would like to find $\boldsymbol{U} \in \mathbb{C}^{N_a \times K}$ and $\boldsymbol{V} \in \mathbb{C}^{K \times T}$, such that

$$\boldsymbol{Y} = \boldsymbol{UV} + \boldsymbol{Z}, \tag{31}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are of the same dimensions as $\boldsymbol{H}$ and $\boldsymbol{X}$, respectively, such that they provide a rank-K factorization of the signal component of $\boldsymbol{Y}$, i.e., $\boldsymbol{UV} = \boldsymbol{HX}$, and the elements of $\boldsymbol{Z}$ are i.i.d. $\mathcal{CN}(0, \sigma^2)$. The alternating minimization approach to low-rank matrix completion can be formulated as follows. For fixed $\boldsymbol{U}$ and for each $t = 1, \ldots, T$, we solve the following regularized least-squares problem to obtain the estimate of $\boldsymbol{V}(:, t)$:

$$\begin{aligned}\hat{\boldsymbol{V}}(:,t) &= \underset{\boldsymbol{V}(:,t)}{\mathrm{argmin}} \|\boldsymbol{Y}_o(:,t) - \boldsymbol{E}(:,t) \circ (\boldsymbol{UV}(:,t))\|_{\mathcal{F}}^2 + \lambda \|\boldsymbol{V}(:,t)\|_{\mathcal{F}}^2 \\ &= \left(\boldsymbol{U}^H (\mathrm{diag}[\boldsymbol{E}(:,t)])\boldsymbol{U} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{U}^H (\mathrm{diag}[\boldsymbol{E}(:,t)])\boldsymbol{Y}_o(:,t),\end{aligned} \tag{32}$$

where $\mathrm{diag}[\boldsymbol{e}]$ denotes a diagonal matrix with the elements of vector $\boldsymbol{e}$ on the main diagonal, and λ is a small constant introduced to avoid singularity in solving the least-squares problems. Then for fixed $\boldsymbol{V}$ and for each $n = 1, \ldots, N_a$, we solve the following regularized least-squares problem to obtain the estimate of $\boldsymbol{U}(n, :)$:

$$\begin{aligned}\hat{\boldsymbol{U}}(n,:) &= \underset{\boldsymbol{U}(n,:)}{\mathrm{argmin}} \|\boldsymbol{Y}_o(n,:) - \boldsymbol{E}(n,:) \circ (\boldsymbol{U}(n,:)\boldsymbol{V})\|_{\mathcal{F}}^2 + \lambda \|\boldsymbol{U}(n,:)\|_{\mathcal{F}}^2 \\ &= \boldsymbol{Y}_o(n,:)(\mathrm{diag}[\boldsymbol{E}(n,:)])\boldsymbol{V}^H \left(\boldsymbol{V}(\mathrm{diag}[\boldsymbol{E}(n,:)])\boldsymbol{V}^H + \lambda \boldsymbol{I}\right)^{-1}.\end{aligned} \tag{33}$$

Starting from randomly initialized $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$, we then compute (32) and (33) alternatively until convergence. The low-rank matrix completion algorithm is summarized in Alg. 2(a), whose computational complexity is dominated by the matrix inversions and products in (32) and (33) and on the order of $\mathcal{O}\left(K^3T + KTN_a^2 + KT^2N_a\right)$.

Note that such a low-rank factorization is not unique, e.g., for any invertible matrix $\boldsymbol{P} \in \mathbb{C}^{K \times K}$,

$$\boldsymbol{U} = \boldsymbol{HP}, \tag{34}$$

$$\text{and } \boldsymbol{V} = \boldsymbol{P}^{-1}\boldsymbol{X} \tag{35}$$

is a valid decomposition.

Algorithm 2(a): Alternating Least-Squares for Matrix Completion

Input: Partially observed received signal $\boldsymbol{Y}_o$, sampling pattern $\boldsymbol{E}$, parameter λ
Output: Low-rank factors $\hat{\boldsymbol{U}} \in \mathbb{C}^{N_a \times K}$, $\hat{\boldsymbol{V}} \in \mathbb{C}^{K \times T}$
Initialize $\boldsymbol{U}$ and $\boldsymbol{V}$ with entries sampled from distribution $\mathcal{CN}(0, 1)$;
repeat
  Compute $\boldsymbol{V}(:, t), t = 1, \ldots, T$ according to (32) ;
  Compute $\boldsymbol{U}(n, :), n = 1, \ldots, N_a$ according to (33);
until convergence;
$\hat{\boldsymbol{U}} \leftarrow \boldsymbol{U}$, $\hat{\boldsymbol{V}} \leftarrow \boldsymbol{V}$.

Given $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$, in order to estimate the transmitted signals $\boldsymbol{X}$, we need to estimate the ambiguity matrix $\boldsymbol{P}$ in (34)-(35). In [12], $\boldsymbol{P}$ is estimated by making use of the pilot symbols in $\boldsymbol{X}$. In this paper, no pilot symbol is transmitted and we estimate $\boldsymbol{P}$ based on the demodulated signal of the first block.

In particular, using $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$, we obtain an estimate of the complete received signal $\boldsymbol{Y}(1)$ during the first block as

$$\hat{\boldsymbol{Y}}(1) = \hat{\boldsymbol{U}}\hat{\boldsymbol{V}}(:, 1 : M_s). \tag{36}$$

