Asynchronous Code-Division Random Access
Using Convex Optimization

Lorne Applebaum, Waheed U. Bajwa, Marco F. Duarte, and Robert Calderbank

Abstract

Many applications in cellular systems and sensor networks involve a random subset of a large number of users asynchronously reporting activity to a base station. This paper examines the problem of multiuser detection (MUD) in random access channels for such applications. Traditional orthogonal signaling ignores the random nature of user activity in this problem and limits the total number of users to be on the order of the number of signal space dimensions. Contention-based schemes, on the other hand, suffer from delays caused by colliding transmissions and the hidden node problem. In contrast, this paper presents a novel asynchronous (non-orthogonal) code-division random access scheme along with a convex optimization-based MUD algorithm that overcomes the issues associated with orthogonal signaling and contention-based methods. Two key distinguishing features of the proposed algorithm are that it does not require knowledge of the delay or channel state information of every user and it has polynomial-time computational complexity. The main analytical contribution of this paper is the relationship between the performance of the proposed MUD algorithm and two simple metrics of the set of user codewords. The study of these metrics is then focused on two specific sets of codewords, random binary codewords and specially constructed algebraic codewords, for asynchronous random access. The ensuing analysis confirms that the proposed scheme together with either of these two codeword sets significantly outperforms the orthogonal signaling-based random access in terms of the total number of users in the system.

Index Terms

Asynchronous random access, cyclic codes, lasso, multiuser detection, non-orthogonal codes, sparse signal recovery, spread spectrum communication

This paper was presented in part at the Forty-Eighth Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, Sep./Oct. 2010. LA is with Princeton University, Princeton, NJ 08544, while WUB, MFD and RC are with Duke University, Durham, NC 27708 (Emails: lappleba@princeton.edu, {w.bajwa, marco.duarte, robert.calderbank}@duke.edu).
I. INTRODUCTION

Many applications of wireless networks require servicing a large number of users that share limited communication resources. In particular, the term random access is commonly used to describe that a random subset of users in the network communicate with a base station (BS) in an uncoordinated fashion [1]. In this paper, we study random access in large networks for the case when active users transmit single bits to the BS. This so-called “on–off” random access channel (RAC) [2] represents an abstraction that arises frequently in many wireless networks. In third-generation cellular systems, for example, control channels used for scheduling requests can be modeled as on–off RACs; in this case, users requesting permissions to send data to the BS can be thought of as transmitting 1’s and inactive users can be thought of as transmitting 0’s. Similarly, uplinks in wireless sensor networks deployed for target detection can also be modeled as on–off RACs; in this case, sensors that detect a target can be made to transmit 1’s and sensors that have nothing to report can be thought of as transmitting 0’s.

The primary objective of the BS in on–off RACs is to reliably and efficiently carry out multiuser detection (MUD), which translates into recovery of the set of active users in our case. The two biggest impediments to this goal are that (i) random access tends to be asynchronous in nature, and (ii) it is quite difficult, if not impossible, for the BS to know the channel state information (CSI) of every user. Given a fixed number of temporal signal space dimensions $N$ in the uplink, the system-design goal therefore is to simultaneously maximize the total number of users $M$ in the network and the average number of active users $k$ that the BS can reliably handle without requiring knowledge of the delays or CSIs of the individual users at the BS.

Traditional approaches to random access fall significantly short of this design objective. In random access methods based on orthogonal signaling, the $N$ signal space dimensions are orthogonally spread among the $M$ users in either time, frequency, or code [1]. While this establishes a dedicated, interference-free channel between each user and the BS, this approach ignores the random nature of user activity in RACs. Therefore, by its very structure, random access based on orthogonal signaling dictates the relationship $k \leq M \leq N$. On the other hand, contention-based random access schemes such as ALOHA and carrier sense multiple access (CSMA) do take advantage of the random user activity [3]. However, significant problems arise in these schemes when the average number of active users $k$ and/or the total

---

1The focus of this paper is on servicing a large number of users that share limited communication resources in the uplink. Limiting ourselves to on–off RACs in this case helps up isolate the key issues associated with designing arbitrary RACs involving (multiple-bit) packet transmissions in large networks.
number of users $M$ gets large \[3\]. In the case of ALOHA, collisions and retransmissions accumulate to significant delays as $k$ becomes large. In the case of CSMA, the number of potential “hidden nodes” grows as $M$ increases, resulting in unintended and unrecognized collisions in large networks.

In order to overcome the issues associated with orthogonal signaling and contention-based methods, we present in this paper a novel code-division random access (CDRA) scheme that spreads the uplink communication resources in a non-orthogonal manner among the $M$ users and leverages the random user activity to service significantly more total users than $N$. A key distinguishing feature of the proposed scheme is that it makes use of a convex optimization-based MUD algorithm that does not require knowledge of the delays or CSIs of the users at the BS. In addition, we present an efficient implementation of the proposed algorithm based on the *fast Fourier transform* (FFT) that ensures that its computational complexity at worst differs by a logarithmic factor from an *oracle-based* MUD algorithm that has perfect knowledge of the user delays. Our main analytical contribution is the relationship between the probability of error $P_{err}$ of the proposed MUD algorithm and two metrics of the set of codewords assigned to the users. We then make use of these metrics to analyze two specific sets of codewords, random binary codewords and specially constructed (deterministic) algebraic codewords, for the proposed random access scheme. Specifically, we show that both these codewords allow our scheme to successfully manage an average number of active users that is almost linear in $N$, $k \lesssim N/\log M$. More importantly, we show that the set of random codewords enable our scheme to service a number of total users that is exponential in $N$, $M \lesssim \exp(O(N^{1/3}))$, while the set of deterministic codes, which facilitate efficient codeword construction and storage, enable it to service a number of total users that is polynomial in $N$, $M \lesssim N^t$ for any reasonably sized $t \geq 2$.$^2$

It is useful at this point to also consider non-orthogonal code-division *multiple access* (CDMA), which—like our scheme—also spreads the uplink communication resources in a non-orthogonal manner among $M > N$ users \[1\]. However, despite similarities at the codeword-assignment level, there are significant differences between non-orthogonal CDMA and the work presented here. First, non-orthogonal CDMA is used for applications in which a fixed set of users continually communicate with the BS, whereas our scheme corresponds to a random subset of users in a large network communicating single bits to the BS. Second, MUD schemes for non-orthogonal CDMA require that the BS has knowledge of the individual user delays, whereas we assume—partly because of the random user activity—that user delays are unknown at the BS.

\[2\] Recall the “Big-O” notation: $f(n) = O(g(n))$ (alternatively, $f(n) \lesssim g(n)$) if $\exists c_o > 0, n_o : \forall n \geq n_o, f(n) \leq c_o g(n)$. 

