Hyper-Dimensional Modulation for Robust Short Packets in Massive Machine-Type Communications

Chin-Wei Hsu and Hun-Seok Kim, *Senior Member, IEEE*

***Abstract*— In this paper, we introduce Hyper-Dimensional Modulation (HDM) for massive machine-type communications (mMTC). HDM enables robust communication of short packets by spreading information bits across many elements in a hyper-dimensional vector and superimposing a set of such non-orthogonal vectors. The proposed CRC-aided K-best decoding algorithm for HDM can achieve a very low packet error rate (PER) in additive white Gaussian noise (AWGN) channels for short packets. Furthermore, extended decoding algorithms are proposed to combat overwhelming interference in an mMTC network. Comprehensive simulation and real-world experiment results show that HDM outperforms sparse superposition codes in AWGN channels and state-of-the-art short codes such as polar and tail-biting convolutional codes in interference-heavy channels for short packet transmissions.**

***Index Terms*— Massive machine-type communications (mMTC), multiple access (MA) interference.**

I. Introduction

THE ITU-R categorizes the 5G and beyond technologies into three classes: enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), and Ultra-Reliable and Low Latency Communications (URLLC), while each targets different applications [4]. mMTC use-case examples include transportation, utilities, health, environment, and security [5]. Packets for these mMTC applications usually carry relatively small amount of information such as control commands or sensor readings. However, the number of nodes in mMTC networks can be much greater than that of consumer (non-machine) mobile cellular networks. Thus it poses new challenges in the physical layer (PHY) design for reliable communication of short packets in interference-heavy channels [6].

Up to 4G, the main focus of the development had been to boost the data rate with high spectral efficiency for human-oriented communications. However, novel applications in mMTC usually convey relatively short information as small as a few bytes per packet. The conventional PHY and network design optimized for large amount of information is not necessarily efficient in those applications. First, the overhead of preamble and pilot symbols is no longer negligible compared to the small number of information bits. Therefore, the frame structure needs to be re-designed with consideration of the overhead [7]. Second, the efficiency of modern codes such as Turbo and LDPC codes greatly relies on the long block-length and when the packet size is small, reliability of these codes significantly degrades. To quantify the efficiency of short codes, Polyanskiy showed that the normal approximation [8] provides a closed-form expression that tightly follows the achievability and converse bounds for short blocklength. Motivated by this, new coding schemes have been recently investigated [9], [10] to approach the limit for short codes.

Another challenge in mMTC is the interference especially when many nodes share the same unlicensed ISM (industrial, scientific and medical) band with heterogeneous PHY and multiple access (MA) protocols. Since grant-based multiple access protocols are often inefficient when the number of nodes is large [6], grant-free non-orthogonal multiple access schemes that allow multiple nodes accessing the channel at the same time have been investigated to support more nodes in a network [11], [12], [13], [14]. However, these schemes still require slot-based time synchronization among nodes.

The overhead of control signals for network coordination often offsets the potential benefits of being synchronous. Moreover, precise synchronization in time and frequency is impractical for many narrowband low power mMTC nodes because of accuracy limitations in carrier frequency and sampling frequency generation. When an asynchronous mMTC network without strict synchronization among nodes operates in an unlicensed ISM band shared with heterogeneous networks such as WiFi, Bluetooth, Zigbee, etc., it is inevitable to observe severe intra- and inter-network interference. Therefore, it is a critical task for mMTC to design a novel PHY and multiple access scheme for short packets to mitigate severe interference from both intra- and inter-network traffic.

We propose Hyper-Dimensional Modulation (HDM) [1], [2], [3] as a potential solution to address aforementioned challenges in mMTC. HDM is a non-orthogonal modulation scheme that can provide excellent reliability with short packet lengths, and it is inherently tolerant to interference.

Manuscript received 31 December 2021; revised 11 May 2022, 16 September 2022, and 3 December 2022; accepted 13 January 2023. Date of publication 24 January 2023; date of current version 17 March 2023. This work was funded in part by DARPA YFA #D18AP00076 and NSF CAREER #1942806. This work is expanded from the authors' previous conference publications [DOI: 10.1109/ICC.2018.8422472], [DOI: 10.1109/GLOBECOM38437.2019.9013603], and [DOI: 10.1109/GLOBECOM42002.2020.9348238]. The associate editor coordinating the review of this article and approving it for publication was J. Yuan. *(Corresponding author: Chin-Wei Hsu.)*

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: chinweih@umich.edu; hunseok@umich.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2023.3239512.

Digital Object Identifier 10.1109/TCOMM.2023.3239512

HDM is a special case of sparse superposition codes (SPARC) [15], [16]. For the encoding/modulation of SPARC, multiple columns from a dictionary matrix are selected and superimposed together based on multiple sparse vectors that convey information. This modulation process is equivalent to projecting sparse vectors onto a hyper-dimensional space. For HDM, such projection is defined by a fast linear transformation and pseudo-random permutation. This resembles the principles of compressive sensing [17], whereas the unique modulation structure of HDM makes it feasible to apply efficient decoding algorithms. Moreover, its robustness against interference makes HDM appealing to low cost mMTC networks where many low power mMTC nodes transmit short packets in an asynchronous (grant-free) manner to a more resourceful (computation capability and energy) gateway receiver to form a star network with pure ALOHA random access [18].

The main contributions of this paper are summarized as follows:

1) We propose HDM with a cyclic redundancy check (CRC)-aided K-best decoding algorithm that can achieve very low error rate for short packets in additive white Gaussian noise (AWGN) channels. The proposed algorithm traverses a tree structure with pruning to find a candidate list for transmitted vectors with higher likelihood probabilities. CRC is then used to check all the candidates to find a valid codeword.

2) We propose extended algorithms to further combat both intra- and inter-network interference caused by packet collisions from unsynchronized transmissions. We evaluate different objective metrics in the K-best tree pruning algorithm to make the proposed scheme more robust when the system performance is limited by the interference, not by the channel noise.

3) We evaluate the packet error rate (PER) performance of HDM with extensive simulations and real-world experiments using a software-defined radio platform. Results show that HDM greatly outperforms SPARC and is on par with state-of-the-art short codes in AWGN channels. HDM outperforms TBCC and polar codes in interference-heavy scenarios.

The rest of the paper is organized as follows. In Section II, HDM is introduced and related prior works are discussed. We propose a CRC-aided K-best decoding algorithm for HDM in Section III. Then we extend the decoding algorithm in Section IV to combat interference in various mMTC network scenarios. Section V discusses practical considerations of HDM including the complexity, peak-to-average power ratio (PAPR), and choices of linear transform. Evaluation with computer simulations and hardware measurements in the real-world are provided in Section VI. Finally, Section VII concludes the paper.

II. Hyper-Dimensional Modulation

HDM [1], [2], [3] is inspired by hyper-dimensional computing [19] where hyper-dimensional vectors are used to represent information and perform cognitive computing. The hyper-dimensional presentation is tolerant of component failure, and thus is suitable for communicating message through wireless channels where excessive noise and interference can cause signal corruption. This robustness comes from redundant representation, in which information symbols are spread across many components in the hyper-dimensional vector [19].

The modulation process of HDM utilizes two observations from using hyper-dimensional vectors: *near-orthogonality* and *linearity*. Consider a hyper-dimensional vector space $\mathbb{C}^D$ where D is the dimension of the hyper-dimensional vector. Similarity between two energy-normalized vectors $\mathbf{x}$ and $\mathbf{y}$ can be measured by cross-correlation $\mathbf{x}^H\mathbf{y}$. Here, $\mathbf{x}^H$ stands for transpose conjugate of $\mathbf{x}$. Two identical vectors result in a cross-correlation output that is equal to the vector energy $\mathbf{x}^H\mathbf{x} = ||\mathbf{x}||^2$. The first observation that motivates HDM is the fact that two hyper-dimensional vectors whose components are i.i.d. zero-mean random variables have *nearly-orthogonal* cross-correlation; $\mathbf{x}^H\mathbf{y} \approx 0$ for a large D (hyper-dimension). Randomly-selected two vectors in the hyper-dimensional space have very small cross-correlation with high probability. The second observation is that the sum of two random vectors have high correlation with both vectors being added together. That is, the vector $\mathbf{x}+\mathbf{y}$ has high cross-correlation with both $\mathbf{x}$ and $\mathbf{y}$ since $\mathbf{x}^H(\mathbf{x}+\mathbf{y}) \approx \|\mathbf{x}\|^2$ because of near-orthogonality between $\mathbf{x}$ and $\mathbf{y}$. In other words, addition/superimposition of multiple independent hyper-dimensional vectors preserves the information that each vector carries without significant interference from each other although they are not strictly orthogonal. Based on these observations, HDM superimposes multiple (near-orthogonal) vectors to transmit numerous information bits using a single D-dimensional vector.

A. Modulation Process

The HDM modulation process is expressed by

$$\mathbf{s} = \sum_{i=1}^{V} \mathbf{A}_i\mathbf{x}_i = \sum_{i=1}^{V} \mathbf{P}_i\mathbf{W}\mathbf{x}_i = \sum_{i=1}^{V} \mathbf{P}_i\mathbf{W}(s_i\mathbf{e}_{p_i}) \quad (1)$$

where $\mathbf{s}$ denotes the complex-valued transmitted vector with dimension of $D \times 1$ ($\mathbf{s} \in \mathbb{C}^D$) and V is defined as the number of non-orthogonal vectors $\mathbf{A}_i\mathbf{x}_i$, $i = 1, \cdots, V$, that are transmitted at the same time. Each $\mathbf{A}_i\mathbf{x}_i$ is obtained by projecting an information vector $\mathbf{x}_i$ onto a hyper-dimensional space using a matrix $\mathbf{A}_i \in \mathbb{C}^{D\times D}$.