Recall that in massive MIMO UMA, the first block is always nonlinearly modulated, i.e., $\boldsymbol{x}_k(1) = \boldsymbol{C}\boldsymbol{\gamma}_k(1), k = 1, \ldots, K$. Then by applying Alg. 1(a) to $\hat{\boldsymbol{Y}}(1)$, we will obtain an estimate of the transmitted signal $\hat{\boldsymbol{X}}(1) = [\boldsymbol{C}(:, \hat{n}_1(1)), \ldots, \boldsymbol{C}(:, \hat{n}_K(1))]^T$. From (35), assuming that $K < M_s$, the ambiguity matrix can then be estimated as

$$\hat{\boldsymbol{P}}^{-1} = \hat{\boldsymbol{V}}(1)\hat{\boldsymbol{X}}(1)^{\dagger} = \hat{\boldsymbol{V}}(1)\hat{\boldsymbol{X}}(1)^H(\hat{\boldsymbol{X}}(1)\hat{\boldsymbol{X}}(1)^H)^{-1}, \tag{37}$$

where $\hat{\boldsymbol{V}}(1) = \hat{\boldsymbol{V}}(:, 1 : M_s) \in \mathbb{C}^{K \times M_s}$ is the first block of $\hat{\boldsymbol{V}}$. Then, using (34), we obtain the estimate of the channel as

$$\hat{\boldsymbol{H}} = \hat{\boldsymbol{U}}\boldsymbol{P}^{-1}. \tag{38}$$

Note that even though the estimate of $\boldsymbol{P}^{-1}$ in (37) is based on the signals in the first block, the channel estimate in (38) is based on the received signals over the entire coherence time interval, i.e., J blocks, since $\hat{\boldsymbol{U}}$ is obtained from $\boldsymbol{Y}_o \in \mathbb{C}^{N_a \times T}$.

B. Refined Channel Estimate by Exploiting Channel Sparsity

Recall that the massive MIMO UMA system is assumed to operate in the mmWave band and therefore the channels exhibit the sparse structure described in Sec. II-B, which however, is not exploited by the channel estimator given by (38). We next enhance the channel estimation accuracy by exploiting such channel sparsity structure, which will in turn lead to more accurate estimate of the ambiguity matrix $\boldsymbol{P}$, and consequently more accurate estimate of the transmitted signal $\boldsymbol{X}$.

In the mmWave channel model (10) each $\tilde{\boldsymbol{\alpha}}_k \in \mathbb{C}^{\tilde{N}}$ has L non-zero entries, with $L \ll \tilde{N}$. Hence estimating the channel $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K]$ is equivalent to estimating the sparse vectors $\tilde{\boldsymbol{\alpha}}_k, k = 1, \ldots, K$. To that end, we apply the OMP to the estimate given by (38) to refine the channel estimates by assuming that $\hat{\boldsymbol{h}}_k \approx \tilde{\boldsymbol{A}}_R\tilde{\boldsymbol{\alpha}}_k$ for each $k = 1, \ldots, K$. The massive MIMO UMA channel estimation algorithm is summarized in Alg. 2(b). The computational complexity of this algorithm is dominated by the pseudo-inverse calculation and on the order of $\mathcal{O}\left(KL^4\right)$.

Given the refined channel estimate $\tilde{\boldsymbol{H}}$, we then obtain the refined estimates of the ambiguity matrix, and the transmitted signals respectively as

$$\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{H}}^{\dagger}\hat{\boldsymbol{U}} = (\tilde{\boldsymbol{H}}^H\tilde{\boldsymbol{H}})^{-1}\tilde{\boldsymbol{H}}^H\hat{\boldsymbol{U}}, \tag{39}$$

$$\tilde{\boldsymbol{X}}(j) = [\tilde{\boldsymbol{x}}_1(j), \ldots, \tilde{\boldsymbol{x}}_K(j)]^T = \tilde{\boldsymbol{P}}\hat{\boldsymbol{V}}(j), \quad j = 1, \ldots, J, \tag{40}$$

where $\hat{\boldsymbol{V}}(j) = \hat{\boldsymbol{V}}(:, (j-1)M_s + 1 : jM_s)$ is the j-th block of $\hat{\boldsymbol{V}}$.

Algorithm 2(b): Massive MIMO UMA Channel Estimator

Input: Initial channel estimate $\hat{\boldsymbol{H}} = [\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_K] \in \mathbb{C}^{N_a \times K}$ given by (38), the channel dictionary $\tilde{\boldsymbol{A}}_R \in \mathbb{C}^{N_a \times \tilde{N}}$

Output: Refined channel estimate $\tilde{\boldsymbol{H}}$

for $k = 1, \ldots, K$ **do**

  Initialize $\boldsymbol{r} = \hat{\boldsymbol{h}}_k$, $\mathcal{J} = \emptyset$;

  while $|\mathcal{J}| < L$ **do**

    $\tilde{n}^* = \text{argmax}_{\tilde{n} \in \{1, \ldots, \tilde{N}\}} \frac{\boldsymbol{r}^H \tilde{\boldsymbol{A}}_R(:, \tilde{n})}{\|\tilde{\boldsymbol{A}}_R(:, \tilde{n})\|_2}$;

    $\mathcal{J} \leftarrow \mathcal{J} \cup \{\tilde{n}^*\}$;

    $\boldsymbol{\delta} = \tilde{\boldsymbol{A}}_R(:, \mathcal{J})^{\dagger}\hat{\boldsymbol{h}}_k$;

    $\boldsymbol{r} = \hat{\boldsymbol{h}}_k - \tilde{\boldsymbol{A}}_R(:, \mathcal{J})\boldsymbol{\delta}$;

  end

  $\tilde{\boldsymbol{h}}_k = \tilde{\boldsymbol{A}}_R(:, \mathcal{J})\boldsymbol{\delta}$;

end

$\tilde{\boldsymbol{H}} = \left[\tilde{\boldsymbol{h}}_1, \ldots, \tilde{\boldsymbol{h}}_K\right]$.

C. Single-User Demodulation and Decoding

Given the estimated transmitted signals $\tilde{\boldsymbol{x}}_k(j)$ in (40) of transmitter k for $j = 1, \ldots, J$, we can then perform single-user demodulation and decoding to obtain the estimate of its information bits $\boldsymbol{d}_k$. When nonlinear modulation is employed, i.e., $\boldsymbol{x}_k(j) = \boldsymbol{C}\boldsymbol{\gamma}_k(j), j = 1, \ldots, J$, the hard demodulation is given by

$$\hat{n}_k(j) = \arg\min_{n=1,\ldots,N} \|\tilde{\boldsymbol{x}}_k(j) - \boldsymbol{C}(:, n)\|^2, \tag{41}$$

$$\hat{\boldsymbol{b}}_k(j) = \text{binary expansion of } \hat{n}_k(j), \ j = 1, \ldots, J. \tag{42}$$