January 10, 2011 DRAFT
In terms of related prior work, Fletcher et al. [2] have also recently studied the problem of MUD in on–off RACs. However, the results in [2]—while similar in spirit to the ones in here—are limited by the facts that [2]: (i) assumes perfect synchronization among the $M$ users, which is hard to guarantee in practical settings for large $M$; (ii) assumes that CSIs of the individual users are available to the BS in certain cases, which is difficult—if not impossible—to justify for the case of fading RACs; and (iii) only guarantees that the probability of error $P_{err}$ at the BS goes to zero asymptotically in $M$, which does not shed light on the scaling of $P_{err}$. More recently, [4] has also considered on–off RACs in the context of configuration in ad-hoc wireless networks, but it too makes assumptions about the channel that are similar to [2]. Finally, the work presented here also has implications in the area of sparse signal recovery and it relates to some recent work in model selection and compressed sensing. We defer a detailed discussion of these implications and relationships to Section V of the paper.

In this paper we use the following notational conventions. We use lowercase and uppercase bold-faced letters, such as $x$ and $X$, to represent vectors and matrices, respectively, while we use $(\cdot)^T$ to denote transposition of vectors and matrices. The identity matrix and the all-zeros vector are denoted by $I$ and $0$, respectively, and their dimensions are either given by context or explicitly shown in subscripts. The notation $\mathcal{N}(m,\sigma^2)$ signifies the Gaussian distribution with mean $m$ and standard deviation $\sigma$, binary$(\pm 1/\sqrt{N},I_N)$ denotes an $N$-length Rademacher distribution in which each entry independently takes value $+1/\sqrt{N}$ or $-1/\sqrt{N}$ each with probability 1/2, and $E[\cdot]$ denotes the expectation of a random variable. We use $\Pr(\cdot)$ to denote the probability of an event and $\Pr(\cdot|C)$ as the probability conditioned on an event $C$. The notation $\langle x,y \rangle$ is used to denote the inner product between vectors $x$ and $y$. Finally, $\log(\cdot)$ is taken as the natural logarithm throughout the paper.

The remainder of the paper is organized as follows. In Section II, we introduce our system model and the accompanying assumptions. In Section III, we describe our approach to MUD for asynchronous (non-orthogonal) CDRA and specify its performance in terms of two metrics of the set of user codewords. In Section IV, we specialize the results of Section III to random binary codewords and specially constructed algebraic codewords. We finally conclude in Section V by reporting results of some numerical experiments and discussing connections of our work in the area of sparse signal recovery.

II. System Model

In this section, we formalize the problem of MUD in asynchronous on–off RACs by introducing our system model and accompanying assumptions. To begin, we assume that there are a total of $M$ users in the network that communicate with the BS using waveforms of duration $T$ and (two-sided) bandwidth $W$;
in other words, the total number of temporal signal space dimensions (degrees of freedom) in the uplink are \( N = TW \). In this paper, we propose that users communicate using spread spectrum waveforms:

\[
x_i(t) = \sqrt{\mathcal{E}_i} \sum_{n=0}^{N-1} x_n^i g(t - nT_c), \quad t \in [0, T)
\]

where \( g(t) \) is a unit-energy prototype pulse (\( \int |g(t)|^2 dt = 1 \)), \( T_c \approx \frac{1}{W} \) is the chip duration, \( \mathcal{E}_i \) denotes the transmit power of the \( i \)-th user, and

\[
x_i = \begin{bmatrix} x_{i0}^i & x_{i1}^i & \ldots & x_{iN-1}^i \end{bmatrix}^T, \quad i = 1, \ldots, M
\]

is the \( N \)-length real-valued codeword of unit energy (\( \|x_i\|_2 = 1 \)) assigned to the \( i \)-th user.

In the context of on–off RACs, we assume that on average a total of \( k \) of the \( M \) users transmit 1’s at time \( t = 0 \) (without loss of generality), resulting in the following received signal at the BS

\[
y(t) = \sum_{i=1}^{M} h_i \delta_i x_i(t - \tau_i) + w(t).
\]

Here, \( h_i \in \mathbb{R} \) and \( \tau_i \in \mathbb{R}_+ \) are the channel fading coefficient and the delay associated with the \( i \)-th user, respectively, \( w(t) \) is additive white Gaussian noise (AWGN) introduced by the receiver circuitry, and \( \{\delta_i\} \) are independent 0–1 Bernoulli random variables that model the random activation of the \( M \) users in the sense that \( \Pr(\delta_i = 1) = k/M \). Finally, we assume that user transmissions undergo independent Rayleigh fading in the sense that the \( h_i \)’s are independently distributed as \( \mathcal{N}(0, \rho_i^2) \).

Next, we define the individual discrete delays \( \tau'_i \in \mathbb{Z}_+ \) as \( \tau'_i \stackrel{\text{def}}{=} \lfloor \frac{\tau_i}{T_c} \rfloor \) and define the maximum discrete delay \( \tau \in \mathbb{Z}_+ \) in the system as \( \tau \stackrel{\text{def}}{=} \max_i \tau'_i \). It is easy to see that the received signal \( y(t) \) at the BS can be sampled at the chip rate to obtain the equivalent discrete representation

\[
y = \sum_{i=1}^{M} h_i \delta_i \sqrt{\mathcal{E}_i} \tilde{x}_i + w.
\]

Here, the AWGN vector \( w \) is distributed as \( \mathcal{N}(0_{N+\tau}, I_{N+\tau}) \) (i.e., the instantaneous received signal to noise ratio (SNR) of the active users is \( \mathcal{E}_i|h_i|^2 \)) and the vectors \( \tilde{x}_i \in \mathbb{R}^{N+\tau} \) are defined as

\[
\tilde{x}_i = \begin{bmatrix} 0_{T_c}^T & x_i^T & 0_{T_c}^T \end{bmatrix}^T, \quad i = 1, \ldots, M.
\]

The assumptions we make here are that (i) the maximum delay \( \tau \) (or an upper bound on \( \tau \)) is known at the BS and (ii) each user has knowledge of the SNR at which its transmitted signal arrives at the BS

---

\(^3\)We take fading coefficients in \( \mathbb{R} \) since we are assuming real-valued codewords.

\(^4\)One of the major differences between [2], [4] and the setup in here is that it is assumed in [2], [4] that \( \max_{i,j}(\tau_i - \tau_j) < T_c \) whereas we do not make this assumption since it is nearly impossible to satisfy this condition for large-enough values of \( M \).
(in other words, the $i$-th user knows $|h_i|$). Both these assumptions are quite reasonable from a practical perspective; in particular, if one assumes that the BS transmits a beacon signal before the users start transmitting then the last assumption follows because of reciprocity between the downlink and uplink.

Our goal now is to specify a MUD algorithm for this asynchronous CDRA scheme that returns an estimate $\hat{I}$ of the set of active users $I \triangleq \{ i : \delta_i = 1 \}$ from the $(N + \tau)$-dimensional vector $y$ without knowledge of the set of delays $\{\tau_i^\prime\}$ or the set of channel coefficients $\{h_i\}$ at the BS. Note that a benchmark for any such algorithm is synchronous, orthogonal signaling-based random access, which dictates the relationship $k \leq M \leq N$. Therefore, the primary objective of our algorithm must be to successfully manage an average number of active users that is almost linear in $N$, but also service a total number of users in the uplink that is significantly larger than $N$. In addition to this primary objective, we are also interested in specifying probability of error, $P_{err} \triangleq \Pr(\hat{I} \neq I)$, and providing a low-complexity implementation of the MUD algorithm. In the next section, we propose an algorithm that explicitly takes advantage of the random user activity in the network to successfully meet all these objectives.