The information vectors $\mathbf{x}_i \in \mathbb{C}^D$ for $i = 1, \cdots, V$ have 'sparse' representations with only one non-zero element, embedding information bits in the position of the non-zero element by $\mathbf{e}_{p_i}$ and its non-zero value (phase) s_i. The sparse vector $\mathbf{e}_{p_i} = [e_0, \cdots, e_{D-1}]^T$ is a $D\times 1$ unit vector with $e_p = 0\ \forall\ p \neq p_i$ and $e_{p_i} = 1$. The non-zero position p_i is selected based on the information bits. We use a QPSK symbol $s_i \in \{\pm\sqrt{E_i/2}, \pm j\sqrt{E_i/2}\}$ for each non-zero element where E_i is the energy allocated to $\mathbf{x}_i$. The results in [16] (and our prior work [1]) show that QPSK is more efficient than other M-ary phase shift keying (PSK) schemes for SPARC (and HDM) to attain a lower PER given the same energy, bandwidth, and throughput. $\mathbb{E}\{\|\mathbf{s}\|^2\} = D$ and $E_i = D/V$, $i = 1, \cdots, V$ hold for energy-normalized HDM using equal-energy for each superimposed vector. In SPARC point of view, this modulation

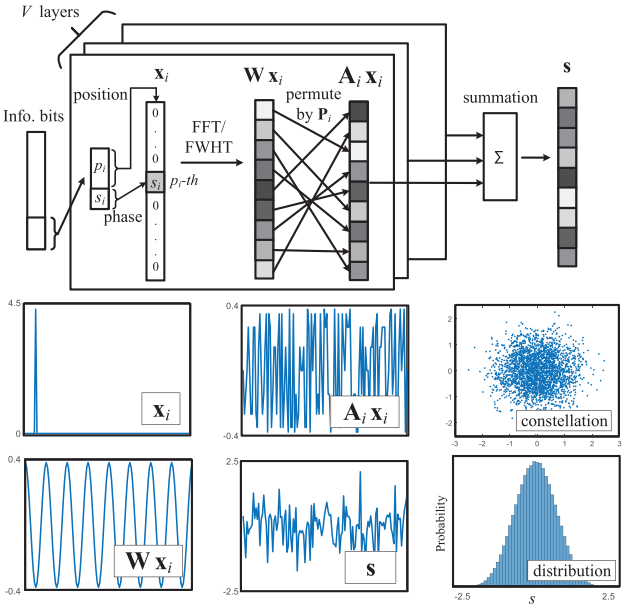


Fig. 1. HDM modulation process visualization.

process is equivalent to selecting a column from a dictionary $\mathbf{A}_i$ based on the position index p_i, and multiplying it with a QPSK symbol s_i.

In HDM, the projection matrix $\mathbf{A}_i = \mathbf{P}_i\mathbf{W}$ is obtained by a fast linear transformation $\mathbf{W}$ such as fast Fourier transform (FFT) or fast Walsh–Hadamard transform (FWHT) followed by a pseudo-random permutation $\mathbf{P}_i$. Note that $\mathbf{P}_i$ is different for each i but $\mathbf{W}$ is common to all i's. Since HDM uses a fast linear transform whose complexity is $O(D\log_2 D)$, it can be efficiently implemented in low power mMTC transmitters without explicitly computing costly matrix-vector multiplications with $\mathbf{W}$ that has a large dimension of $D \times D$.

The modulation process along with signal visualization of each step (only showing the real part) is summarized in Figure 1. In the modulation process, each independent vector goes through a separate layer/path with a different permutation pattern $\mathbf{P}_i$. Since HDM adds V i.i.d. vectors, elements of the final output vector $\mathbf{s}$ approximately follow a complex Gaussian distribution as shown in Figure 1 (bottom right).

One problem of using typical fast linear transform such as FFT or FWHT is that the first column of $\mathbf{W}$ has all ones, which makes the pseudo-random permutation meaningless and therefore violates the near-orthogonal property with other vectors. To avoid this problem, we use a pseudo-random vector whose elements are randomly selected from the set $\{\exp(j2\pi\frac{n}{D}), n = 0, \cdots, D-1\}$ to replace the all-ones column in $\mathbf{W}$ for FFT. Similarly, a pseudo-random vector with random 1 and -1 is used to replace that column for FWHT.

Parameters D and V determine the length and rate of transmission. For a given D, the number of information bits is proportional to V and each vector $\mathbf{x}_i$ conveys $\log_2 D$ information bits by the non-zero element position and additional 2 bits by the phase of the non-zero QPSK symbol. The modulation rate (or coding rate) C_R is the ratio of the number of information bits to the dimension of the transmitted vector, thus it is given by

$$C_R = \frac{V(\log_2 D + 2)}{D}. \tag{2}$$

With a unit energy constraint, the energy allocated to a vector $\mathbf{A}_i\mathbf{x}_i$ decreases as V increases and the inter-vector interference also increases at the same time because of non-orthogonality among vectors. Therefore, there is a fundamental trade-off between the rate C_R and the error probability. Vector dimension D also affects the performance of HDM transmission. Since the near-orthogonality improves as D increases, larger D results in lower error rate for a fixed rate C_R. However, it also leads to higher demodulation/decoding complexity, which poses another trade-off between complexity and performance. It is worth noting that as the information length increases with a larger D, the relative advantage of HDM diminishes compared to other schemes such as LDPC, Turbo, and polar codes which are capacity achieving when the message length is sufficiently long.

B. Related Works

The modulation process of HDM involves sparse vector mapping via index modulation. There are related prior works that also use sparse vectors with a random dictionary or index-based modulation for robust communication, thus they possess similar properties as in HDM. Here we describe prior schemes and their differences compared to our scheme.

Sparse Superposition Codes (SPARC): [15], [16], [20] are capacity achieving schemes in the AWGN channel. Their codewords are constructed by sparse linear combinations of entries in a dictionary, or equivalently, superposition of matrix-(sparse)vector products as in (1). Therefore, HDM can been considered as a special case of SPARC. For decoding of SPARC, approximate message passing algorithm [20] is widely used. SPARC is proven to achieve the channel capacity if the size of the dictionary is large enough under some parameter constraints. The main distinctions between HDM and SPARC are the size of the codeword and the design of the dictionary. SPARC typically uses a very long codeword length (e.g., 5000 bits) to achieve low error rates and the dictionary is constructed with randomly generated entries. On the other hand, HDM is designed for a relative short message length (e.g., $D = 128$) and it uses a structured modulation process that combines sparse index encoding, fast linear transform, and vector permutation to enable computationally-efficient yet powerful algorithms to achieve superior (compared to general SPARC schemes) error rate performance for short packets. Whereas SPARC typically operates with a relatively long outer code such as LDPC, the proposed HDM adopts a CRC-assisted error correction scheme which is more efficient for short packets.

Multi-Dimensional Modulation (MDM): [21], [22] is a modulation scheme that uses multi-dimensional lattices. By exploiting coding gain from the lattice and shaping gain, a well-designed constellation for MDM can outperform conventional QAM schemes with less energy per information bit for the same error rate without sacrificing bandwidth efficiency. Although higher dimensions improve both coding

and shaping gain, a practical well-designed multi-dimensional modulation scheme usually has a moderate dimension M because it becomes very difficult to design a good constellation in a high dimension space as the demodulation complexity dramatically increases with M. MDM is more beneficial in high SNR conditions when the constellation size can be relatively large and the spectral efficiency is $\geq$1 bps/Hz. Whereas, HDM is designed to operate in a low SNR or interference-dominated channel with a relatively low spectral efficiency of $<$1 bps/Hz for reliable communications of short messages. Unlike MDM that requires a deliberately designed codebook/constellation for a specific, relatively small dimension space of size M, HDM uses a fast linear transform and random permutation defined with a much larger dimension $D \gg M$.

Sparse Vector Coding (SVC): [23] is a non-orthogonal encoding scheme based on the theory of compressive sensing. The encoding process is similar to HDM as it selects columns from a dictionary according to a sparse vector. Dictionaries in SVC are typically constructed by randomly sampling from Gaussian or Bernoulli distribution without any elaborate structure, which is widely assumed for compressive sensing. A predefined table is used for mapping information bits to a sparse vector, whose non-zero elements are not restricted to be placed in different layers/sections as in HDM or SPARC. Multipath matching pursuit for sparse recovery [24] is a popular algorithm to decode SVC. In contrast, HDM uses an elaborate layer structure with a common linear transformation across all layers for efficient encoding and decoding. Sparse recovery algorithms generally do not work well for HDM because of its unique structure and elaborate constraints, which lead to dedicated (and efficient) decoding algorithms.

Orthogonal Frequency Division Multiplexing With Index Modulation (OFDM-IM): [25] uses indices of active subcarriers and modulated symbols on these subcarriers to embed information message bits via OFDM. Since index selection is followed by IFFT for OFDM, the modulation process resembles HDM. The goal of OFDM-IM is to improve robustness to inter-carrier interference caused by high mobility in OFDM systems. However, OFDM-IM only involves strictly orthogonal subcarriers and does not combine multiple non-orthogonal vectors. Thus its demodulation process consists of finding the most probable active subcarriers and demodulating their symbols without considering any interference caused by the superposition of non-orthogonal vectors. On the other hand, HDM is a non-orthogonal modulation scheme that applies element-wise permutations to fast linear transform (which does not have to be FFT) results before combining non-orthogonal vectors. And it employs a dedicated decoding algorithm to mitigate interference among superimposed vectors within the same packet.

Integer-HDM: [26] is a modulation scheme inspired by our original HDM [1] and thus it has a modulation structure similar to this work. Instead of using fast linear transformation on complex-valued vectors, Integer-HDM constrains the superimposed vectors to take values only from the binary set $\{1, -1\}$. This enables an even simpler decoding algorithm without compromising the error rate compared to the original HDM in [1].

Despite that all these prior schemes share some similarities in the modulation structure, their decoding algorithms significantly differ because of differences in their design principle and target operating scenarios. In this work, we propose advanced decoding methods for HDM to further improve the performance in AWGN and also in interference-limiting scenarios which are not explicitly considered in aforementioned prior schemes.

III. K-Best Decoding Algorithm

In this paper, we use *decoding* and *demodulation* interchangeably for HDM. Although HDM does not employ an explicit error correction scheme (except for CRC-based codeword selection), it exhibits superior or similar performance compared to conventional error correction codes applied to orthogonal modulation such as B/QPSK for short messages. Using the term *decoding* emphasizes the aspect of HDM imposing redundancy to increase robustness against signal corruption by noise or interference during transmission.

We consider an AWGN channel or narrowband frequency flat fading channel with perfect channel estimation. The received signal $\mathbf{y}$ can be represented by a model in (3), assuming the flat fading channel is equalized.

$$\mathbf{y} = \mathbf{s} + \mathbf{n} = \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i + \mathbf{n} \tag{3}$$

In (3), $\mathbf{n} \sim \mathcal{CN}(0, N_0\mathbf{I})$ is the complex Gaussian noise vector with zero mean and element-wise variance N_0. Note that under this model with an energy-normalized packet, the SNR is defined as $1/N_0$. The decoding process in AWGN can be considered as finding the optimal solution of the non-convex minimization problem:

$$\text{P1:} \quad \underset{\mathbf{x}_i \in \mathcal{X}, i=1,\cdots,V}{\operatorname{argmin}} \|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_2^2, \tag{4}$$

where $\mathcal{X}$ represents the set of all possible sparse information vectors $\mathbf{x}_i$ (i.e., each $\mathbf{x}_i$ contains only one non-zero QPSK symbol encoding $\log_2 D + 2$ bits by the position and phase).

A successive interference cancelling (SIC)-based decoding algorithm was proposed in the original HDM paper [1]. It first finds the vectors with the largest probability ignoring the interference and then performs interference cancellation for the next iteration of decoding.