Similar to (27)-(29), the soft demodulation is performed according to the following

$$\begin{aligned}\hat{\boldsymbol{Y}}_k(j) &= \hat{\boldsymbol{Y}}(j) - \sum_{k' \neq k} \tilde{\boldsymbol{h}}_{k'}\tilde{\boldsymbol{x}}_{k'}(j)^T \\ &\approx \tilde{\boldsymbol{h}}_k\boldsymbol{x}_k(j)^T + \boldsymbol{Z}(j) \in \mathbb{C}^{N_a \times M_s}, \ j = 1, \ldots, J, \end{aligned} \tag{43}$$

where $\hat{\boldsymbol{Y}}(j) = \hat{\boldsymbol{U}}\hat{\boldsymbol{V}}(:, (j-1)M_s + 1 : jM_s)$,

$$\begin{aligned} P\big(\boldsymbol{x}_k(j) &= \boldsymbol{C}(:, n) \mid \hat{\boldsymbol{Y}}_k(j)\big) \\ &\propto \exp\left(\underbrace{-\frac{\|\hat{\boldsymbol{Y}}_k(j) - \tilde{\boldsymbol{h}}_k\boldsymbol{C}(:, n)^T\|_{\mathcal{F}}^2}{\sigma^2}}_{\beta_{k,j,n}}\right), \\ & \qquad n = 1, \ldots, N = 2^{M_b}, \end{aligned} \tag{44}$$

$$\begin{aligned} \text{and} \quad \text{LLR}_k(j, i) &\triangleq \log \frac{P\big([\boldsymbol{b}_k(j)]_i = 0 \mid \hat{\boldsymbol{Y}}_k(j)\big)}{P\big([\boldsymbol{b}_k(j)]_i = 1 \mid \hat{\boldsymbol{Y}}_k(j)\big)} \\ &\approx \max_{n:\text{bin}(n,i)=0} \beta_{k,j,n} - \max_{n:\text{bin}(n,i)=1} \beta_{k,j,n}, \\ & \qquad i = 1, \ldots, M_b, \ j = 1, \ldots, J. \end{aligned} \tag{45}$$

On the other hand, when hybrid modulation is employed, the code bits of the first block, i.e., $\boldsymbol{b}_k(1)$, is nonlinearly modulated. Then $\hat{n}_k(1)$ and the hard demodulation $\hat{\boldsymbol{b}}_k(1)$ is given by (41)-(42) for $j = 1$. For soft demodulation, the LLRs of the code bits of $\boldsymbol{b}_k(1)$, i.e., $\text{LLR}_k(1, i), i = 1, \ldots, M_b$, are given by (43)-(45) for $j = 1$.

For the rest of the blocks, $j = 2, \ldots, J$, the code bits $\boldsymbol{b}_k(j)$ are linearly modulated with the spreading sequence $\boldsymbol{s}_k = \boldsymbol{x}_k(1) = \boldsymbol{C}(:, n_k(1))$ as in (2)-(4), where recall that $\tilde{\boldsymbol{b}}_k(j) = 2\boldsymbol{b}_k(j) - 1$. We denote the segments of $\tilde{\boldsymbol{x}}_k(j)$, $\hat{\boldsymbol{Y}}_k(j)$ and $\boldsymbol{Z}(j)$ in (40) and (43) corresponding to the i-th bit of $\boldsymbol{b}_k(j)$ (denoted by $[\boldsymbol{b}_k(j)]_i$) respectively as

$$[\tilde{\boldsymbol{x}}_k(j)]_i = \tilde{\boldsymbol{x}}_k(j)\left(\frac{(i-1)M_s}{M_b} + 1 : \frac{iM_s}{M_b}\right) \in \mathbb{C}^{\frac{M_s}{M_b}}, \quad (46)$$

$$[\hat{\boldsymbol{Y}}_k(j)]_i = \hat{\boldsymbol{Y}}_k(j)\left(:, \frac{(i-1)M_s}{M_b} + 1 : \frac{iM_s}{M_b}\right) \in \mathbb{C}^{N_a \times \frac{M_s}{M_b}}, \quad (47)$$

$$[\boldsymbol{Z}(j)]_i = \boldsymbol{Z}(j)\left(:, \frac{(i-1)M_s}{M_b} + 1 : \frac{iM_s}{M_b}\right) \in \mathbb{C}^{N_a \times \frac{M_s}{M_b}}, \quad i = 1, \ldots, M_b \quad (48)$$

Moreover, the estimated spreading sequence for $[\tilde{\boldsymbol{b}}_k(j)]_i$ is

$$\hat{\boldsymbol{s}}_{k,i} = \boldsymbol{C}\left(\frac{(i-1)M_s}{M_b} + 1 : \frac{iM_s}{M_b}, \hat{n}_k(1)\right) \in \mathbb{C}^{\frac{M_s}{M_b}}, \quad i = 1, \ldots, M_b. \quad (49)$$

Then the hard demodulation of $[\hat{\boldsymbol{b}}_k(j)]_i$ is given by

$$[\hat{\boldsymbol{b}}_k(j)]_i = \frac{1}{2}\left[\text{sign}(\Re\{\hat{\boldsymbol{s}}_{k,i}^H [\tilde{\boldsymbol{x}}_k(j)]_i\}) + 1\right], \quad i = 1, \ldots, M_b,\ j = 2, \ldots, J, \quad (50)$$

where $\Re(\cdot)$ is the real operator. The above linear hard demodulator has a complexity of $\mathcal{O}(M_b)$ *per block per transmitter*, which is the same as that of the nonlinear hard demodulator.