III. Multiuser Detection Using The Lasso

In this section, we propose a MUD algorithm for asynchronous CDRA that is based on the mixed-norm convex optimization program known as the lasso [5]. The lasso was first proposed in the statistics literature for linear regression in underdetermined settings. In [2], the lasso has been suggested as a potential method for MUD in synchronous on–off RACs. However, extending the ideas of [2] to the asynchronous case using the standard lasso formulation seems very difficult. In contrast, while the MUD algorithm proposed in this section is based on the lasso, we present a rather nonconventional usage of the lasso that is specific to the problem at hand. One of our major contributions indeed is establishing that this formulation is guaranteed to yield successful MUD with high probability. The fact that further differentiates our work from [2] is that we relate the performance of the proposed MUD algorithm to two simple metrics of the set of user codewords, which enables us to construct specialized codewords for different applications. The analysis carried out in this regard might also be of independent interest to researchers working on configuration (neighbor discovery) in ad-hoc wireless networks and sensor networks. These results also have implications in the area of sparse signal recovery, as noted in Section V-B.
A. Main Result

In order to make use of the lasso for MUD in asynchronous on-off RACs, we first rewrite (4) as

\[ y = \begin{bmatrix} \tilde{x}_1 & \tilde{x}_2 & \ldots & \tilde{x}_M \end{bmatrix} \tilde{\beta} + w \]  

(6)

where the \( i \)-th entry of the vector \( \tilde{\beta} \in \mathbb{R}^M \) is described as \( \tilde{\beta}_i \overset{def}{=} h_i \delta_i \sqrt{\mathcal{E}_i} \). While (6) appears superficially similar to the standard lasso formulation, we cannot use the lasso to obtain an estimate of the set of active users \( \mathcal{I} \) from (6) since the \( (N + \tau) \times M \) matrix \( \tilde{X} \) in (6) is unknown due to the asynchronous nature of the problem. In order to overcome this obstacle, we first define \( (N + \tau) \times (\tau + 1) \) Toeplitz matrices \( X_i \) as follows

\[ X_i = \begin{bmatrix} x_i & 0_{\tau} & \cdots & 0_{\tau} \\ \vdots & \ddots & \ddots & \vdots \\ 0_{\tau} & \cdots & x_i & \vdots \end{bmatrix}, \quad i = 1, \ldots, M \]  

(7)

and observe that we can equivalently write (6) in the form

\[ y = \begin{bmatrix} X_1 & X_2 & \ldots & X_M \end{bmatrix} \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_M^T \end{bmatrix} + w \]  

(8)

where \( X \) is now a \( (N + \tau) \times M(\tau + 1) \) known matrix, which we term the expanded codebook. The vector \( \beta \in \mathbb{R}^{M(\tau + 1)} \) is a concatenation of \( M \) vectors, each of length \( (\tau + 1) \), whose entries are given by \( \beta_{i,j} = \tilde{\beta}_i 1_{(\tau'_i = j)} \), \( i = 1, \ldots, M, \quad j = 1, \ldots, \tau + 1 \). We make use of this notation to describe the proposed lasso-based MUD algorithm for asynchronous CDRA in Algorithm 1.

We next state the main result of this section, which bounds the probability of error of Algorithm 1.

Here we present MUD guarantees for arbitrary codebooks, parametrized by two metrics of the expanded codebook \( X \). The first is the worst-case coherence of the expanded codebook, defined by

\[ \mu(X) \overset{def}{=} \max_{(i,j) \neq (i', j')} \left| \langle x_{i,j}, x_{i',j'} \rangle \right| \]  

(9)

where \( x_{i,j} \) denotes the \( j \)-th column of the Toeplitz matrix \( X_i \). In words, the worst-case coherence is the largest inner product between any two codewords with arbitrary shifts. The second metric is the spectral norm of the expanded codebook: \( \|X\|_2 \overset{def}{=} \sqrt{\lambda_{\max}(X^T X)} \).

5Algorithm 1 acts as a hybrid between the standard lasso and the group lasso [6]. Specifically, it is clear from the problem formulation that the group lasso is ill-suited for the specified MUD problem since each of the sub-vectors \( \{\beta_i\} \) in (8) has at most one non-zero entry. On the other hand, we are only interested in detecting the active users and need not estimate their delays; hence, the group nature of the detection criterion in the definition of \( \tilde{I} \).
Algorithm 1: Multiuser Detection in Asynchronous On–Off Random Access Channels Using the Lasso

**Inputs**
1) The chip-rate sampled vector \( y \)
2) Set of \( N \)-dimensional codewords \( \{ x_i \}_{i=1}^{M} \)
3) Maximum discrete delay \( \tau \) in the uplink
4) A regularization parameter \( \lambda \) for the lasso

**Construct** the expanded codebook \( X \) described in (8) using \( \{ x_i \} \) and \( \tau \)

\[
\hat{\beta} \leftarrow \arg \min_{b \in \mathbb{R}^{M(M+1)}} \frac{1}{2} \| y - Xb \|_2^2 + \lambda \| b \|_1 \quad \text{(LASSO)}
\]

\[\hat{\mathcal{I}} \leftarrow \{ i : \| \hat{\beta}_i \|_0 > 0 \}\]

**Return** \( \hat{\mathcal{I}} \) as an estimate of the set of active users \( \mathcal{I} \)

**Theorem 1:** Suppose that users in the network become active according to independent and identically distributed (iid) Bernoulli random variables such that \( \Pr(\delta_i = 1) = k/M \) and the users have transmit powers satisfying

\[
E_i > \frac{128 \log (M \sqrt{\tau + 1})}{|h_i|^2}, \quad i \in \mathcal{I}.
\]

Then, with \( \lambda = 2\sqrt{2 \log(M \sqrt{\tau + 1})} \), Algorithm 1 successfully carries out multiuser detection with

\[
P_{\text{err}} \leq \left( \pi M \log(M \sqrt{\tau + 1}) \right)^{-1} + 5(M(\tau + 1))^{-2 \log 2} + 3M^{-2 \log 2}
\]

when

\[
M \leq \exp((c(\tau + 1)\mu(X))^{-1}) \quad \text{and} \quad \frac{M}{c \log(M(\tau + 1))}\|X\|_2^2.
\]

Here, the constant \( c > 0 \) is independent of the problem parameters.