While the SIC-based decoding algorithm in [1] is efficient, it does not directly solve the optimization problem of (4). A brute-force method to find the minimum of (4) by trying all possible combinations of $\mathbf{x}_i, i = 1, \cdots, V$ is practically infeasible due to excessive complexity. Therefore, in this section we propose a tree-based algorithm that finds a suboptimal solution of (4) through a K-best breath-first search that is similar to a variant in MIMO decoding [27].

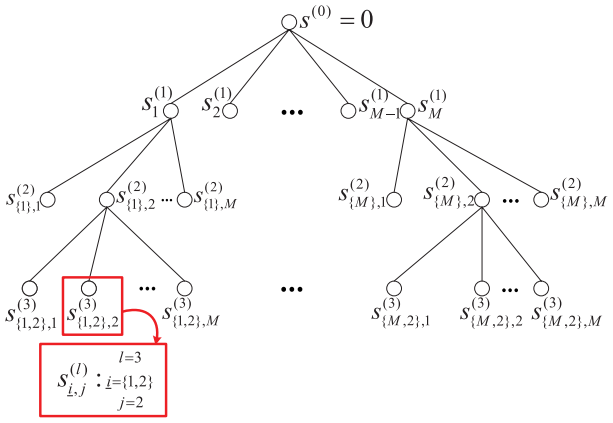


Fig. 2. The tree structure of K-best algorithm. $M = 4D$ is the total number of candidates $\mathbf{x}_l$ at each layer.

First observe that the objective in (4) can be expressed as

$$\begin{aligned}&\|\mathbf{y}-\sum_{i=1}^{V}\mathbf{A}_i\mathbf{x}_i\|_2^2\\&=\|\mathbf{y}-\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i\|_2^2+\|\mathbf{A}_V\mathbf{x}_V\|_2^2\\&\quad-2\Re\{\mathbf{y}^H\mathbf{A}_V\mathbf{x}_V\}+2\Re\{(\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i)^H\mathbf{A}_V\mathbf{x}_V)\},\end{aligned}\quad(5)$$

where $\Re\{\}$ and $\Im\{\}$ are the operations of taking real and imaginary parts of a complex number or vector, respectively. In (5), the right-hand side consists of four terms. The first term $\|\mathbf{y}-\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i\|_2^2$ has the same form as the left-hand side except that the summation is now from 1 to $V-1$. The second term $\|\mathbf{A}_V\mathbf{x}_V\|_2^2$ is constant regardless of $\mathbf{x}_V$ as long as the fast linear transformation matrix $\mathbf{W}$ has equal norm columns (as in FFT and FWHT). The third term is the correlation between $\mathbf{y}$ and $\mathbf{A}_V\mathbf{x}_V$. Finally, the last term is the correlation between $\mathbf{A}_V\mathbf{x}_V$ and the accumulative vector $\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i$. With recursion, the first term can be further decomposed until only $\mathbf{y}$ remains.

Subtracting constant terms $\|\mathbf{A}_i\mathbf{x}_i\|_2^2$ for $i=1,\cdots,V$ and division by 2 does not change the solution of (4). Hence we define the *score metric* at recursion layer l as $s^{(l)}=\frac{1}{2}(\|\mathbf{y}-\sum_{i=1}^{l}\mathbf{A}_i\mathbf{x}_i\|_2^2-\sum_{i=1}^{l}\|\mathbf{A}_i\mathbf{x}_i\|_2^2)$, which can be expressed in an iterative form:

$$s^{(l)}=s^{(l-1)}-\Re\{\mathbf{x}_l^H\mathbf{A}_l^H\mathbf{y}\}+\Re\{\mathbf{x}_l^H\mathbf{A}_l^H\mathbf{u}^{(l-1)}\},\quad(6)$$

where $\mathbf{u}^{(l)}=\sum_{i=1}^{l}\mathbf{A}_i\mathbf{x}_i$. This score metric depends on the selection of sparse vectors up to layer l, i.e., $\mathbf{x}_i$ for $i=1,\cdots,l$.

The objective is to minimize (6) for the last layer V, $s^{(V)}$. Thus we find the minimum metric through a tree structure by evaluating candidate sparse vectors $\mathbf{x}_l$ for each layer. Note that at each node of the tree, we calculate the metric (6) for each candidate of $\mathbf{x}_l$ with given candidates determined by all previous layers from 1 to $l-1$ (i.e., $s^{(l-1)}$ and $\mathbf{u}^{(l-1)}$).

The tree structure is illustrated in Figure 2. For each node, we calculate its children's metric based on its parent and ancestor path using an iterative equation given as

$$s_{\underline{i},j}^{(l)}=s_{\underline{i}}^{(l-1)}-\Re\{\mathbf{x}_{l,j}^H\mathbf{A}_l^H\mathbf{y}\}+\Re\{\mathbf{x}_{l,j}^H\mathbf{A}_l^H\mathbf{u}_{\underline{i}}^{(l-1)}\}\quad(7)$$

where j is the candidate index of possible $\mathbf{x}_l$, and $\underline{i}=\{i_1,i_2,\cdots,i_{l-1}\}$ is the index list of previously chosen paths/vectors by its ancestor nodes. The accumulative vector $\mathbf{u}_{\underline{i}}^{(l-1)}$ is the interference term given by candidates chosen by parent/ancestor layers.

Without pruning, the number of nodes and the size of possible x_l candidates grow exponentially as we go deeper into the tree. Since the paths with relatively large metrics are very unlikely to be part of the transmitted vector set that minimizes the metric, they can be pruned without degrading the performance much. Therefore, at the i-th layer we only keep best K_i candidates with lowest metrics and prune all the others. The value K_i is dynamically chosen to include all nodes whose metrics are not greater than the minimum metric of the i-th layer plus a pre-determined threshold. Our algorithm also defines a pre-determined $K_{\max}$ to prevent the dynamic K_i being too large so that $K_i \le K_{\max}$ when more than $K_{\max}$ nodes satisfy the aforementioned condition. The iteration continues until the last layer, where no pruning happens. We denote the average number of candidates kept at each layer as $\overline{K}=\frac{1}{V-1}\sum_{i=1}^{V-1}K_i$.

There are efficient ways to calculate the metric (7). Since only one element of $\mathbf{x}_l$ is a non-zero QPSK symbol, evaluating the last two terms in (7) is equivalent to simply choosing a single real or imaginary number multiplied with different signs from the elements of $\mathbf{A}_l^H(\mathbf{y}-\mathbf{u}_{\underline{i}}^{(l-1)})$, which can be computed by a fast linear transform of $(\mathbf{y}-\mathbf{u}_{\underline{i}}^{(l-1)})$ followed by permutation (without matrix–vector multiplication).

One potential issue of the K-best algorithm is that a wrong pruning decision made in an upper layer can not be recovered in lower layers. To mitigate this issue, we propose a strategy to (re-)sort the order of decoding layers based on the score metric along the tree traversal. At each layer, we first evaluate minimum metrics of all remaining layers as the possible next layer based on (7) using the up-to-now best candidate. Then the layer with the lowest metric is selected as the next layer to proceed. This per-layer re-sorting approach significantly (up to 2 dB at PER=10^{-3} compared to no sorting) improves the error rate performance of the K-best HDM decoding.

Finally, CRC-assisted error correction is applied to further increase the error rate performance of the proposed decoding algorithm. Since the K-best algorithm produces a list of candidates at the end, one can try each of them with the order of ascending metric until the candidate passes CRC. The error rate is improved because even in the event that the correct vector does not minimize the metric (6), it is still highly probable to be contained in the final candidate list.

IV. HDM for Massive Machine-Type Communication Networks

A. mMTC Network Model

In a star topology mMTC network that allows grant-free transmissions, plenty of devices can transmit packets simultaneously, thus causing overwhelming interference at the

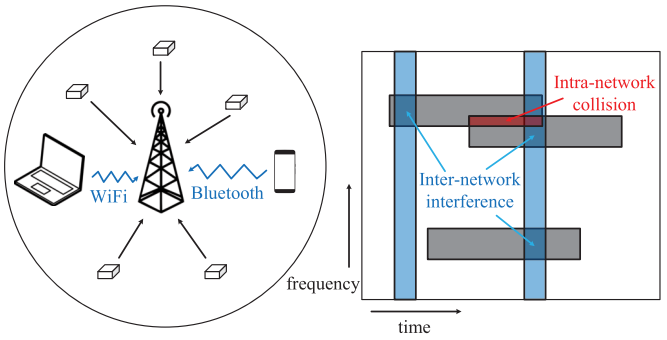


Fig. 3. mMTC network and interference. Left: Star network topology with pure ALOHA. Right: Grey blocks are HDM packets and blue stripes are wideband interference packets.

gateway receiver. In this case, the channel noise may not be the performance limiting factor when the interference is stronger than the noise. An AWGN channel model may not capture the performance of a system in such a scenario, and the decoding algorithm designed for AWGN channels may experience significant performance degradation with interference. In this section, we propose modified versions of the K-best decoding algorithm to make HDM more robust against interference in mMTC networks.

We consider an narrowband uplink star network with multiple transmitters and one receiver. This network topology is widely adopted in mMTC networks because of its simplicity and efficiency [18]. Each device can transmit a packet at any time (i.e., grant-free) without considering other devices. Moreover, carrier frequencies of transmitters are assumed to be uniformly distributed in a pre-defined frequency range [28]. This is because the frequency uncertainty may extend over multiple times of the signal bandwidth of narrowband systems. For example, a low cost crystal with 50 ppm accuracy results in 120kHz carrier frequency offset for the 2.4GHz carrier frequency, which is much larger than the bandwidth of an (ultra) narrowband scheme that often operates with $<$1kHz bandwidth [29]. In the considered scenario, the network adopts a pure unslotted (grant-free) ALOHA scheme in both time and frequency domains, thus multiple packet transmissions can (partly) collide in time and frequency.

In practice, not just transmitters in the same network but also transmitters in other heterogeneous networks using the same band can cause interference. A narrowband mMTC network operating in the 2.4GHz ISM band, for example, experiences inter-network interference from other technologies such as WiFi, Bluetooth Low Energy (BLE), etc. Since the WiFi and BLE bandwidth is $\geq$20MHz and 1MHz, respectively, with a typical packet duration of $\leq$2ms, interference from these networks are wideband and short compared to narrowband mMTC packets as illustrated in Figure 3. While the relatively wideband interference can easily overlap with the desired narrowband mMTC signal, it may only affect a few symbols of an mMTC packet because of its wideband (short symbol) nature.

In our scenario depicted in Figure 3, we categorize the interference into two types: *intra*-network and *inter*-network interference. Intra-network interference comes from other transmitters in the same mMTC network and the statistics of this interference is known to the gateway. On the other hand, inter-network interference is introduced by other heterogeneous networks and the statistics is unknown although it can be modeled as a random arrival process of wideband short pulses/packets.