For soft decision, (43) can be rewritten as

$$[\hat{\boldsymbol{Y}}_k(j)]_i \approx [\tilde{\boldsymbol{b}}_k(j)]_i \tilde{\boldsymbol{h}}_k \hat{\boldsymbol{s}}_{k,i}^T + [\boldsymbol{Z}(j)]_i,\ i = 1, \ldots, M_b. \quad (51)$$

Hence the LLR of $[\boldsymbol{b}_k(j)]_i$ can be computed as

$$\text{LLR}_k(j,i) = \log \frac{P\left([\tilde{\boldsymbol{b}}_k(j)]_i = -1 \mid [\hat{\boldsymbol{Y}}_k(j)]_i\right)}{P\left([\tilde{\boldsymbol{b}}_k(j)]_i = 1 \mid [\hat{\boldsymbol{Y}}_k(j)]_i\right)} \quad (52)$$

$$= -\frac{4}{\sigma^2}\Re\left\{\text{tr}\left([\hat{\boldsymbol{Y}}_k(j)]_i^H \tilde{\boldsymbol{h}}_k \hat{\boldsymbol{s}}_{k,i}^T\right)\right\}, \quad i = 1, \ldots, M_b,\ j = 2, \ldots, J, \quad (53)$$

where $\text{tr}(\cdot)$ is the trace operator. Hence the soft linear demodulator also has a complexity of $\mathcal{O}(M_b)$ *per block per transmitter*, which is much simpler than the $\mathcal{O}\left(2^{M_b}\right)$ complexity of the soft nonlinear demodulator.

The massive MIMO UMA decoding algorithm is summarized in Alg. 2. Note that unlike Alg. 1 which runs the S-OMP algorithm (Alg. 1(a)) on each block for a total of J times followed by the clustering algorithm (Alg. 1(b)), Alg. 2 runs the S-OMP algorithm only once on the first block and no clustering algorithm is needed. This is because in additional to the codeword sparsity, it further exploits channel sparsity and the signal correlation among different blocks due to the low-rank structure caused by the user sparsity.

The computational complexity of Alg. 2 consists of four parts: the complexity of matrix completion using Alg. 2(a), which is $\mathcal{O}\left(K^3T + KTN_a^2 + KT^2N_a\right)$; the complexity of decoding $\{\hat{n}_k(1)\}$ for the first block using Alg. 1(a), which is $\mathcal{O}\left(2^{M_b}M_s^3\right)$; the complexity of refining channel estimates using Alg. 2(b), which is $\mathcal{O}\left(KL^4\right)$; and the complexity of single-user demodulation for nonlinear and hybrid demodulator for all K transmitters, which is $\mathcal{O}(JKM_b)$ for hard nonlinear demodulation and hard and soft linear demodulation, and $\mathcal{O}\left(JK2^{M_b}\right)$ for soft nonlinear demodulation. Note that the matrix completion step of Alg. 2 not only estimates the missing received signals in a hybrid massive MIMO systems, more importantly, it also significantly simplifies the subsequent decoding processes. In particular, it completely eliminates the CS decoding processes for $J-1$ blocks and the channel clustering process in Alg. 1. Moreover, the hybrid modulation further leads to much simpler single-user soft demodulation processes compared with the nonlinear modulation.

Algorithm 2: Massive MIMO UMA Decoding Based on Codeword Sparsity, Channel Sparsity and User Sparsity

Input: Partially observed received signal $\boldsymbol{Y}_o$, sampling pattern $\boldsymbol{E}$, channel noise variance σ^2, sensing matrix $\boldsymbol{C}$, channel dictionary $\tilde{\boldsymbol{A}}_R$

Output: Decoded information bits of all active transmitters $\hat{\boldsymbol{d}}_k$ or $\text{LLR}(\boldsymbol{d}_k)$, $k = 1, \ldots, K$

Run Alg. 2(a) to obtain $\hat{\boldsymbol{U}}, \hat{\boldsymbol{V}}$; Obtain $\hat{\boldsymbol{Y}}(1)$ in (36);

Run Alg. 1(a) for $j = 1$ to obtain $\hat{n}_k(1), k = 1, \ldots, K$;

Compute $\hat{\boldsymbol{P}}^{-1}$ in (37) and $\hat{\boldsymbol{H}}$ in (38);

Run Alg. 2(b) to obtain $\tilde{\boldsymbol{H}} = \left[\tilde{\boldsymbol{h}}_1, \ldots, \tilde{\boldsymbol{h}}_K\right]$;

Compute $\tilde{\boldsymbol{X}}(j)$ using (39)-(40) for $j = 2, \ldots, J$;

For nonlinear modulation:

for $k = 1, \ldots, K$ **do**

  Hard decoder:

  Perform hard demodulation according to (41)-(42) to obtain $\hat{\boldsymbol{b}}_k$ and then perform hard decoding to get $\hat{\boldsymbol{d}}_k$;

  Soft decoder:

  Perform soft demodulation according to (43)-(45), and then perform soft decoding to get $\text{LLR}(\boldsymbol{d}_k)$.

end

For hybrid modulation:

for $k = 1, \ldots, K$ **do**

  Hard decoder:

  Perform hard demodulation using (41)-(42) for $j = 1$, and (46), (49)-(50) for $j = 2, \ldots, J$ to obtain $\hat{\boldsymbol{b}}_k$ and then perform hard decoding to get $\hat{\boldsymbol{d}}_k$;

  Soft decoder:

  Perform soft demodulation for the first block using (43)-(45) with $j = 1$ and for the rest of blocks using (47)-(49) and (52) to obtain $\text{LLR}(\boldsymbol{b}_k)$, then perform soft decoding to get $\text{LLR}(\boldsymbol{d}_k)$.

end

Remark 1: Note that we can also apply Alg. 1 to the completed signal $\hat{\boldsymbol{Y}}$ from Alg. 2(a) if the signals are nonlinearly modulated. However, such an approach has two drawbacks. One is that for nonlinear modulated sub-blocks, such an approach incurs a much higher complexity than the method in Alg. 2 that exploits the low-rank decomposition. In particular, the steps of running Alg. 1(a) for the first sub-block ($\mathcal{O}\left(2^{M_b}M_s^3\right)$) and computing (37)-(40) ($\mathcal{O}\left(K^3\right)$) in Alg. 2 are replaced by running Alg. 1(a) ($\mathcal{O}\left(2^{M_b}M_s^3J\right)$) and

Alg. 1(b) ($\mathcal{O}\left(K^2 N_a J\right)$) for all sub-blocks, and computing (20) ($\mathcal{O}\left(JK\right)$). And the second drawback is that Alg. 1 cannot be used to demodulate linearly modulated sub-blocks.