The proof of this theorem is provided in Appendix A. From (10) we see that to accommodate a large number of total users \( M \), we need codewords that result in an expanded codebook with a small worst-case coherence \( \mu(X) \). Similarly, (11) shows that codewords that result in small spectral norm \( \|X\|_2 \) allow the value of \( k \) to be large. We will specialize this theorem to random codewords in Section IV-A and a deterministic codeword construction in Section IV-B.
TABLE I
AVERAGE RECOVERY TIMES IN MATLAB FOR $N = 1023$, $M = 3072$, AND $k = 50$

<table>
<thead>
<tr>
<th>Maximum delay $\tau$</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard SpaRSA</td>
<td>21.2s</td>
<td>61.6s</td>
<td>96.2s</td>
<td>142.6s</td>
<td>173.0s</td>
</tr>
<tr>
<td>FFT Augmented SpaRSA</td>
<td>54.4s</td>
<td>53.4s</td>
<td>84.3s</td>
<td>98.0s</td>
<td>78.2s</td>
</tr>
</tbody>
</table>

B. Computational Complexity

Theorem 1 characterizes the performance of Algorithm 1 for MUD in asynchronous on–off RACs but fails to shed any light on its computational complexity. However, the lasso is a well-studied program in the statistics literature and—thanks to its convex nature—there exist a number of extremely fast (polynomial-time) implementations of the optimization program specified in (LASSO); see, e.g., [7].

In this regard, computational complexity of the implementations of (LASSO) such as SpaRSA [7] is determined—to a large extent—by the complexity of the matrix–vector multiplications $Xb$ and $X^T y$. It therefore seems that Algorithm 1 increases the computational complexity of the matrix–vector multiplications from $O(NM)$, corresponding to the case of perfectly-known user delays [cf. (6)], to $O(NM(\tau+1))$. This observation, however, ignores the fact that $X$ in (8) has a Toeplitz-block structure. Specifically, if we write $b \in \mathbb{R}^{M(\tau+1)}$ as $b = \begin{bmatrix} b_1^T & \ldots & b_M^T \end{bmatrix}^T$ then it follows from elementary signal processing that

$$Xb = \sum_{i=1}^{M} \mathcal{F}_{N+\tau}^{-1} \left( \mathcal{F}_{N+\tau}(x_i) \odot \mathcal{F}_{N+\tau}(b_i) \right)$$

(12)

where $\mathcal{F}_n(\cdot)$ and $\mathcal{F}_n^{-1}(\cdot)$ denote the FFT implementation of the $n$-point discrete Fourier transform (DFT) and the $n$-point inverse DFT of a sequence, respectively, while $\odot$ denotes pointwise multiplication. Similarly, if we use $(\cdot)[n_1 : n_2]$ to denote the $n_1$-th to $n_2$-th elements of a vector and $(\cdot)^-$ to denote the time-reversed version of a vector, then it follows from routine calculations that $\forall \ i = 1, \ldots, M$, we have

$$X^T y[i(\tau + 1) - \tau : i(\tau + 1)] = \mathcal{F}_{2N+\tau-1}^{-1} \left( \mathcal{F}_{2N+\tau-1}(x_i^-) \odot \mathcal{F}_{2N+\tau-1}(y) \right) [N : N + \tau].$$

It therefore follows from the complexity of the FFT that the matrix–vector multiplications $Xb$ and $X^T y$ in Algorithm 1 can in fact be carried out using only $O(NM \log(N + \tau))$ operations as opposed to $O(NM(\tau + 1))$ operations. This suggests that the computational complexity of Algorithm 1 at worst differs by a factor of $\log(N + \tau)$ from an oracle-based scheme that has perfect knowledge of $\tilde{X}$.

This conclusion is also justified numerically from the results of several numerical experiments reported in Section V. Table I shows typical computation times of Algorithm 1 in Matlab for various values
of \( \tau \). The standard SpaRSA recovery is faster at low values of \( \tau \) due to Matlab’s optimized matrix multiplications. However, for \( \tau \geq 100 \), the advantage of the FFT-based implementation becomes apparent. The non-monotonicity of recovery times in the FFT augmented numerical experiments is due to the complex interaction between padding in Matlab’s FFT implementation, numerical accuracy, and additional SpaRSA iterations, a detailed discussion of which is beyond the scope of this paper.

IV. CODEWORDS FOR MULTIUSER DETECTION

In this section we consider two sets of user codewords for asynchronous CDRA using Algorithm\[1]\ The first is a random construction with normalized iid \( \pm 1 \) user codewords and the second is a deterministic construction based on cyclic codes. The deterministic construction has the advantage that user codewords can be more efficiently stored and generated. However, the random construction is more flexible with regard to codeword length and number of codewords available. Using Theorem\[1]\ we show that both sets of codewords allow the recovery of \( \mathcal{I} \) when \( k \lesssim N/\log M \). Further, the random codewords allow \( M \) to be exponential in \( N \) while the deterministic codewords allow \( M \) to be a power of \( N \).

A. Random Rademacher Codewords: Guarantees

Communication theory often uses random codewords for optimality. Further, random measurement matrices are frequently used in sparse signal recovery. These examples inspire us to analyze randomly generated codewords for MUD in RACs. However, we cannot directly apply results from either field. First, we are interested in estimating set \( \mathcal{I} \) rather than recovering a signal. Second, the structure induced in \( X \) due to unknown delays is novel. Though examples from communication and sparse signal theory motivate our use of random codewords, additional theory is required for MUD in RACs. We provide analysis on a set of randomly generated codewords here.

We assign each user a codeword of length \( N \) that is independently generated from a binary\((\pm 1/\sqrt{N}, I_N)\) distribution. We seek to quantify \( \mu(X) \) and \( \|X\|_2 \) of the expanded codebook to specialize Theorem\[1]\ to these random codewords. This is accomplished in the following lemmas.

**Lemma 1:** Given any fixed \( \varsigma > 0 \), the expanded codebook \( X \) of random Rademacher codewords satisfies \( \mu(X) \leq \varsigma \) with probability exceeding \( 1 - 2M^2(\tau + 1)^2 e^{-\frac{N\varsigma^2}{4}} \).

**Proof:** The proof of this lemma is a consequence of the bound on the worst-case coherence \( \mu \) of random Toeplitz matrices \[8, Theorem 3.5\] and the Hoeffding inequality \[9\]. Specifically, we can write

\[
\mu(X) = \max \left\{ \max_{j \neq j'} \left| \langle x_{i,j}, x_{i,j'} \rangle \right|, \max_{t \neq t'} \left| \langle x_{t,j}, x_{t',j'} \rangle \right| \right\}.
\]
Further, the proof of Theorem 3.5 in [8] implies that \(|\langle x_{i,j}, x_{i,j'}\rangle| \leq \varsigma\) with probability exceeding \(1 - 4e^{-\frac{N\varsigma^2}{4}}\) for any \(j \neq j'\). Finally, since the product of two independent binary random variables is again a binary random variable, it can also be shown using the Hoeffding inequality that \(|\langle x_{i,j}, x_{i,j'}\rangle| \leq \varsigma\) with probability exceeding \(1 - 2e^{-\frac{N\varsigma^2}{4}}\) for any \(i \neq i'\). It therefore follows from the union bound that \(\mu(X) \leq \varsigma\) with probability exceeding \(1 - 2M^2(\tau + 1)^2e^{-\frac{N\varsigma^2}{4}}\).