The receiver signal $\mathbf{y}$ in this mMTC network model with potential interference can be expressed as

$$\mathbf{y} = \mathbf{s} + \mathbf{n} + \mathbf{w} = \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i + \mathbf{n} + \mathbf{w} \tag{8}$$

where $\mathbf{w}$ denotes the sum of all potential interference sources, including intra-network and inter-network interference. Note that elements in $\mathbf{w}$ may not have constant power, and do not necessarily behave as i.i.d. Gaussian random variables. We assume that the receiver is aware of its presence, but may or may not know its statistics. The proposed algorithms try to decode $\mathbf{x}_i$'s without jointly decoding the interference source $\mathbf{w}$.

B. Dealing With Intra-Network Interference

The P1 formulation of (4) assumes an AWGN channel with constant noise variance given observed signal $\mathbf{y}$. However, when the interference exists, the variance of interference plus noise is no longer constant across $\mathbf{y}$ (i.e., a packet). Under this scenario, the solution of (4) is no longer optimal for estimating the transmitted vector.

Considering the case that all devices in the network are transmitting HDM signals, interference can be approximated as a Gaussian random variable as discussed in Section II. And, in a star network where the gateway is listening to all transmitting devices, it is possible that the gateway receiver first identifies the timing and average received power of the signal from each device (via packet detection) before decoding individual packets that are collided.

Assuming the receiver has the information of timing and power level of the interference caused by each collided transmission, the problem statement in (4) can be modified to match this scenario by using weighted L2-norm as in (9)

$$\text{P2:} \quad \underset{\mathbf{x}_i \in \mathcal{X}, i=1,\cdots,V}{\operatorname{argmin}} \|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_{\mathbf{C}}^2, \tag{9}$$

where $\|\mathbf{x}\|_{\mathbf{C}} = \|\mathbf{C}^{\frac{1}{2}}\mathbf{x}\|_2$ and $\mathbf{C}$ is a diagonal matrix. As the interference plus noise has element-dependent variance, each diagonal element of $\mathbf{C}$ can be found by adding the average interference power level to the noise variance at a corresponding sample index, and then taking the inverse such that (10) holds.

$$\mathbf{C}_{j,j} = \frac{1}{\sum_{k \in \mathcal{I}_j} P_k + N_0} \tag{10}$$

In (10), $\mathcal{I}_j$ is the set containing all packets colliding with the desired signal at time index j, and P_k is the average interference power from interfering device k. A more detailed derivation of obtaining P_k in a star network with a pure ALOHA scheme in both time and frequency domain can be found in [2].

To modify the K-best decoding algorithm for weighted L2-norm, observe that the objective function (5) now changes to

$$\begin{aligned}
&\|\mathbf{y}-\sum_{i=1}^{V}\mathbf{A}_i\mathbf{x}_i\|_{\mathbf{C}}^2 \\
&= \|\mathbf{C}^{\frac{1}{2}}\mathbf{y}-\mathbf{C}^{\frac{1}{2}}\sum_{i=1}^{V}\mathbf{A}_i\mathbf{x}_i\|_2^2 \\
&= \|\mathbf{C}^{\frac{1}{2}}\mathbf{y}-\mathbf{C}^{\frac{1}{2}}\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i\|_2^2+\|\mathbf{C}^{\frac{1}{2}}\mathbf{A}_V\mathbf{x}_V\|_2^2 \\
&\quad -2\Re\Big\{(\mathbf{C}^{\frac{1}{2}}\mathbf{y})^H\mathbf{C}^{\frac{1}{2}}\mathbf{A}_V\mathbf{x}_V \\
&\quad -(\mathbf{C}^{\frac{1}{2}}\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i)^H\mathbf{C}^{\frac{1}{2}}\mathbf{A}_V\mathbf{x}_V)\Big\} \\
&= \|\mathbf{y}-\sum_{i=1}^{V}\mathbf{A}_i\mathbf{x}_i\|_{\mathbf{C}}^2+\|\mathbf{A}_V\mathbf{x}_V\|_{\mathbf{C}}^2 \\
&\quad -2\Re\{\tilde{\mathbf{y}}^H\mathbf{A}_V\mathbf{x}_V\}+2\Re\{(\tilde{\mathbf{u}}^{(V-1)})^H\mathbf{A}_V\mathbf{x}_V)\} \quad (11)
\end{aligned}$$

where $\tilde{\mathbf{y}}=\mathbf{C}\mathbf{y}$ and $\tilde{\mathbf{u}}^{(V-1)}=\mathbf{C}\mathbf{u}^{(V-1)}$.

Notice that

$$\begin{aligned}
\|\mathbf{A}_i\mathbf{x}_i\|_{\mathbf{C}}^2 &= \mathbf{x}_i^H\mathbf{A}_i^H\mathbf{C}\mathbf{A}_i\mathbf{x}_i \\
&= \mathbf{x}_i^H\mathbf{W}^H\mathbf{P}_i^H\mathbf{C}\mathbf{P}_i\mathbf{W}\mathbf{x}_i=\mathbf{x}_i^H\mathbf{W}^H\mathbf{\Lambda}\mathbf{W}\mathbf{x}_i \\
&= \mathbf{x}_i^H\mathbf{Q}\mathbf{x}_i=\frac{D}{V}\cdot\mathbf{Q}_{j,j}
\end{aligned}$$

holds where j denotes the non-zero position of $\mathbf{x}_i$, $\mathbf{\Lambda}=\mathbf{P}_i^H\mathbf{C}\mathbf{P}_i$ is a diagonal matrix, and $\mathbf{Q}=\mathbf{W}^H\mathbf{\Lambda}\mathbf{W}$.

As we choose a fast linear transformation matrix $\mathbf{W}$ with constant magnitude elements such as FFT or FWHT, $\|\mathbf{A}_i\mathbf{x}_i\|_{\mathbf{C}}^2$ is constant regardless of the choice of $\mathbf{x}_i$. Therefore, the metric update rule in (6) can be re-written with minor modifications:

$$s^{(l)}=s^{(l-1)}-\Re\{\mathbf{x}_l^H\mathbf{A}_l^H\tilde{\mathbf{y}}\}+\Re\{\mathbf{x}_l^H\mathbf{A}_l^H\tilde{\mathbf{u}}^{(l-1)}\}. \quad (12)$$

Moreover, the computation complexity does not increase much since $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{u}}$ can be obtained by element-wise multiplications since $\mathbf{C}$ is diagonal. Except for this updated metric calculation, the remaining K-best algorithm is identical to the AWGN channel case.

C. Dealing With Inter-Network Interference

When the inter-network interference is involved, the optimization problem P1 in (4) does not yield the optimal performance. For a narrowband mMTC scenario, we assume the inter-network interference burst length is much shorter than the length of the desired HDM packet, and independent random arrival processes can cause multiple interference sources collide with a single HDM packet. In this scenario, it is reasonable to assume the receiver does not know properties of interference such as the average/instantaneous power level and position of interference bursts within a desired packet. This assumption holds for an example scenario where a narrowband mMTC network using HDM operates in the 2.4GHz ISM band with heavy interference coming from WiFi and BLE. A packet from WiFi or BLE is much shorter ($\leq$ 2 ms) than a narrowband (e.g., $D=128$ with 1 kHz symbol rate) mMTC packet, and the HDM receiver does not have the capability of demodulating all WiFi and Bluetooth packets that collide with the HDM packet.

Performance of decoding algorithms under such severe interference can significantly degrade because of sporadic interference causing occasional large deviation (in Euclidean distance) from the transmitted samples. These events can result in large L2-norm during the objective function evaluation. One technique to alleviate this problem is to set a saturation threshold on the received sample to prevent large offsets from the transmitted signal caused by the sporadic strong interference [30]. Another strategy is to use an alternative metric to replace the L2-norm. We propose to use L1-norm as an alternative metric because it is less sensitive than L2-norm to sporadic outlier elements. The optimization in this case changes from L2- to L1-norm objective:

$$\text{P3:}\quad \underset{\mathbf{x}_i\in\mathcal{X},i=1,\cdots,V}{\operatorname{argmin}}\ \|\mathbf{y}-\sum_{i=1}^{V}\mathbf{A}_i\mathbf{x}_i\|_1. \quad (13)$$

Note that the solution of P3 is indeed optimal when the noise plus interference follows Laplace distribution, which has a longer tail compared to Gaussian distribution.

The L1-norm in (13) can not be decomposed in the same form as the L2-norm in (4) with iterative equations. Thus we reformulate (13) with real-valued vectors and matrices with a goal to obtain an iterative additive form to replace (13).

Consider two real-valued scalars $a,b\in\mathbb{R}$ and observe that $|a+b|=|a|+|b|-2\cdot\mathbb{1}(ab<0)\cdot\min(|a|,|b|)$ holds, where $\mathbb{1}(\cdot)$ is the indicator function. Using this, we decompose the L1-norm of the sum of two real-value vectors as

$$\begin{aligned}
&\|\mathbf{a}+\mathbf{b}\|_1 \\
&\quad =\|\mathbf{a}\|_1+\|\mathbf{b}\|_1-2\sum_{i=1}^{D}\mathbb{1}(a_ib_i<0)\cdot\min(|a_i|,|b_i|) \quad (14)
\end{aligned}$$

where $\mathbf{a},\mathbf{b}\in\mathbb{R}^D$, and a_i,b_i denotes the i-th element of $\mathbf{a},\mathbf{b}$.

Now, the objective function (13) can be decomposed as

$$\begin{aligned}
&\|\mathbf{y}-\sum_{i=1}^{V}\mathbf{A}_i\mathbf{x}_i\|_1 \\
&= \|\mathbf{y}-\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i\|_1+\|\mathbf{A}_V\mathbf{x}_V\|_1 \\
&\quad -2\cdot\mathbf{1}^T\big(\mathbb{1}(\mathbf{r}_{V-1}\circ\mathbf{A}_V\mathbf{x}_V>0)\circ\min(|\mathbf{r}_{V-1}|,|\mathbf{A}_V\mathbf{x}_V|)\big) \quad (15)
\end{aligned}$$

where $\mathbf{r}_{V-1}=\mathbf{y}-\sum_{i=1}^{V-1}\mathbf{A}_i\mathbf{x}_i$, $\mathbf{1}$ is a vector with all ones, and $\circ$ denotes element-wise multiplication. The term $\|\mathbf{A}_V\mathbf{x}_V\|_1$ is constant if $\mathbf{A}_V$ has constant L1-norm columns. Hence, the iterative metric calculation for L1-norm has the form:

$$\begin{aligned}
&s^{(l)} \\
&\quad =s^{(l-1)}-\mathbf{1}^T\big(\mathbb{1}(\mathbf{r}_{l-1}\circ\mathbf{A}_l\mathbf{x}_l>0)\circ\min(|\mathbf{r}_{l-1}|,|\mathbf{A}_l\mathbf{x}_l|)\big). \quad (16)
\end{aligned}$$

The same K-best algorithm can be used with this recursive L1-norm metric in (16) replacing the previous L2-norm

metric updating. Note that (16) does not involve matrix-vector multiplications since $\mathbf{x}_l$ has only one non-zero element and the other terms are evaluated by just selecting sign and magnitude values from two vectors.