D. Extension to Systems With Quantized Received Signals

A hybrid massive MIMO system may still require a large number of RF chains, with high power consumption and hardware complexity. In particular, within an RF chain, the power consumption of the high-resolution analog-to-digital converters (ADCs) increases exponentially with the number of bits per sample and linearly with the sampling rate [15]. To reduce the power consumption and hardware complexity, low-resolution ADCs can be employed by quantizing the received signal using a few bits. We next discuss the extension of Alg. 2 to the case when the partially observed received signals $\boldsymbol{Y}_o \in \mathbb{C}^{N_a \times T}$ are quantized.

We define a quantization function $\mathcal{Q}_{b,\Delta}(\cdot)$ that applies to a complex scalar y, where parameter b is the resolution of ADCs in terms of number of bits and Δ is the quantization step-size. As in [12], the value of Δ is picked based on the optimal values of stepsize Δ_b that achieves the minimal NMSE when the input signal is distributed as i.i.d. $\mathcal{N}(0,1)$ (Table I in [12]). We then obtain the stepsize $\Delta = \sqrt{P_y/2} \cdot \Delta_b$, where $P_y = \mathbb{E}\left[|y|^2\right]$ is the average power of y. Then according to Bussagang's theorem [12], when $\mathcal{Q}_{b,\Delta}(\cdot)$ is applied element-wisely to the received signal $\boldsymbol{Y} \in \mathbb{C}^{N_a \times T}$, the resulting quantized signal $\boldsymbol{Q}$ after normalization can be written as

$$\tilde{\boldsymbol{Q}} \triangleq \frac{1}{g}\mathcal{Q}_{b,\Delta}(\boldsymbol{Y}) = \boldsymbol{H}\boldsymbol{X} + \tilde{\boldsymbol{\Psi}}, \tag{54}$$

where g is the scaling constant and $\tilde{\boldsymbol{\Psi}}$ is the scaled noise that contains i.i.d. Gaussian elements $\mathcal{CN}(0, \frac{\nu^2}{g^2} + \sigma^2)$ with the noise scaling factor ν. Hence when Alg. 2 is applied to a hybrid massive MIMO system with low-resolution ADCs, we make the following two changes (g and ν^2 are given by (56) and (57) respectively in Section V-B):

- The partially observed received signal $\boldsymbol{Y}_o$ is replaced by its quantized and scaled version $\frac{1}{g}\mathcal{Q}_{b,\Delta}(\boldsymbol{Y}_o)$.
- The channel noise variance σ^2 is replaced by $\frac{\nu^2}{g^2} + \sigma^2$.

V. Simulation Results

In this section, we present simulation results to illustrate the performances of the proposed sparsity-exploiting blind receiver algorithms for channel estimation, demodulation and decoding in both MIMO UMA and massive MIMO UMA systems. The common system parameters of the two systems are: the number of information bits per transmitter in each coherence time interval $M = 70$, the length of a coherence time interval $T = 1430$, the code rate $r = 7/11$, and the number of sub-intervals $J = 11$. Hence the length of each block is $M_s = T/J = 130$, the total number of code bits is $M_c = M/r = 110$, the number of code bits per block is $M_b = M_c/J = 10$, and the number of columns in the sensing matrix is $N = 2^{M_b} = 1024$. For channel codes, we use the CRC-Aided Polar (CA-Polar) code [16] with 11 CRC bits because Polar codes are well suited for short data packets. Both the CA-Polar encoder and decoder are implemented using MATLAB's Communication Toolbox. To mitigate the burst error across different blocks of each transmitter's demodulated coded bits, each transmitter randomly interleaves the coded bits before dividing them into blocks. The sensing matrix $\boldsymbol{C}$ is an $M_s \times N = 130 \times 1024$ normalized complex Gaussian matrix consisting of i.i.d. $\mathcal{CN}(0,1)$ elements, i.e., each column is normalized to have unit norm. The performance of channel estimation is measured by the normalized mean-squared error (NMSE) defined as $\text{NMSE}(\hat{\boldsymbol{H}}) = \mathbb{E}\{\|\boldsymbol{H} - \hat{\boldsymbol{H}}\|_{\mathcal{F}}^2\}/\mathbb{E}\{\|\boldsymbol{H}\|_{\mathcal{F}}^2\}$. The bit error rates (BERs) of demodulation and decoding are defined as $P([\hat{\boldsymbol{b}}_k]_i \neq [\hat{\boldsymbol{b}}_k]_i)$ and $P([\hat{\boldsymbol{d}}_k]_i \neq [\hat{\boldsymbol{d}}_k]_i)$, respectively. Similarly, the frame error rates (FERs) are defined as $P(\hat{\boldsymbol{b}}_k \neq \hat{\boldsymbol{b}}_k)$ and $P(\hat{\boldsymbol{d}}_k \neq \hat{\boldsymbol{d}}_k)$.

A. Performance of Alg. 1 in MIMO UMA Systems

For MIMO UMA systems, we set the number of antennas $N_a = 4$ and consider two types of channel models: for the i.i.d. channel, the channel vectors are generated according to $\boldsymbol{h}_k \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}), k = 1, \ldots, K$; and for the geometric wideband channel, the channel vectors are generated according to (8), with antenna spacing $d/\lambda = 1/2$, the number of scatters $L = 4$, and $\alpha_{k,\ell} \sim \mathcal{CN}(0,1)$. The DoA range is divided into $\tilde{N} = 96$ grids and hence the dimension of the channel dictionary matrix $\boldsymbol{A}_R$ is $N_a \times \tilde{N} = 4 \times 96$. The channel quality is measured by the signal-to-noise ratio (SNR) per channel tap per transmitter given by $\text{SNR} = \mathbb{E}\{\|\boldsymbol{h}_k\|^2\}/N_a\sigma^2$. We first compare our proposed MIMO UMA decoding algorithm (Alg. 1) with the tree decoder in [5] that is specifically designed for parity check codes. To obtain the parity bits, 70 information bits are divided into 11 blocks of sizes $m_1 = 10$ and $m_j = 6, j = 2, \ldots, 11$. For the j-th block, ℓ_j parity bits are appended to the m_j information bits such that all blocks have equal length, where $\ell_1 = 0$ and $\ell_j = 4, j = 2, \ldots, 11$. Denote the information bits and parity check bits in the j-th block as $\boldsymbol{d}_k(j) \in \mathbb{C}^{m_j}$ and $\boldsymbol{p}_k(j) \in \mathbb{C}^{l_j}, j = 1, \ldots, J$. Specifically, the ℓ_j parity check bits in the j-th block are generated by the mod-2 multiplication of all the information bits in the preceding blocks and a Rademacher matrix, i.e., $\boldsymbol{p}_k(j) = [\boldsymbol{d}_k(1)^T, \ldots, \boldsymbol{d}_k(j-1)^T]^T \boldsymbol{G}, j = 2, \ldots, J$ where the entries of $\boldsymbol{G} \in \{0,1\}^{\left(\sum_{j'=1}^{j-1} m_{j'}\right) \times l_j}$ are uniform Bernoulli trials. In Alg. 1, hard demodulation and hard decoding are employed for parity-check codes.