Lemma 2: The spectral norm of the expanded codebook \(X\) of random Rademacher codewords satisfies

\[
\|X\|_2 \leq 26\sqrt{\tau + 1}\left(1 + \sqrt{\frac{M}{N}}\right)
\]

with probability exceeding \(1 - e^{-\frac{\sqrt{MN}}{8}}\).

Proof: We first recall that the spectral norm is invariant under column-interchange operations. Now define \(\Phi \stackrel{def}{=} [x_1 \ldots x_M]\) and \(\Psi \stackrel{def}{=} [\Phi_0 \Phi_1 \ldots \Phi_{\tau}],\) where each block \(\Phi_i\) is an \((N + \tau) \times M\) matrix that is constructed by prepending and appending \(\Phi\) with \(i\) rows and \((\tau - i)\) rows of all zeros, respectively. It is then easy to see that \(\|X\|_2 = \|\Psi\|_2\) and \(\|\Phi_0\|_2 = \cdots = \|\Phi_\tau\|_2 = \|\Phi\|_2\). Further, we can write for any \(M(\tau + 1)\)-dimensional vector \(z = [z_0^T \ z_1^T \ldots z_{\tau}^T]^T\)

\[
\frac{\|\Psi z\|_2}{\|z\|_2} \overset{(a)}{\leq} \sum_{i=0}^{\tau} \frac{\|\Phi_i z_i\|_2}{\|z_i\|_2} \leq \frac{\|\Phi\|_2}{\|z\|_2} \sum_{i=0}^{\tau} \|z_i\|_2 \overset{(b)}{\leq} \sqrt{\tau + 1}\frac{\|\Phi\|_2}{\|z\|_2} = \sqrt{\tau + 1}\frac{\|\Phi\|_2}{\|z\|_2}
\]

where \((a)\) follows from the definition of \(\Psi\) and the triangle inequality, while \((b)\) follows from the Cauchy–Schwarz inequality. It therefore follows from the previous discussion and (14) that \(\|X\|_2 \leq \sqrt{\tau + 1}\|\Phi\|_2\).

In order to complete the proof, notice that \(\Phi\) is an \(N \times M\) random matrix whose entries are independently distributed as \(\text{binary}(\pm1/\sqrt{N})\). It can therefore be established, similar to [10, Proposition 2.4], that \(\|\Phi\|_2 \leq 26\left(1 + \sqrt{\frac{M}{N}}\right)\) with probability exceeding \(1 - e^{-\frac{\sqrt{MN}}{8}}\).

We now want to apply these lemmas to specialize Theorem 1 to the expanded codebook matrix of random codewords. We begin by noting that with \(\varsigma\) of Lemma 1 chosen appropriately, the event

\[
G_1 = \left\{ \mu(X) \leq \sqrt{\frac{12\log \left(M(\tau + 1)\right)}{N}} \right\}
\]

holds with probability exceeding \(1 - 2(M(\tau + 1))^{-1}\). Further, Lemma 2 implies that the event

\[
G_2 = \left\{ \|X\|_2 \leq 52\sqrt{\frac{M(\tau + 1)}{N}} \right\}
\]

holds with probability exceeding \(1 - e^{-\frac{\sqrt{MN}}{8}}\).
Since we assume that the random codewords are assigned independently of the set of active users \( \mathcal{I} \), we can substitute (15) and (16) into (10) and (11) while adding the failure probabilities \( \Pr(G_1^c) \) and \( \Pr(G_2^c) \) with the probability of error in Theorem 1 via the union bound. This results in the following theorem.

**Theorem 2:** Suppose that the \( M \) codewords \( \{x_i \in \mathbb{R}^N\}_{i=1}^M \) are drawn independently from a binary \((\pm 1/\sqrt{N}, I_N)\) distribution. Further, let \( \lambda \) and \( E_i \) satisfy the conditions in Theorem 1. Then Algorithm 1 successfully carries out multiuser detection with \( P_{err} \leq \left( \pi M \log (M \sqrt{\tau + 1}) \right)^{-1} + 5(M(\tau + 1))^{-2} \log^2 + 2(M(\tau + 1))^{-1} + e^{-\frac{\sqrt{NM}}{8}} \) if

\[
M \leq \exp \left( c_{1,r}(\tau + 1)^{-2/3} N^{1/3} \right) \quad \text{and} \quad (17)
\]

\[
k \leq \frac{c_{2,r}N}{(\tau + 1) \log (M(\tau + 1))}. \quad (18)
\]

Here, the constants \( c_{1,r}, c_{2,r} > 0 \) are independent of the problem parameters.

### B. Deterministic Codewords: Construction and Guarantees

Though random codewords allow our proposed scheme to service a large number of users (with respect to \( N \)), deterministic codewords can have significant advantages. In particular, they tend to be much easier to generate and store. We will consider one such codeword construction in this section.

Our deterministic construction uses codewords derived from algebraic error correcting codes. In particular, we consider a cyclic code for which the codebook is closed under circular shifts of codewords. We use a cyclic code since we use cyclic shifts as approximations of delayed user codewords. As such, the full cyclic code is closely related to the expanded codebook matrix \( X \) which contains the delayed user codewords. To construct this relationship, we will select a subset of our cyclic code to assign to users. In order to remove ambiguity when discussing both the full cyclic code and the subset, we will call the complete cyclic code the **ambient code** while the subset assigned to users will be called the **user codebook** (i.e., the user codebook is the set \( \{x_i\} \)).

Our construction is parametrized by two positive integers, \( m \) and \( 2 \leq t < m/2 \). We will operate in the Galois finite field of size \( 2^m \), which we denote as \( \text{GF}(2^m) \). Our code is constructed via the trace function \( \text{Tr} : \text{GF}(2^m) \to \text{GF}(2) \) ([11] Ch4.8) defined by

\[
\text{Tr}(a) = a + a^2 + \cdots + a^{2^m} = \sum_{j=0}^{m-1} a^{2^j}.
\]
Taking $z$ as a primitive element of $\text{GF}(2^m)$ we define the $j^{th}$ element of a codeword in the ambient code as

$$C^j_\alpha = \frac{1}{\sqrt{2^m - 1}}(-1)^{\text{Tr}[\alpha_0 z^j + \sum_{i=1}^t \alpha_i z^{(2^i+1)}]}, \quad j = 0, 1, 2, \ldots, 2^m - 2$$  \hspace{1cm} (19)

where the vector $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_t]$ with $t + 1$ elements in $\text{GF}(2^m)$ indexes the codeword.

Since $z$ is primitive, $\{z^j : j = 0, 1, \ldots, 2^m - 2\}$ is simply the set of all non-zero elements in $\text{GF}(2^m)$, which we denote $\text{GF}(2^m)^*$. Thus, we can equivalently enumerate the elements by $x \in \text{GF}(2^m)^*$ as

$$C^j_\alpha(x) = \frac{1}{\sqrt{2^m - 1}}(-1)^{\text{Tr}[\alpha_0 x + \sum_{i=1}^t \alpha_i x^{2^i+1}]}, \quad x \in \text{GF}(2^m)^*. \hspace{1cm} (20)$$

The above construction produces codewords of length $2^m - 1$ and, since each of $\alpha_i, i = 0, \ldots, t$ can be any value in $\text{GF}(2^m)$, there are $2^{m(t+1)}$ codewords in the ambient code.