However, the formulation of (16) only works for real-valued vectors, and this L1-norm metric cannot be directly applied to HDM that involves complex-valued vectors. Addressing this issue, we relax the problem and minimize an upper bound of the L1-norm.

Observe that for a complex-valued vector $\mathbf{z}$,

$$\begin{aligned}\|\mathbf{z}\|_1 &= \sum_i \sqrt{\Re\{z_i\}^2 + \Im\{z_i\}^2} \\ &\le \sum_i |\Re\{z_i\}| + |\Im\{z_i\}| = \| \begin{bmatrix} \Re\{\mathbf{z}\} \\ \Im\{\mathbf{z}\} \end{bmatrix} \|_1\end{aligned}$$

holds. Then by denoting $\mathbf{y}' = [\Re\{\mathbf{y}\}^T \ \Im\{\mathbf{y}\}^T]^T$, $\mathbf{x}'_i = [\Re\{\mathbf{x}_i\}^T \ \Im\{\mathbf{x}_i\}^T]^T$ and $\mathbf{A}'_i = \begin{bmatrix} \Re\{\mathbf{A}\} & -\Im\{\mathbf{A}\} \\ \Im\{\mathbf{A}\} & \Re\{\mathbf{A}\} \end{bmatrix}$, a relaxed problem of (13) can be expressed as

$$\text{P3}': \underset{\mathbf{x}_i \in \mathcal{X}, i=1,\cdots,V}{\operatorname{argmin}} \|\mathbf{y}' - \sum_{i=1}^{V} \mathbf{A}'_i \mathbf{x}'_i\|_1, \tag{17}$$

which is now real-valued and can be solved with the proposed K-best algorithm via L1-norm minimization. Note that, this L1 optimization formulation requires a fast linear transform matrix $\mathbf{W}$ whose real and imaginary parts have constant L1-norm columns. Because of this requirement, we use FWHT instead of FFT when the L1-norm metric is adopted. An alternative method (with worse performance) that separates the I and Q channels and sends two independent real-valued HDM vectors is described in our prior work [3].

V. Discussion

A. Decoding Complexity and Latency

Besides the error rate performance, decoding complexity is a crucial factor when choosing a practical scheme. In this section we discuss the complexity of the K-best based HDM decoding algorithm proposed in Section III.

The complexity of the proposed K-best decoding algorithm can be estimated by summing the number of operations of the following four parts: 1) Sorting and selecting the next layer for decoding, 2) Calculating metrics, 3) Selecting survivor nodes, and 4) Calculating cumulative interference vector. Note that these steps are repeated for each layer processing. For simplicity, we assume that all layers has the same $K_i = \overline{K}$ which is the average value. FFT is used for the fast linear transform.

1) Sorting and Selecting the Next Layer: At the beginning of each layer, we evaluate $\Re\{\mathbf{x}_l^H \mathbf{A}_l^H (\mathbf{y} - \mathbf{u})\}$ for all remaining layers as a potential l-th layer and select the one with the highest value for the l-th layer processing. This requires $V_{\text{rem}}(4D \log_2 D - 5D + 8)$ operations, where V_{rem} indicates the number of remaining layers to decode. The value inside the parentheses can be obtained in a similar way in the next step 2).

2) Calculating Metrics: Metric calculation is performed for every node to evaluate (7). Starting with $\overline{K}$ surviving nodes from the last layer, we first calculate $-\Re\{\mathbf{x}_l^H \mathbf{A}_l^H (\mathbf{y} - \mathbf{u}_k)\}$ in (7) for the k-th node among $\overline{K}$. To get the vector $(\mathbf{y} - \mathbf{u}_k)$ we need D additions. Since HDM utilizes a fast linear transformation and $\mathbf{x}_l$ has only one non-zero element, all $\mathbf{x}_l$ candidates can be evaluated by performing FFT (and pseudo-random permutation), which requires $4D \log_2 D - 6D + 8$ operations. Phase rotation due to the QPSK modulation is equivalent to taking real and imaginary parts of the result with different signs without additional operations. Finally, the results are added to the metric of the parent node, which requires another $4D$ additions. Therefore, the total number of operations for this step is $\overline{K}(4D \log_2 D - D + 8)$.

3) Selecting Survivor Nodes: To select $\overline{K}$ nodes out of $4\overline{K}D$, we use partial QuickSort [31]. This step requires $8\overline{K}D + (\ln \overline{K}D + 1.27)(2\overline{K} - 4) - 6\overline{K} + 6$ comparisons on average.

4) Cumulative Interference Vector: For each surviving node, we calculate the cumulative interference vector $\mathbf{u}_k = \mathbf{u}_{\text{old},k} + \mathbf{u}_{\text{new},k}$, where the former is the interference from previous layers, and the latter is the newly introduced interference from the current layer. This requires $\overline{K}D$ additions.

In summary, the total number of operations required to process all V layers is

$$\begin{aligned}N_{\text{op}} = \overline{K}(V-1)\Big(&4D \log_2 D + 2 \ln \overline{K}D + 8D + 4.54\Big) \\ &+2(V^2 + V)D \log_2 D - \frac{D}{2}(5V^2 + 5V - 8) \\ &+2(V-1)(\ln \overline{K}D + 1.27) + 4V^2 + 10V - 6.\end{aligned} \tag{18}$$

It has $O(V\overline{K}D \log_2 D)$ complexity as the first term dominates when $\overline{K}$ is large. For a $\{D = 128, V = 8\}$ HDM packet with 64 information bits (excluding CRC), the number of operations per bit is about $505\overline{K} + 1667$. Note, for comparison, that the number of operations required for polar and TBCC decoding for the same rate is on the order of $10^3 - 10^4$ per information bit depending either on the list size of the successive cancellation list (SCL) polar decoder or the constraint length of the TBCC [32].

For practical systems, the number of operations is not the only complexity indicator and it is not necessarily proportional to the latency (or run-time) in modern parallel computing processors and accelerators. While the HDM decoding complexity scales with $V\overline{K}D \log_2 D$, the decoder can take the advantage of a fully parallelizable structure of K-best decoding to reduce the decoding latency in practical implementations on many-core processors and hardware accelerators. When the gateway has sufficient compute resources, steps 2) and 4) can be calculated in parallel for different candidates, removing the computation latency dependency on $\overline{K}$. Moreover, step 3) can also use a parallelized version of the algorithm [33] to achieve the latency of $O(V \log\log \overline{K}D)$. The resulting decoding time complexity, or latency, becomes $O(VD \log D + V \log\log \overline{K}D)$, which only grows with $\log\log \overline{K}$. This implies that increasing $\overline{K}$ for better PER performance does not significantly increase the latency as long as the receiver has a proportional number of parallel processing elements.

It is worth noting that, although SCL decoding for polar codes can also be parallelized [34], the number of newly generated paths increases exponentially at each time step with the number of parallel decoders. This makes parallel execution of the algorithm practically difficult, unlike the proposed K-best decoding for HDM.

B. PAPR and Clipping

Peak-to-average power ratio (PAPR) is an important metric for a practical modulation scheme because a high PAPR requires a wide dynamic range power amplifier (PA). A PA typically exhibits the highest power efficiency at its peak output power but a large PAPR forces the PA to mostly operate around the average power where power efficiency is significantly reduced. A constant envelope (baseband PAPR is 1) or low PAPR modulation scheme such as BPSK/QPSK is generally preferred to achieve higher PA efficiency.

However, the use of a fast linear transform and superposition of V vectors results in a relatively high PAPR of the HDM signal. When a normalized HDM vector has the element-wise average power of 1, the worst case peak value may occur when all V superimposed vector $\mathbf{A}_i\mathbf{x}_i$'s have the same phase on the same element after the fast linear transformation and pseudo-random permutation. As each element of superimposed vector $\mathbf{A}_i\mathbf{x}_i$ has the average amplitude of $\sqrt{1/V}$, the worst case PAPR of HDM is $10\log_{10}V$ dB. Note that most practical systems apply an additional pulse shaping filter before the signal goes into a PA to tighten the spectrum. This further (by $\approx$2.5 dB) increases the PAPR regardless of the modulation type.

One practical method to constrain the PAPR to a lower target level is to apply intentional deliberate clipping to the signal as discussed in [35]. Consider a power normalized HDM vector whose unclipped samples have the form of $s = Ae^{j\phi}$ where A is the sample amplitude, ϕ is the sample phase, and $\mathbb{E}\{A^2\} = 1$ holds. The clipped sample $\tilde{s}$ after the signal clipping at a pre-defined level c is obtained by

$$\tilde{s} = \begin{cases} Ae^{j\phi}, & \text{if } A \le c \\ ce^{j\phi}, & \text{otherwise.} \end{cases} \tag{19}$$

The clipped signal can be regarded as the combination of the desired signal and distortion. With an assumption that HDM samples are approximated by complex Gaussian random variables, the PAPR after clipping (but before pulse shaping) can be calculated [36] by

$$\text{PAPR} = \frac{c^2}{1 - e^{-c^2}}. \tag{20}$$

The PAPR after pulse shaping depends on the pulse shaping function itself. Since it is not straightforward to characterize the impact of pulse shaping on PAPR with deliberate clipping [36], we use numerical analysis in Section VI to quantify the HDM's PAPR with pulse shaping.

Denoting the *after clipping* average signal power E_s and noise variance N_0, the signal-to-noise-plus-distortion ratio (SNDR) after clipping is obtained [36] by

$$\text{SNDR} = \frac{\mathcal{K}E_s/N_0}{(1-\mathcal{K})E_s/N_0 + 1}, \tag{21}$$

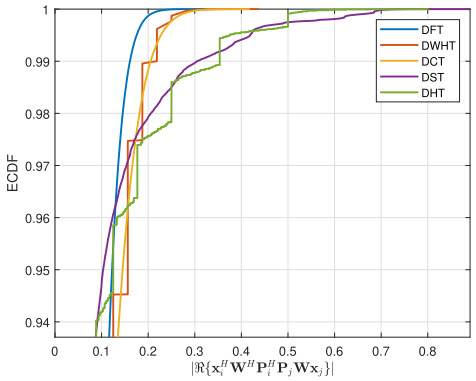


Fig. 4. ECDF of interference magnitude. The size of $\mathbf{W}$ is 128×128.

where $\mathcal{K}$ is the signal attenuation factor given by

$$\mathcal{K} = \frac{\left(1 - e^{-c^2} + \frac{\sqrt{\pi}c}{2}\text{erfc}(c)\right)^2}{1 - e^{-c^2}}. \tag{22}$$

The parameter c determines the tradeoff between PAPR reduction and SNDR degradation.