Fig. 3 and Fig. 4 show the decoding FER and BER performances of parity check codes using tree decoder and Alg. 1. We can see that Alg. 1 significantly outperforms the tree decoder for different values of K and SNR conditions. In particular, the tree decoder has reasonable decoding performance only when K is very small, e.g., $K < 10$, while Alg. 1 is able to decode successfully for a much larger range of K. Moreover, as SNR increases, the FER and BER curves of Alg. 1 exhibit a steeper slope than those of the tree decoder. Note that both methods perform the CS decoding on each block. Their performance differences result from how they assemble the J partially decoded signal segments for each transmitter. For random parity check codes, the tree decoder

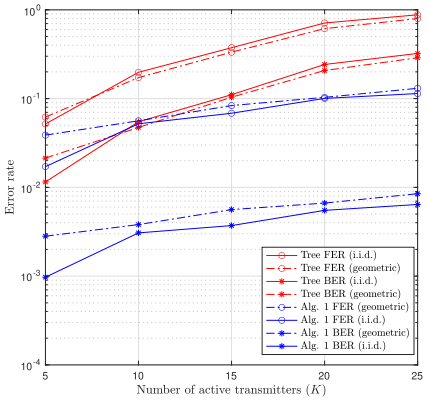


Fig. 3. Decoding error rates for parity check codes in MIMO UMA systems. $N_a = 4$, SNR $= 0$ dB.

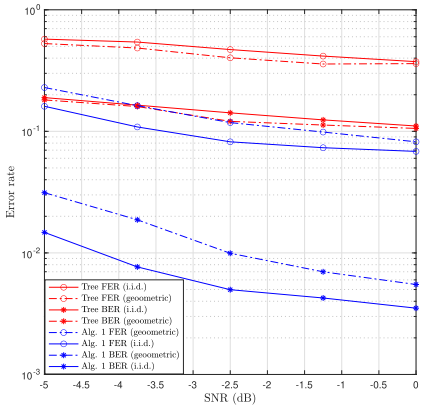


Fig. 4. Decoding error rates for parity check codes in MIMO UMA systems. $N_a = 4, K = 15$.

usually finds more than one path that satisfy the parity checks for each root of the tree, so decoding error occurs quite often. On the other hand, Alg. 1 exploits the estimated channel state at each block, which is ignored by the tree decoder. The channel state is transmitter specific and essentially serves as a tag for the signal from that transmitter. Hence the proposed channel clustering method (i.e., Alg. 1(b)) is a simpler yet more accurate way of resolving the permutation ambiguity than the tree search. We can also observe that the tree decoder has similar performances in i.i.d. and geometric channels while Alg. 1 performs better in i.i.d. channels.

Fig. 5 and Fig. 6 show the hard/soft demodulation and decoding FER performances using the proposed Alg. 1 when the CA-Polar code is employed. We can observe that soft demodulation and decoding exhibit a significant performance gain over hard demodulation and decoding for different value of K and SNR conditions; and such a gain increases as K decreases or SNR increases. Moreover, both hard and soft decoders achieve a higher coding gain under better channel conditions, i.e., higher SNR or smaller K. For both hard and soft demodulation and decoding, the performance in i.i.d. channels is better than that in geometric channels. Furthermore, comparing Fig. 3 and Fig. 5, we see that by employing the CA-Polar code rather than the parity check code, a much larger number of active transmitters can be accommodated.

Fig. 7 shows the channel estimation performance of Alg. 1(b) under different number of active transmitters and SNR. We observe that the performance of channel estimation improves as SNR increases and the number of active transmitters decreases.

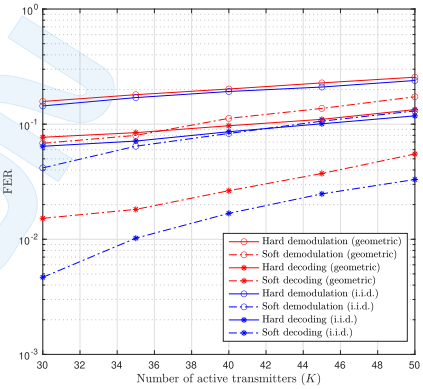


Fig. 5. Demodulation and decoding error rates for CA-Polar codes in MIMO UMA systems. $N_a = 4$, SNR $= 0$ dB.

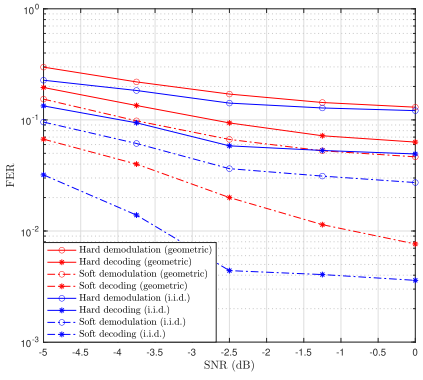


Fig. 6. Demodulation and decoding error rates for CA-Polar codes in MIMO UMA systems. $N_a = 4, K = 25$.