We will be using a subset of the ambient code as the user codebook. In selecting the subset we will require two conditions. The first condition restricts us to a subset where no codeword in the subset is a cyclic shift of another. Such a restriction is necessary since, in bounding $\mu(X)$, we will link cyclic shifts with different user delays. We will call such a set a cyclic restricted subset.

Consider that, under the element enumeration of (19), a codeword and its cyclic shift by $T$ elements are related as $C_{\alpha}^{j+T} = C_{\alpha'}^j$, where $\alpha = [\alpha_0, \ldots, \alpha_t]$ and $\alpha' = [z^T \alpha_0, z^{3T} \alpha_1, \ldots, z^{(2^t+1)T} \alpha_t]$. If, for example, we required that all codewords in our subset had $\alpha_0 = c$ for $c \in \text{GF}(2^m)^*$, no codewords would be the shift of another. Explicitly, the codewords indexed by $\{\alpha \in \text{GF}(2^m)^t : \alpha_0 = c\}$ form a cyclic restricted subset for any $c \neq 0 \in \text{GF}(2^m)$. Since we need only restrict a single entry in the vector $\alpha$, enumerating over the remaining entries allows us to have a cyclic restricted subset of size $2^{mt}$ from the full $2^{m(t+1)}$ ambient code. We may choose to use a smaller set, affording flexibility in choosing the value of $M$, and the set would remain a cyclic restrict subset.

The second condition on selecting the user codebook as a subset of the ambient code is used to ensure we can appropriately bound $\|X\|_2$. As above, our condition will be on the set of vectors $\alpha$ which enumerate the codewords in the user codebook. To describe the condition we first define a wildcard index of the user codebook. We call $w$ a wildcard index if the element $\alpha_w$ of the vector $\alpha$ is unrestricted in the following sense. If $[\alpha_0, \ldots, \alpha_w, \ldots, \alpha_t]$ indexes a user codeword then, fixing all other elements, $[\alpha_0, \ldots, c, \ldots, \alpha_t]$ is also a user codeword for all $c \in \text{GF}(2^m)$. We require that the set of vectors $\alpha$ enumerating the user codebook contain a wildcard index $w$ such that $2^w + 1$ does not divide $2^m - 1$ (denoted $2^w + 1 \nmid 2^m - 1$). A consequence of requiring the existence of a wildcard index is that the user codebook must be a multiple of $2^m$ in size.
To summarize our construction, we assign to users codewords of the form \((19)\). We require that the user codebook satisfy two conditions. The first is that it forms a cyclic restricted subset of the ambient code of all possible codewords. The second is that the set of vectors \(\alpha\) enumerating the user codewords contain a wildcard index \(w\) with \(2^w + 1 \nmid 2^m - 1\). This construction allows us to have \(N = 2^m - 1\) while \(M\) may be a multiple of \(2^m\) up to \(2^{mt}\).

We now seek to apply Theorem 1 to this construction which requires us to bound the two metrics \(\mu(X)\) and \(\|X\|_2\). We consider each of the two metrics in turn. The metric \(\mu(X)\) bounds inner products between any two shifted codewords in the user codebook. As discussed earlier, we will be relating the set of shifted codewords to the ambient code. As a result, our first goal is to bound the inner product of any two codewords in the ambient code. The bound can be obtained easily by exploiting properties of the ambient code. By the linearity of the trace [11, p.116], we find that the element-wise product of two codewords satisfies \(C^j_{\alpha}C^j_{\alpha'} = C^j_{\alpha + \alpha'}\). As a result, the inner product between two non-identical codewords is simply the sum of the entries of a different codeword. This leads to an equivalent goal of bounding the sum of an arbitrary non-trivial codeword. That is, allowing \(\{\alpha_i \in \text{GF}(2^m)\}_{i=0}^t\) to be arbitrary with \(\alpha \neq 0\), we attempt to bound the sum

\[
S = \sum_{x \in \text{GF}(2^m)^*} (-1)^{\text{Tr}\left[\alpha_0 x + \sum_{i=1}^t \alpha_i x^{2^i + 1}\right]}
\]

\((21)\)

**Lemma 3:** The sum given in \((21)\) satisfies \(|S| \leq 2^m + t + 1/2\) for any codeword.

This lemma is proved in Appendix \(B\). We leverage this result to provide a bound on \(\mu(X)\).

**Lemma 4:** Let the user codebook \(\{x_i\}\) be a cyclic restricted subset of the ambient code defined by \((19)\). Then the worst-case coherence is bounded by

\[
\mu(X) \leq \frac{2^{t + 1/2} + m/2 + \tau}{2^m - 1}.
\]

**Proof:** We are interested in bounding \(\langle x_{i,j}, x_{i',j'} \rangle\). We will do so by relating this inner product to one in the ambient code. Using the fact that \(x_{i',j'}\) is of the form \((5)\), we can replace vectors of zeros above and below the codeword with shifted copies of \(x_{i'}\) in order to make the vector periodic with period \(N\) on its \(N + \tau\) length. We call this the *periodic extension* of \(x_{i',j'}\). Let \(\hat{x}_{i',j'}\) be the periodic extension of \(x_{i',j'}\) and \(\Delta\) be its difference from the original such that \(x_{i',j'} = \hat{x}_{i',j'} + \Delta\). By the triangle inequality,

\[
|\langle x_{i,j}, x_{i',j'} \rangle| \leq |\langle x_{i,j}, \hat{x}_{i',j'} \rangle| + |\langle x_{i,j}, \Delta \rangle|.
\]

\(6\)This is a simple reformulation of the fact that the non-exponentiated code is linear.
The periodic extension has converted the shift \(j'\) into a cyclic shift on the support of \(x_{i,j}\). Furthermore, since the two unshifted codewords \(x_i\) and \(x_{i'}\) come from a cyclic restricted subset, they are guaranteed to be different on the support. Thus, for the first term we use the bound of Lemma 3 divided by \(2^m - 1\), thereby accounting for the normalization of the users’ codewords. The second term is bounded by the fact that the support of \(\Delta\) overlaps with that of \(x_{i,j}\) with at most \(\tau\) elements of value \(\pm 1/\sqrt{2^m - 1}\).

Having bounded \(\mu(X)\), we now turn to the second metric \(\|X\|_2\) which we bound using the following lemma.

**Lemma 5:** Let the user codebook of cardinality \(M\) selected from the ambient code contain a wildcard index \(w\) such that \(2^w + 1 \nmid 2^m - 1\). Then spectral norm of the expanded codebook \(X\) is bounded by \(\|X\|_2 \leq \sqrt{\frac{M}{2^m - 1}}(\tau + 1)\).

**Proof:** We begin the proof similarly to Lemma 2. Let \(\Phi\) be an \(N \times M\) matrix of the user codewords and recall from Lemma 2 that \(\|X\|_2 \leq \sqrt{\tau + 1}\|\Phi\|_2\). We will now show that the rows of \(\Phi\) are orthogonal, such that \(\Phi \Phi^T = \frac{M}{2^m - 1}I\), which is sufficient for the proof since \(\lambda_{\max}(\Phi \Phi^T) = \lambda_{\max}(\Phi^T \Phi)\).