Since HDM is designed to operate in relatively high noise/interference scenarios, it can tolerate moderate clipping distortion as long as it does not dominate the channel noise. One can choose the clipping parameter c such that the corresponding SNDR (21) is comparable to the original (pre-clipping) SNR given N_0 without significant PER degradation. The impact of signal clipping with various levels of c on PER is evaluated in Section VI.

C. Linear Transforms for HDM

There are multiple options for the (fast) linear transform matrix $\mathbf{W}$ in HDM. In this section, we discuss the impact of choosing different transforms. While there is no strict constraint on the orthogonality among transform matrix columns for HDM, we only consider common discrete linear transforms whose matrices have orthogonal columns. This means that all valid HDM vectors that belong to the same layer are also orthogonal and the HDM performance is governed by the interference between different layers that use different pseudo-random permutations. This can be seen in (5), where the last term is due to the interference from the other layers.

To determine the 'effectiveness' of a certain (fast) linear transform, we quantify its inter-layer interference by analyzing the statistics of the last term in (5). Roughly speaking, a linear transform that does not have large amplitude realizations of $\Re\{\mathbf{x}_i^H\mathbf{A}_i^H\mathbf{A}_j\mathbf{x}_j\} = \Re\{\mathbf{x}_i^H\mathbf{W}^H\mathbf{P}_i^H\mathbf{P}_j\mathbf{W}\mathbf{x}_j\}$ leads to a lower error rate for HDM decoding. Note that $\mathbf{x}_i, \mathbf{x}_j, \mathbf{P}_i, \mathbf{P}_j$ are random variables/matrices.

Figure 4 plots the empirical CDF of $\Re\{\mathbf{x}_i^H\mathbf{W}^H\mathbf{P}_i^H\mathbf{P}_j\mathbf{W}\mathbf{x}_j\}$ for different linear transform matrices $\mathbf{W}$. We test the following five discrete linear transforms that allow fast/efficient algorithms such as FFT: discrete Fourier transform (DFT), discrete cosine transform (DCT), discrete Walsh-Hadamard transform (DWHT), discrete Slant transform (DST), and discrete Haar transform (DHT) [37]. As shown in the figure, DFT has fewest large amplitude values, while DWHT

and DCT are behind it. DST and DHT have much more large amplitude values than the others. Based on this observation, we use DFT (i.e., FFT) for all cases except for the L1-norm minimization algorithm (i.e., P3 formulation in (13)). Recall that the L1-norm minimization algorithm cannot use DFT/FFT because the real and imaginary parts of a DFT matrix do not satisfy the constant L1-norm column condition. Hence we use DWHT/FWHT instead as it is more computationally efficient than DCT which exhibits a similar CDF.

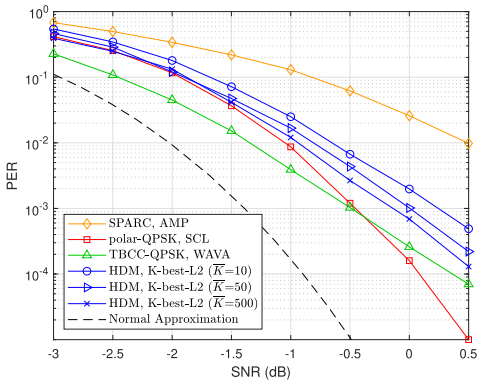
VI. Evaluation

A. Simulation Results

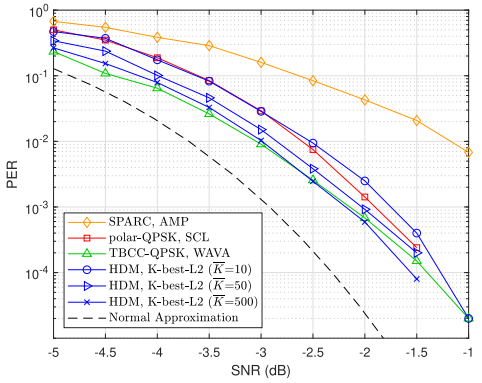
The proposed HDM schemes are compared with QPSK modulation protected by a 3GPP specified CRC-aided polar code with an SCL decoding algorithm [38], [39] and a tail-biting convolutional code (TBCC) decoded by a wrap-around viterbi algorithm (WAVA) [40]. Both polar code and TBCC configurations are known to be very robust for short-length codes [41]. For the fast linear transformation in HDM, we use FFT for HDM with (weighted) L2-norm minimization (in AWGN and intra-network interference-heavy scenarios) and FWHT for L1-norm minimization (in inter-network interference-heavy scenarios).

For the narrowband mMTC scenario, we assume a short packet with length $D = 128$. The number of information bits is either 64 for a rate-1/2 packet, or 43 for a rate-1/3 packet. For HDM and polar codes, additional CRC bits are concatenated with the information bits before modulation/encoding. Polar codes use an 11-bit CRC as in the 3GPP uplink setting, while HDM use an 8-bit and 11-bit CRC for rate-1/2 and rate-1/3 packet, respectively. These additional CRC-bits require the information bits to be coded with a higher rate (to keep the effective rate unchanged with and without the CRC) which could potentially lead to a worse error rate. However, the SNR gain of CRC-based valid codeword/vector identification offsets the loss of using a higher rate for information bits. TBCC does not utilize any CRC bits as the decoding algorithm does not create a list (unlike polar and HDM decoding) and thus CRC is not directly usable to correct decoding errors. Regardless of the CRC-usage, the rate of HDM, polar-QPSK and TBCC-QPSK schemes is identical as we send the same number of information bits (64 or 43) with the same number of complex-valued channel use: $D = 128$ for HDM and 128 QPSK symbols for polar-QPSK and TBCC-QPSK. Note that for the rate-1/3 TBCC-QPSK, 2-bit punctuation is used to change the length of (258,43)-TBCC to 256 bits before QPSK transmission. HDM superimposes $V = 8$ and $V = 6$ layers of vectors for rate-1/2 and rate-1/3 settings, respectively. Constraint lengths for TBCC are 9 and 8 for rate-1/2 and rate-1/3 packet, respectively. We also show the comparison to complex modulated SPARC [16] with the same length and similar rates (0.4922 and 0.3515 to be precise). AMP decoding is adopted for SPARC, but no outer code or CRC is utilized.

Figure 5 shows the packet error rate (PER) of two rate settings. One packet corresponds to a single transmit vector



(a) Rate-1/2 packet



(b) Rate-1/3 packet

Fig. 5. PER performance comparison in the complex AWGN channel.

in HDM/SPARC, or a single codeword in polar-QPSK/TBCC-QPSK. The list size of polar SCL decoding is set to 8 and the maximum iteration number for WAVA is set to 10. For HDM, we test different values of $\overline{K}$ for the proposed K-best decoding algorithm with L2-norm minimization by setting a proper threshold and $K_{\max}$ for each SNR. As shown in Figure 5, the proposed HDM decoding algorithm greatly outperforms AMP decoder for SPARC, and its performance is on par with polar-QPSK and TBCC-QPSK in the AWGN channel. While the HDM performance is moderately worse than polar-QPSK and TBCC-QPSK with the rate-1/2 setting, HDM can slightly outperform them in the AWGN channel with the rate-1/3 setting. Normal approximations [8], [42] for both rate settings are also shown in the figures. Although $\overline{K}$ for HDM is larger than the list size of the SCL decoding, the runtime of HDM decoding is substantially faster (about $5\times$ on Intel Core i7-7700 CPU for $\overline{K} = 50$) than polar-QPSK SCL decoding (implementation in a Matlab toolbox) due to the computation-friendly parallel processing nature of the proposed K-best decoding. The runtime of TBCC-QPSK (our own implementation) is similar to that of HDM with $\overline{K} = 50$.

Next we examine the trade-off between PAPR reduction and SNR loss caused by intentional clipping on the transmitted HDM signal. Figure 6 shows the resulting PAPR and SNR loss at different clipping levels c for a power normalized

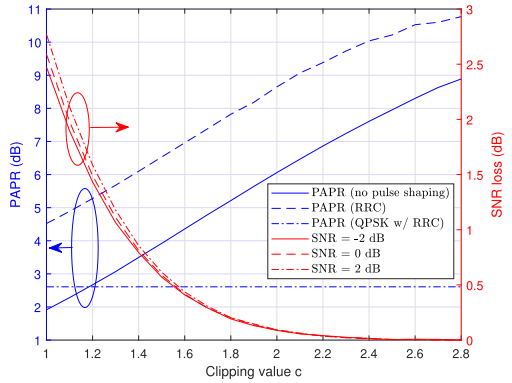


Fig. 6. PAPR and SNR loss with intentional clipping.

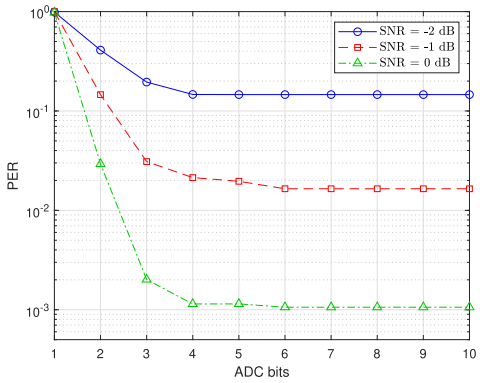


Fig. 7. PER performance with ADC quantization.

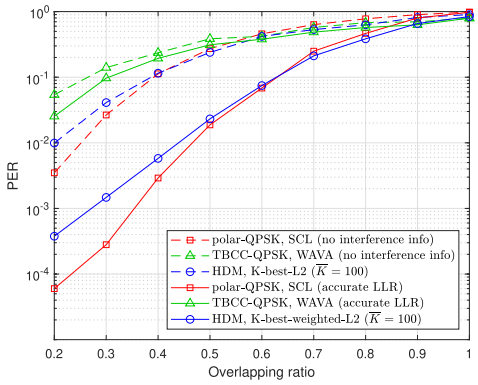
HDM vector. A root-raised-cosine (RRC) filter with roll-off factor 0.5 is used for pulse shaping. Different operating conditions regarding SNR (after clipping) are tested and the corresponding SNR losses are plotted. SNR loss is defined as the difference between original SNR before clipping and the resulting SDNR after clipping. Figure 6 shows that the PAPR can be reduced to $\leq$6.5 dB with only 0.5 dB S(D)NR degradation. It is observed that with RRC filter, the PAPR is further increased by roughly a constant 2.5 dB. Note that PSK signaling also undergoes the same/similar PAPR increase after pulse shaping. Hence the PAPR gap (in dB) in maintained the same with or without RRC filtering. It confirms that aggressive intentional clipping can be applied to HDM without significant degradation in PER performance.