B. Performance of Alg. 2 in Massive MIMO UMA Systems

For the massive MIMO UMA, we adopt the mmWave channel model in (8) with $N_a = 64$ receive antennas, $N_r = 32$ RF chains, and $\alpha_{k,\ell} \sim \mathcal{CN}(0, 16)$. Other parameters are the same as that of the MIMO UMA systems in Section V-A hence the dimension of the channel dictionary matrix $\boldsymbol{A}_R$ is $N_a \times \tilde{N} = 64 \times 96$. For low-resolution ADC scenarios, we consider 2-bit resolution, i.e., $b = 2$. The quantization function is defined as

$$\begin{aligned}&\mathcal{Q}_{b,\Delta}(y)\\ &= \operatorname{sign}(\Re\{y\})\left(\min\left\{\left\lceil\frac{|\Re\{y\}|}{\Delta}\right\rceil, 2^{b-1}\right\} - \frac{1}{2}\right)\Delta\\ &\quad + i\,\operatorname{sign}(\Im\{y\})\left(\min\left\{\left\lceil\frac{|\Im\{y\}|}{\Delta}\right\rceil, 2^{b-1}\right\} - \frac{1}{2}\right)\Delta. \quad (55)\end{aligned}$$

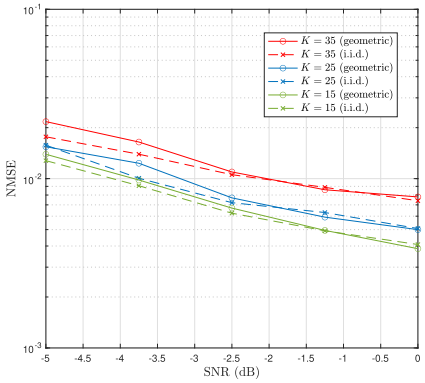


Fig. 7. Channel estimation performance in MIMO UMA systems. $N_a = 4$.

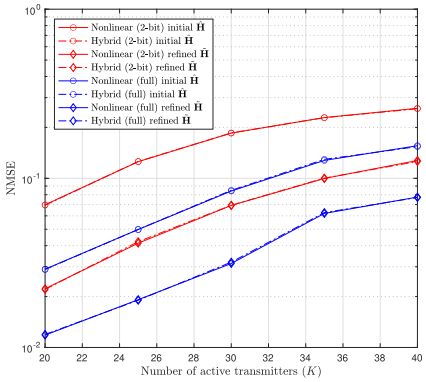


Fig. 8. Channel estimation performance in massive MIMO UMA systems. $N_a = 64, N_r = 32, \text{SNR} = -5$ dB.

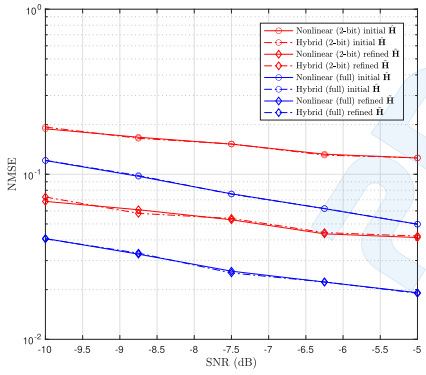


Fig. 9. Channel estimation performance in massive MIMO UMA systems. $N_a = 64, N_r = 32, K = 25$.

The parameters in model (54) are given by

$$g = \frac{\Delta}{\sqrt{\pi P_y}} + \sum_{\ell=1}^{2^{b-1}-1} \frac{2\Delta}{\sqrt{\pi P_y}} \exp\left(-\frac{(\Delta\ell)^2}{P_y}\right), \tag{56}$$

and

$$\nu^2 = 4[(2^{b-1} - 0.5)\Delta]^2 \left(1 - \Phi\left(\sqrt{\frac{[(2^{b-1}-1)\Delta]^2}{P_y/2}}\right)\right) + 4 \sum_{\ell=1}^{2^{b-1}-1} [(\ell - 0.5)\Delta]^2 \Phi\left(\sqrt{\frac{(\ell\Delta)^2}{P_y/2}}\right)$$

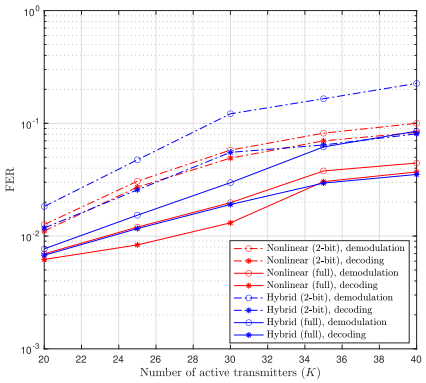


Fig. 10. Demodulation and decoding FER in massive MIMO UMA systems. $N_a = 64, N_r = 32, \text{SNR} = -5$ dB.

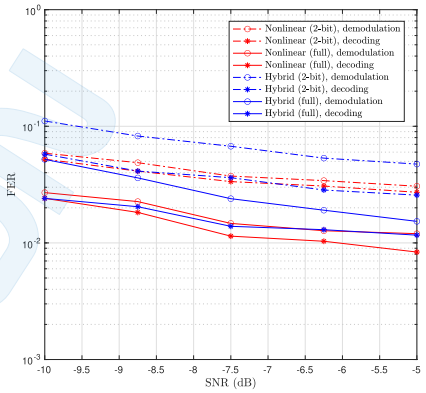


Fig. 11. Demodulation and decoding FER in massive MIMO UMA systems. $N_a = 64, N_r = 32, K = 25$.

$$- 4 \sum_{\ell=1}^{2^{b-1}-1} [(\ell - 0.5)\Delta]^2 \Phi\left(\sqrt{\frac{[(\ell-1)\Delta]^2}{P_y/2}}\right) - g^2 P_y, \tag{57}$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$ is the cumulative distribution function of the standard normal distribution.