Using (20), the inner product between two rows, indexed by \(x\) and \(y\) in \(GF(2^m)\), is the following sum over the vectors \(\alpha\) indexing the user codebook:

\[
\frac{1}{2^m - 1} \sum_{\alpha} C^{(x)}(\alpha) C^{(y)}(\alpha) = \frac{1}{2^m - 1} \sum_{\alpha} (-1)^{\text{Tr} \left[ \alpha_0(x+y) + \sum_{i=1}^t \alpha_i(x^{2^i+1} + y^{2^i+1}) \right]}
\]

\[
= \frac{1}{2^m - 1} \sum_{\alpha \in GF(2^m)} (-1)^{\text{Tr} \left[ \alpha_0(x^{2^w+1} + y^{2^w+1}) \right]} \sum_{\alpha \setminus \{\alpha_w\}} (-1)^{\text{Tr} \left[ \alpha_0(x+y) + \sum_{i=1}^t \alpha_i(x^{2^i+1} + y^{2^i+1}) \right]}
\]

where in (22b) we have separated the wildcard element \(\alpha_w\), which takes every value in \(GF(2^m)\), into a separate sum. For all \(\alpha \in GF(2^m)\) we have \(a + a = 0\). Thus, from the sum in (22a), when \(x = y\) each term takes unit value and their sum equals \(M\), the number of \(\alpha\). In the case when \(x \neq y\), we examine (22b). By our wildcard index condition we are guaranteed that \(x^{2^w+1} \neq y^{2^w+1}\) so that \(\alpha_w\) has a non-zero coefficient in the trace. By Proposition 3(a) in Appendix B precisely half the terms of the sum over \(\alpha_w\) are \(-1\) and thus the whole sum evaluates to 0. Therefore, the inner product between the two rows evaluates to \(M/(2^m - 1)\) when \(x = y\) and 0 otherwise. This completes the proof of the lemma.

Having bound both \(\mu(X)\) and \(\|X\|_2\), we are able to apply Theorem 1 to this deterministic construction and give the following recovery guarantee.

**Theorem 3:** Suppose that the \(M\) codewords \(\{x_i \in \mathbb{R}^N\}_{i=1}^M\) are chosen from the code defined by (19) such that they form a cyclic restricted subset and contain a wildcard index \(w\) with \(2^w + 1 \nmid 2^m - 1\). Further, let \(\lambda\) and \(\{E_i\}\) satisfy the conditions in Theorem 1. Then Algorithm 1 successfully carries out...
multiuser detection with $P_{err} \leq \frac{2}{2\pi M \log(M\sqrt{\tau + 1})} + 5(M(\tau + 1))^{-2\log 2} + 3M^{-2\log 2}$ if

$$M \leq \exp\left(\frac{c_1d(2^m-1)}{\tau+1}\right)$$

and

$$k \leq \frac{c_2d(2^m-1)}{(\tau+1)\log(M(\tau+1))}. \quad (23)$$

Here, the constants $c_{1,d}, c_{2,d} > 0$ are independent of the problem parameters.

Though (23) appears to allow an exponential number of users $M$, we are restricted by our construction which allows for at most $2^{mt}$ codewords to be assigned to users. Thus, in comparing our deterministic construction of codewords to the random generation in Section IV-A we find that the deterministic codewords have advantages in storage and generation while randomly generated codes have the advantages that $N$ is arbitrary and that $M$ can be exponential in $N$.

V. NUMERICAL RESULTS AND DISCUSSION

A. Monte Carlo Experiments

To verify and illustrate the results presented in this paper for MUD in asynchronous RACs, we make use of Monte Carlo trials. Our numerical experiments assume a total of $M = 3072$ users communicating to the BS using codewords of length $N = 1023$. We report the MUD results for both the random codewords of Section IV-A and the deterministic construction of Section IV-B. For the deterministic construction, a code generated with $m = 10$ and $t = 2$ is used with a subset of the ambient code of size $M = 3(2^m)$ assigned to users. Random user activity is generated using independent $0$–$1$ Bernoulli random variables $\{\delta_i\}$ such that $\Pr(\delta_i = 1) = k/M$ for a given $k$. Further, for a given maximum delay $\tau$, the individual user delays $\{\tau_i\}$ are generated once at random for each experiment and then fixed for the remainder of the experiment, keeping with the fact that our results are valid for arbitrary user shifts. The implementation of Algorithm I uses the SpaRSA package [7] in order to solve (LASSO) and includes the modifications described in Section III-B for speeding up the matrix–vector multiplications $Xb$ and $X^Ty$.

In all the numerical plots, results for random codewords are displayed using solid lines while those for deterministic codewords are displayed using dashed lines.

The numerical experiments correspond to the ability of Algorithm I to correctly recover the active user set $I$ for varying values of the number of active users $k$ and maximum delay $\tau$. The results of these experiments are reported in Figure I, which shows that when $k$ is below a certain threshold, $I$ is successfully recovered in the vast majority of Monte Carlo trials. Beyond the threshold of $k \approx 50$, the fraction of Monte Carlo trials in error quickly approaches one. The other thing to note from Figure I...
is that codewords generated as described in Section IV-B perform nearly identically in performance to those randomly generated.

Here, it is worthwhile to highlight an important aspect of Theorems 1-3 which illuminates a difference seen between the theory and Figure 1. The recovery guarantee provided in the theory applies to any arbitrary set of user shifts \( \{\tau_i\} \). This is desirable since any fixed probability distribution on the user shifts
Fig. 3. Normalized per user error as a function of the received user power with $\tau = 50$

would have restricted the applicability of our results to certain classes of networks. However, such a worst-case analysis is difficult to reproduce in numerical experiments since this would require generating all $(\tau + 1)^k$ possible combinations of $\{\tau_i\}$ in each Monte Carlo trial. Instead, the numerical experiments reported in Figure 1 correspond to a single set of shifts for each value of $k$. This difference between the theory and our experiments manifests itself in the relationship between the threshold of $k$ beyond which recovery of $\mathcal{I}$ fails and the maximum delay $\tau$ in the network. The worst-case analyses of (18) and (24) suggest that the threshold should be inversely proportional to $(\tau + 1)$. On the other hand, Figure 1 does not exhibit this behavior since our numerical experiments do not correspond to the worst-case choice of the delays $\{\tau_i\}$. Rather, they show that the recovery threshold of $k$ for a typical set of $\{\tau_i\}$ is not a strong function of $\tau$.

The recovery metric in Figure 1 matches that of the theorems and declares a trial to be in error when $\hat{\mathcal{I}} \neq \mathcal{I}$. However, it is also useful in many cases to consider how far the estimate $\hat{\mathcal{I}}$ is from the correct set. Therefore, in Figure 2 we use the performance metric of average fraction of detection errors, which corresponds to $\frac{|(\mathcal{I} \setminus \hat{\mathcal{I}}) \cup (\hat{\mathcal{I}} \setminus \mathcal{I})|}{k}$ and describes the number of errors in the estimated set of active users as a fraction of the average number of active users. With this metric, we see that Algorithm 1 fails gracefully as $k$ increases.