We also quantify the performance degradation caused by the quantization during analog-to-digital conversion (ADC). Any value whose amplitude is outside the quantization range is saturated to the highest quantization point, and thus it can be interpreted as signal clipping in the receiver. In the simulation, the largest amplitude of the quantized signal is determined by the clipping level that results in 0.1 dB SNR degradation using the analysis in the previous PAPR tradeoff simulation. Given this saturation level, the number of ADC bits determines the other quantized levels which are uniformly spaced. Figure 7 shows the PER with different numbers of ADC bits. We observe that 4 or 5 ADC bits are sufficient for PER $\sim 10^{-3}$ at 0 dB SNR. Note that constant-envelope BPSK / QPSK schemes require a similar number of ADC bits in order to reliably operate without excessive signal distortion in a low SNR condition where PAPR is increased and set by the noise.

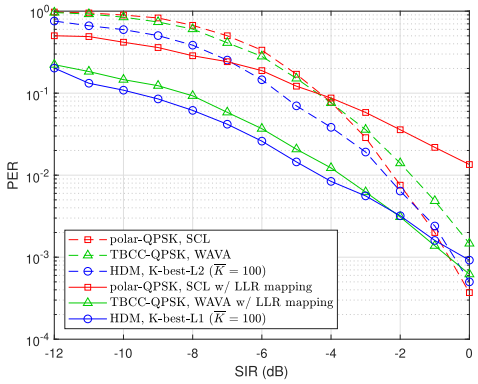
To evaluate the robustness of HDM against intensive interference in an mMTC network, we consider a narrowband mMTC system with 1kHz bandwidth that coexists with relatively wideband systems such as WiFi and BLE. Following a star network topology with grant-free pure ALOHA, an HDM packet may collide with one or more packets from other devices in the network, leading to *intra*-network interference whose timing and power information is available (estimated) at the receiver. Packets from other non-mMTC networks may also cause interference to an HDM packet. Each *inter*-network interference packet is assumed to have a fixed length of 2 ms, which is much shorter than the length of 128 ms for 1kHz-bandwidth HDM ($D = 128$) and polar/TBCC-QPSK (128-symbol) packets. We evaluate PER for intra- and inter-network interference cases separately using different HDM decoding algorithms.

Figure 8a shows the PER when a desired packet collides with another interference packet with different overlapping ratios (1 indicates complete overlap and 0.5 means one half of the packet is overlapped). The interference packet is set to have 2 times stronger power than the desired packet to simulation an interference dominated scenario. The background (interference-free) SNR is set to 1 dB, which is sufficient for all schemes to achieve PER less than 10^{-3} when the interference is absent. The timing and power information of the interference is assumed to be perfectly estimated at the receiver so it can either adopt the weighted-L2 K-best algorithm for HDM, or calculate a more accurate log-likelihood ratio (LLR) for each bit for polar codes and TBCC. Note that for QPSK packets, the resulting noise plus interference is not Gaussian distributed, thus the LLR calculation is not exact even if the interference power variance is known. As shown in the figure, HDM with the weighted-L2 K-best algorithm and polar-QPSK outperform TBCC, which turns out to be significantly more vulnerable to a packet collision. This is because of its trellis-based structure, which is vulnerable to *consecutive* corrupted symbols observed during collision. Note that for all three schemes, the performance (in dotted lines) significantly degrades when interference information is unavailable.

Figure 8b shows the PER in strong inter-network interference scenarios. The background (interference-free) SNR of the desired packet is set to 3 dB for this simulation to evaluate an interference-dominant condition. The power of interference packet follows a log-normal distribution with variance of 10 dB [3] while their mean power is set by the simulated signal-to-interference ratio (SIR), which is the ratio between average signal power of the desired packet and average interference power. Note that SIR is defined by only the part where the interference burst overlaps with the desired packet. The arrival of interference packets follows a



(a) PER performance with intra-network interference



(b) PER performance with inter-network interference

Fig. 8. PER performance in interference-heavy scenarios. Note that HDM is robust for both cases whereas TBCC-QPSK is vulnerable to intra-network interference and polar-QPSK suffers in the inter-network interference scenario.

Poisson arrival process with a mean interval of 5 ms while each interference packet is 2 ms long. Each sample in an interference packet is an i.i.d. Gaussian random variable that emulates the amplitude of OFDM signals. The desired signal has 1kHz bandwidth, thus each sample/symbol in the desired packet spans 1 ms. To increase the robustness against the outlier samples caused by strong short interference packets, HDM with the L2-norm minimization K-best algorithm sets an amplitude saturation threshold of 2 for each I and Q channel for a power-normalized HDM packet. For polar codes and TBCC, an LLR mapping method [30] designed to works with a wide range of interference-to-noise ratio (INR) between -5 and 40 dB is used to enhance the robustness to pulse-like interference. Note that HDM does not require SI(N)R information for decoding while polar/TBCC-QPSK uses the (average) SNR information for LLR computation. Using the (average) SINR for LLR computation degrades PER for polar/TBCC-QPSK since interference is short and sporadic.

Figure 8b confirms that HDM with the L1-norm minimization K-best algorithm yields the best performance. It is observed that the LLR mapping method improves polar codes very little while it can be even harmful when SIR is high. This is related to the polar decoding process, where the effects of inaccurate LLR propagates through successive cancellation, often causing unrecoverable errors. On the other hand, HDM

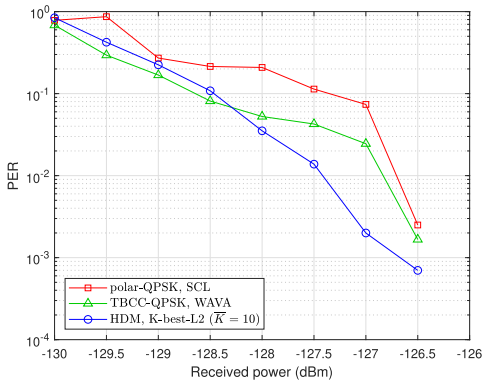


Fig. 9. PER measurement at 915MHz.

and TBCC are more robust to the sporadic outliers. Note that the gap between L1- and L2-norm minimizations for HDM reduces as SIR increases and eventually the L2-norm minimization scheme outperforms the L1-norm counterpart when interference does not dominate the channel noise any more.

From Figures 5 and 8, we have shown that HDM works reliably for all scenarios unlike polar and TBCC-based schemes which are vulnerable to some scenarios. Specifically, TBCC is relatively more vulnerable to heavy intra-network interference scenarios (Fig. 8a) whereas polar codes suffer in inter-network interference scenarios (Fig. 8b).

B. Real-World Experiments

To evaluate the performance in real-world scenarios, a wireless end-to-end system testing setup is constructed using a software define radio platform, USRP X310 [43]. Two USRPs are used as a transmitter and receiver pair for wireless communication in uncontrolled real-world channels which may corrupt the signal by noise and interference. The signal has 10kHz bandwidth and each packet contains a pre-defined preamble. The payload is modulated either by HDM or QPSK with polar/TBCC encoding for the length of 128 symbols (12.8 ms) to contain 64 information bits with 1/2 rate. The preamble is used for packet detection and channel estimation. We assume the channel is block fading with a constant amplitude and phase over one packet. We then use the estimated channel to equalize the received packet. We design the preamble to be sufficiently long (60 ms) so that the packet detection and channel estimation does not limit the decoding error performance. The USRP transmitter and receiver pair exhibits inevitable small carrier frequency offset which causes slow phase rotation of the received signal. Hence the transmitter also sends an unmodulated pilot tone on a different carrier frequency along with the signal to assist frequency offset tracking. To control the transmit power of the packet, signal attenuators are used in addition to the digital gain control feature provided by the USRP.

We first test the performance in the 915MHz ISM band, which is less crowded with fewer interference sources compared to the 2.4GHz ISM band. Figure 9 shows that HDM

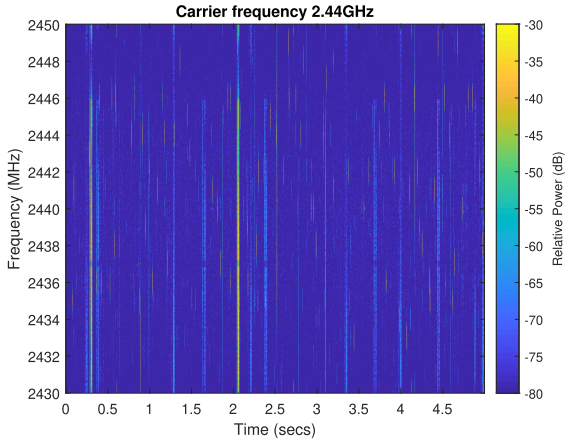


Fig. 10. Spectrogram at 2.44GHz.

outperforms polar/TBCC-QPSK even with a moderate $\overline{K} = 10$ in the real-world channel, unlike the simulated AWGN case shown in Figure 5a. This may be due to the uncontrolled interference and inaccurate SNR (consequently, inaccurate LLR) estimation, which cause more significant performance degradation to polar codes and TBCC than HDM. The presence of interference and/or other non-idealities can be observed by the offset between the expected sensitivity of -134 dBm (in ideal AWGN) and the measured sensitivity of -126.75 dBm for the PER of 10^{-3}.

Next, we test the performance in the 2.4GHz ISM band, which contains severe uncontrolled interference including WiFi and Bluetooth. Figure 10 shows an example spectrogram of the signal captured in the 2.4GHz band with 20MHz sampling rate, where wideband WiFi signals ($\geq$20MHz) and frequency hopping Bluetooth signals ($\geq$1MHz) dominate the spectrum. These interference sources are both wideband and short compared to the desired narrowband (10kHz) signal, which justifies our assumption made in the previous sections. Although not visible in the spectrogram because of the limited time resolution, there are also many very short ($\ll$1ms) interference signals, which may be short Bluetooth control packets or other wireless devices operating in the university campus network.

Since the interference in the environment is not controlled, it is not practically feasible to completely distinguish the interference from the noise. Therefore, in our experiments we define the interference any signal that has 10 dB higher power than average noise power. By definition, INR is $>$10dB for our experiment. The power distribution of the measured interference signal is shown in Figure 11, which reveals that the instantaneous INR can be higher than 30 dB in the real-world 2.4GHz channel.