Since the performances of analog combining based on switches and phase shifters are almost the same, here we present the results for switch-based combining method only. For the t-th column of the connection matrix $\boldsymbol{E}$ in (13), we randomly generate N_r distinct indices $\mathcal{I}_t \subset \{1, \ldots, N_a\}$ and set the elements with row index in $\mathcal{I}_t$ as 1, and set others as 0, $t = 1, \ldots, T$. Both the nonlinear modulation and the hybrid modulation are considered, and soft demodulation and decoding are employed. Note that they have the same channel rate of $M_b/M_s = 1/13$ code bits per channel use per transmitter. Since the receiver uses only N_r out of N_a antennas, the SNR per channel tap per transmitter in the massive MIMO system is defined as $\text{SNR} = \frac{\mathbb{E}\{\|\boldsymbol{h}_k\|^2\} N_r}{N_a^2 \sigma^2}$. In Alg. 2(a), we set $\lambda = \sigma^2$.

Fig. 8 and Fig. 9 show the channel estimation performance under different modulations and resolutions. We compare the initial estimate $\hat{\boldsymbol{H}}$ in (38) and the refined estimate $\tilde{\boldsymbol{H}}$ given by Alg. 2(b). It is seen that the performance of channel estimation improves as the resolution and SNR increase and K decreases.

The two modulation schemes have almost the same channel estimation performance. By exploiting the channel sparsity, Alg. 2(b) significantly improves the accuracy of the initial channel estimate given by (38).

Fig. 10 and Fig. 11 show the demodulation and decoding error rates for different modulations and resolutions. In general, both demodulation and decoding performances improve as the resolution and SNR increase and K decreases for all methods. Comparing the nonlinear modulation and the hybrid modulation, we observe that after demodulation, the nonlinear modulation tends to have less erroneous frames, but more erroneous bits within an erroneous frame; whereas the hybrid modulation tends to have more erroneous frames, but less erroneous bits in each frame. This is because the nonlinear demodulator finds the best codeword corresponding to a code bit sequence block; whereas the linear demodulator finds the best estimate of each code bit. Hence we can see from Fig. 10 and Fig. 11 that the demodulation FER performance of the hybrid demodulation is worse than that of the nonlinear demodulation while the decoding FER performances of the two modulation schemes are similar.

VI. Conclusion

We have proposed new transmission schemes and blind receiver algorithms for unsourced multiple access over both MIMO and massive MIMO channels. Each transmitter's information bits are encoded by a channel code and the coded bits are divided into sub-blocks, and each sub-block is modulated and then transmitted. For the MIMO channel, the conventional nonlinear modulation is employed where each sub-block of coded bits is mapped to a transmitted signal vector. We have proposed a receiver algorithm that exploits the codeword sparsity using the S-OMP algorithm to estimate the channels and the transmitted signals in each sub-block; it then properly assembles the estimates of each transmitted codeword using a channel clustering algorithm. For the massive MIMO channel, in addition to the nonlinear modulation, we have also proposed a hybrid modulation that adopts linear modulation and spreading for all sub-blocks except for the first one, to reduce the receiver complexity. The proposed massive MIMO receiver algorithm exploits codeword sparsity, channel sparsity and user sparsity, and it can handle both missing and quantized received signals, caused by lower number of RF chains than the number of antennas and low-resolution ADCs, respectively. Both receiver algorithms can output soft information of the coded bits facilitating single-user decoding of advanced channel codes, such as Polar codes.

References

[1] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2523–2527.

[2] X. Chen, T.-Y. Chen, and D. Guo, "Capacity of Gaussian many-access channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3516–3539, Jun. 2017.

[3] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access Gaussian channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2528–2532.

[4] A. Vem, K. R. Narayanan, J.-F. Chamberland, and J. Cheng, "A user-independent successive interference cancellation based coding scheme for the unsourced random access Gaussian channel," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8258–8272, Dec. 2019.

[5] V. K. Amalladinne, A. Vem, D. K. Soma, K. R. Narayanan, and J.-F. Chamberland, "A coupled compressive sensing scheme for unsourced multiple access," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6628–6632.

[6] A. Fengler, P. Jung, and G. Caire, "SPARCs and AMP for unsourced random access," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 2843–2847.

[7] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, "Grant-free massive random access with a massive MIMO receiver," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 23–30.

[8] J. Ma, S. Zhang, H. Li, F. Gao, and S. Jin, "Sparse Bayesian learning for the time-varying massive MIMO channels: Acquisition and tracking," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1925–1938, Mar. 2019.

[9] M. Li, S. Zhang, F. Gao, P. Fan, and O. A. Dobre, "A new path division multiple access for the massive MIMO-OTFS networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 903–918, Apr. 2021.

[10] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "A performance comparison of measurement matrices in compressive sensing," *Int. J. Commun. Syst.*, vol. 31, no. 10, Jul. 2018, Art. no. e3576.

[11] H. Chu, L. Zheng, and X. Wang, "Super-resolution mmwave channel estimation for generalized spatial modulation systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 6, pp. 1336–1347, Oct. 2019.

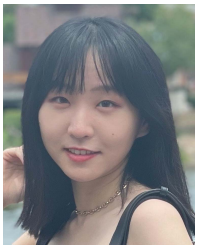
[12] S. Liang, X. Wang, and L. Ping, "Semi-blind detection in hybrid massive MIMO systems via low-rank matrix completion," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5242–5254, Nov. 2019.

[13] 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, 2006.

[14] R. Calderbank and A. Thompson, "CHIRRUP: A practical algorithm for unsourced multiple access," 2018, *arXiv:1811.00879*. [Online]. Available: http://arxiv.org/abs/1811.00879

[15] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[16] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.



Jiaai Liu received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2018, and the M.S. degree in electrical and computer engineering from Duke University, Durham, NC, USA, in 2020. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Columbia University, New York, NY, USA. Her research interests include channel coding, massage passing algorithms, and machine learning for communications.

Xiaodong Wang (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Princeton University. He is currently a Professor of electrical engineering with Columbia University, New York, NY, USA. He has published the book *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Prentice Hall, 2003). His research interests fall in the general areas of computing, signal processing, and communications. He has published extensively in these areas. His current research interests include wireless communications, statistical signal processing, and genomic signal processing. He received the 1999 NSF CAREER Award, the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award, and the 2011 IEEE Communication Society Award for outstanding paper on new communication topics. He has served as an Associate Editor for IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Transactions on Signal Processing, and IEEE Transactions on Information Theory. He is listed as an ISI Highly-Cited Author.