Figure 12 shows the distribution of the interference duration and the interval between two closest interference signals. It is observed that most of the interference is short ($\leq$1 ms). The intervals between interference are also relatively short compared to the narrowband HDM and polar/TBCC-QPSK packets. This implies that one packet may encounter more than one interference burst with high probability. Although this

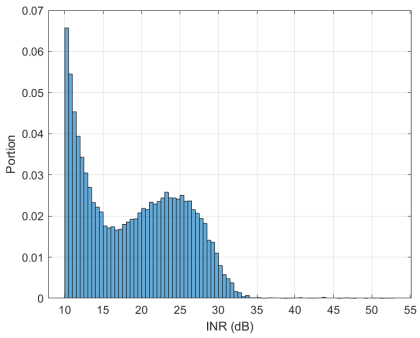


Fig. 11. Power distribution of the interference.

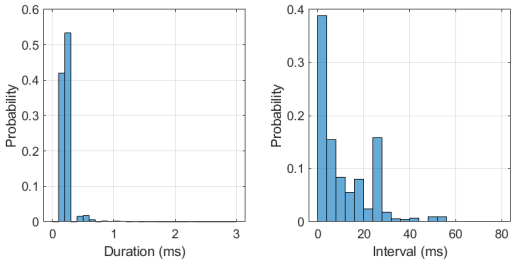


Fig. 12. Duration and interval statistics of the interference.

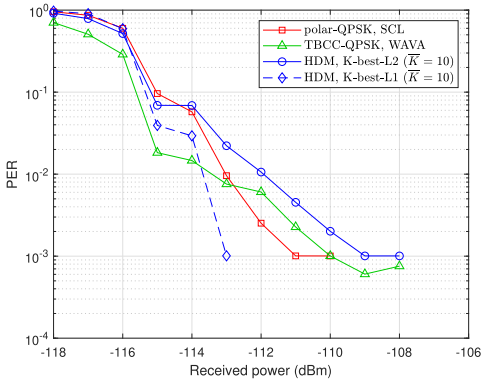


Fig. 13. PER measurement at 2.4GHz.

observation is based on our own definition of interference with INR $>$ 10dB, it is generalizable for a busy 2.44GHz band such as on-campus networks where WiFi and Bluetooth dominate the wireless traffic.

Figure 13 shows the PER measurement in the 2.4GHz ISM band. Both L2-norm and L1-norm minimization algorithms are shown for HDM with a modest $\overline{K} = 10$ setting. The LLR mapping method [30] is adopted and verified to improve the performance during the real-world experiments. Figure 13 shows that, for the same PER, the required received signal power is higher for the 2.4GHz band than that of the 915MHz band because of the stronger background interference. It is observed that HDM with L1-norm minimization K-best algorithm has the best performance for the target PER of $<10^{-3}$. For HDM, no error is observed for at least 5000 packets when the received power is ≥ -113 dBm. The results do not closely follow the simulated results in Figure 8b because of the mismatch in interference characteristics between the

uncontrolled real-world environment and our simulation setup. More burst errors are observed in this real-world experiment that cause more degradation for TBCC and HDM with the L2-norm minimization algorithm.

VII. Conclusion

In this paper, we propose hyper-dimensional modulation (HDM) specifically designed for interference-heavy mMTC networks. HDM is a special case of sparse superposition codes as a non-orthogonal modulation scheme that superimposes multiple independent vectors for concurrent transmission. The proposed decoding scheme uses a CRC-aided K-best algorithm with L2-norm minimization to achieve robust performance in the AWGN channel. Furthermore, the algorithm is extended to weighted L2- and L1-norms to combat intra- and inter-network interference, respectively. Both simulations and real-world experiments are provided to show that the proposed schemes greatly improves SPARC for short packets and HDM can outperform conventional orthogonal transmission schemes that use strong channel coding such as polar codes and TBCC. The proposed HDM is particularly advantageous in interference-heavy scenarios as a promising solution for practical mMTC networks.

References

[1] H.-S. Kim, "HDM: Hyper-dimensional modulation for robust low-power communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[2] C.-W. Hsu and H.-S. Kim, "Collision-tolerant narrowband communication using non-orthogonal modulation and multiple access," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[3] C.-W. Hsu and H.-S. Kim, "Non-orthogonal modulation for short packets in massive machine type communications," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[4] M. Series, "IMT vision-framework and overall objectives of the future development of IMT for 2020 and beyond," Int. Telecommun. Union, Geneva, Switzerland, Tech. Rep., ITU-Recommendation M.2083-0, Sep. 2016.

[5] 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.

[6] C. Bockelmann et al., "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sep. 2016.

[7] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Aug. 2016.

[8] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[9] L. Gaudio, T. Ninacs, T. Jerkovits, and G. Liva, "On the performance of short tail-biting convolutional codes for ultra-reliable communications," in *Proc. 11th Int. ITG Conf. Syst., Commun. Coding (SCC)*, Feb. 2017, pp. 1–6.

[10] M. Baldi, F. Chiaraluce, N. Maturo, G. Liva, and E. Paolini, "A hybrid decoding scheme for short non-binary LDPC codes," *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2093–2096, Dec. 2014.

[11] H. Nikopour and H. Baligh, "Sparse code multiple access," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 332–336.

[12] S. Chen, B. Ren, Q. Gao, S. Kang, S. Sun, and K. Niu, "Pattern division multiple access—A novel nonorthogonal multiple access for fifth-generation radio networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3185–3196, Apr. 2017.

[13] L. Ping, L. Liu, K. Wu, and W. K. Leung, "Interleave division multiple-access," *IEEE Trans. Wireless Commun.*, vol. 5, no. 4, pp. 938–947, Apr. 2006.

[14] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "A novel analytical framework for massive grant-free NOMA," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2436–2449, Mar. 2019.

[15] A. Joseph and A. R. Barron, "Fast sparse superposition codes have near exponential error probability for $R < C$," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 919–942, Feb. 2014.

[16] K. Hsieh and R. Venkataramanan, "Modulated sparse superposition codes for the complex AWGN channel," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4385–4404, Jul. 2021.

[17] 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, 3rd Quart., 2017.

[18] X. Xiong, K. Zheng, R. Xu, W. Xiang, and P. Chatzimisios, "Low power wide area machine-to-machine networks: Key techniques and prototype," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 64–71, Sep. 2015.

[19] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognit. Comput.*, vol. 1, no. 2, pp. 139–159, Oct. 2009.

[20] C. Rush, A. Greig, and R. Venkataramanan, "Capacity-achieving sparse superposition codes via approximate message passing decoding," *IEEE Trans. Inf. Theory*, vol. 63, no. 3, pp. 1476–1500, Mar. 2017.

[21] G. D. Forney and L.-F. Wei, "Multidimensional constellations. I. Introduction, figures of merit, and generalized cross constellations," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 6, pp. 877–892, Aug. 1989.

[22] T. Koike-Akino and V. Tarokh, "Sphere packing optimization and EXIT chart analysis for multi-dimensional QAM signaling," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[23] H. Ji, S. Park, and B. Shim, "Sparse vector coding for ultra reliable and low latency communications," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6693–6706, Oct. 2018.

[24] S. Kwon, J. Wang, and B. Shim, "Multipath matching pursuit," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2986–3001, May 2014.

[25] E. Başar, U. Aygölü, E. Panayirci, and H. V. Poor, "Orthogonal frequency division multiplexing with index modulation," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5536–5549, Nov. 2013.

[26] M. Hersche, S. Lippuner, M. Korb, L. Benini, and A. Rahimi, "Near-channel classifier: Symbiotic communication and classification in high-dimensional space," *Brain Informat.*, vol. 8, no. 1, p. 16, Dec. 2021.

[27] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[28] C. Goursaud and Y. Mo, "Random unslotted time-frequency ALOHA: Theory and application to IoT UNB networks," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.

[29] S. Popli, R. K. Jha, and S. Jain, "A survey on energy efficient narrowband Internet of Things (NBIoT): Architecture, application and challenges," *IEEE Access*, vol. 7, pp. 16739–16776, 2019.

[30] M. Mehrnoush and S. Roy, "Coexistence of WLAN network with radar: Detection and interference mitigation," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 655–667, Dec. 2017.

[31] C. Martinez, "Partial quicksort," in *Proc. 6th ACMSIAM Workshop Algorithm Eng. Exp., 1st ACM-SIAM Workshop Analytic Algorithmics Combinatorics*, 2004, pp. 224–228.

[32] *Short Block-Length Design*, document R1-1610141, 3GPP, 2016.

[33] M. Ajtai, J. Komlós, W. L. Steiger, and E. Szemerédi, "Optimal parallel selection has complexity O (log log N)," *J. Comput. Syst. Sci.*, vol. 38, no. 1, pp. 125–133, Feb. 1989.

[34] B. Li, H. Shen, and D. Tse, "Parallel decoders of polar codes," 2013, *arXiv:1309.1026*.

[35] H. S. Kim and B. Daneshrad, "Power optimized PA clipping for MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 9, pp. 2823–2828, Sep. 2011.

[36] H. Ochiai and H. Imai, "Performance of the deliberate clipping with adaptive symbol selection for strictly band-limited OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 11, pp. 2270–2277, Nov. 2000.

[37] T. Koike-Akino, K. J. Kim, M. Pajovic, and P. V. Orlik, "Universal multi-stage precoding with monomial phase rotation for full-diversity M2M transmission," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–7.

[38] *NR; Multiplexing and Channel Coding (Release 15)*, document TS 38.212, 3GPP, 2018.

[39] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

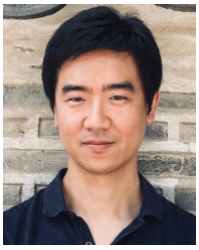
[40] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two decoding algorithms for tailbiting codes," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1658–1665, Oct. 2003.

[41] M. C. Coşkun et al., "Efficient error-correcting codes in the short blocklength regime," *Phys. Commun.*, vol. 34, pp. 66–79, Jun. 2019.

[42] W. Yang, Y. Wang, J. Soriaga, T. Ji, and K. Mukkavilli, "Coding performance modeling for short-packet communications," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 820–826.

[43] *Ettus Research X310*. Accessed: Dec. 3, 2022. [Online]. Available: https://www.ettus.com/all-products/x310-kit/

Chin-Wei Hsu received the B.S. degree in electrical engineering and the M.S. degree in communication engineering from the National Taiwan University, Taipei, Taiwan, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. His research interests include novel modulation, coding schemes, and low power design for wireless communications.



Hun-Seok Kim (Senior Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2001, and the Ph.D. degree in electrical engineering from the University of California at Los Angeles, Los Angeles, CA, USA, in 2010. He is currently an Associate Professor with the University of Michigan, Ann Arbor, MI, USA. His research interests include system analysis, novel algorithms, and VLSI architectures for low-power/high-performance wireless communications, signal processing, computer vision, and machine learning systems. He was a recipient of the DARPA Young Faculty Award in 2018 and the NSF CAREER Award in 2019. He is an Associate Editor of IEEE TRANSACTIONS ON MOBILE COMPUTING.