We also use the recovery metric of average fraction of detection errors to describe the power requirements of the active users in Figure 3. This figure shows that the power requirement, $\mathcal{E}_i$, described in
Theorem 1 is overly restrictive. Specifically, the rightmost point of the horizontal axis at 31dB provides the reference point as the power seen at the receiver as required by Theorem 1. The figure shows much less power is needed for recovering $\mathcal{I}$. It also shows that the required power is not a function of $k$ which is exactly in line with the results of our theory.

### B. Discussion

In order to place our results in context we note that $k \lesssim N/\log M$ scaling has also been suggested in [2] for the case of MUD in synchronous on–off RACs using the lasso and random Gaussian codewords. Here, however, we provide non-asymptotic results, as opposed to asymptotic results in [2], in the more general asynchronous case. Further, we provide guarantees that can be applied to arbitrary user codewords. For the codewords studied in Theorems 2 and 3 we have established that the MUD scheme for asynchronous on–off RACs has the ability to achieve roughly the same (non-asymptotic) scaling of the system parameters $k, N,$ and $M$ as that reported in [2] for the ideal case of synchronous channels.

With regard to the deterministic codewords introduced in Section IV-B, our construction is representative of a larger class of deterministic matrices derived from cyclic codes. We consider a particular cyclic code where the codewords are obtained by evaluating quadratic forms at elements of the field $\text{GF}(2^m)$. The worst case coherence $\mu(X)$ of the expanded codebook matrix is determined by the minimum weight of the code and we bound this quantity by elementary methods in Appendix B. We note that Yu and Gong [12] have calculated the exact weight distribution of a very similar code using more sophisticated methods from symplectic geometry.

Beyond the application to MUD for RACs, our results can also be related to work on model selection. Most directly, our work builds on the model selection theory of Candès and Plan [13] for the lasso. As in this paper, [13] provides guarantees for lasso that are based on worst-case coherence $\mu(X)$ and spectral norm $\|X\|_2$. However, a key assumption in [13] requires that the vector $\beta$ be “generic” in the sense that its support is uniform over its $(\tau + 1)M$ elements. In this paper, however, we assume a much different model: the support of $\beta$ is uniformly random over blocks of elements. In this light, the work here is related to recent work on block-sparse signals such as [14] which considers block-sparse signal recovery using a variant of orthogonal matching pursuit as opposed to the lasso. However, as work in the context of signal recovery rather than model selection, the work in [14] is not directly concerned with estimating $\mathcal{I}$ and cannot be applied to the MUD problem in RACs.

As a study of sparse signal recovery using a structured measurement matrix, this work relates to that of Romberg et al. [15]. Though [15] considers a different application and is concerned with signal recovery...
rather than estimating \( \mathcal{I} \), it studies block-Toeplitz matrices in a similar in structure to \( \mathbf{X} \). The approach in \cite{15}, however, differs since they provide recovery guarantees based on the restricted isometry property (RIP) of the matrix. However, by working with the RIP, the analysis is particular to randomly generated Toeplitz columns. In contrast, here we provide guarantees for any matrix \( \mathbf{X} \) with sufficiently small \( \mu(\mathbf{X}) \) and \( \|\mathbf{X}\|_2 \). Subsequently, we give both randomly generated and deterministic codeword designs satisfying the requirements. Further, our work provides support set recovery guarantees—in the spirit of model selection \cite{16}—rather than bounds on recovered signal error guaranteed by RIP.

VI. Conclusion

In this paper, we have described a novel system for MUD in RACs with the significant feature that we assume that user codewords are received asynchronously at the receiver. We leveraged and generalized sparse signal theory to provide recovery guarantees of a lasso based algorithm to find the set of active users. While our results are general and apply for arbitrary sets of codewords, we specialized them to both a randomly generated set as well as a code we designed for the application.

The implications of the scaling behavior outlined in the pairs of inequalities in Theorems 2 and 3 are quite positive in the important special case of fixed-bandwidth spread spectrum waveforms and a BS serving a bounded geographic region. Specifically, they signify that—for any fixed number of temporal signal space dimensions \( N \) and maximum delay \( \tau \) in the system—the proposed MUD scheme can accommodate \( M \lesssim \exp(O(N^{1/3})) \) total users in the case of random signaling and \( M \) polynomial in \( N \) when using our algebraic code design. Both sets of codewords allow \( k \lesssim N/\log M \) active users in the system. This is a significant improvement over the \( k \leq M \lesssim N \) scaling suggested by the use of classical matched-filtering-based approaches to MUD employing orthogonal signaling.

Appendix A

Proof of the Main Result

In this appendix, we provide a proof of Theorem 1. Before proceeding further, however, let us develop some notation to facilitate the forthcoming analysis. Throughout this appendix, we use \( \mathbf{X}_\mathcal{B} \) to denote the block subdictionary of \( \mathbf{X} \) obtained by collecting the Toeplitz blocks of \( \mathbf{X} \) corresponding to the indices of the active users: \( \mathbf{X}_\mathcal{B} \overset{\text{def}}{=} \{ \mathbf{X}_i : i \in \mathcal{I} \} \). In addition, we use \( \mathbf{X}_\mathcal{S} \) to denote the \((N + \tau) \times |\mathcal{I}| \) submatrix obtained by collecting the columns of \( \mathbf{X} \) corresponding to the nonzero entries of \( \mathbf{\beta} \), while we use \( \mathbf{\beta}_\mathcal{S} \) to denote the \(|\mathcal{I}| \)-dimensional vector comprising of the nonzero entries of \( \mathbf{\beta} \). Finally, we use \( \text{sgn}(\cdot) \) for elementwise signum function: where \( \text{sgn}(z) \overset{\text{def}}{=} z/|z| \) for any \( z \in \mathbb{R} \).
The basic idea behind the proof of Theorem 1 follows from the proof of [13, Theorem 1.3]. Specifically, using \( S \subset \{1, \ldots, M(\tau + 1)\} \) to denote the set of the locations of the nonzero entries of \( \beta \), we have from [13, Lemma 3.4] that the lasso solution \( \hat{\beta} \equiv \beta + h \) satisfies \( h_{S^c} = 0 \) and
\[
h_S = (X_S^T X_S)^{-1}[X_S^T w - \lambda \text{sgn}(\beta_S)]
\] (25)
if \( \min_{i \in S} |\beta_i| > 4\lambda \) and the following five conditions are met:

- \( C_1 \) – Invertibility condition: \( \|(X_S^T X_S)^{-1}\|_2 \leq 2 \).
- \( C_2 \) – Noise stability: \( \|X_S^T (I - X_S (X_S^T X_S)^{-1} X_S^T) w\|_\infty \leq \lambda \).
- \( C_3 \) – Complementary noise stability: \( \|X_S^T c (I - X_S (X_S^T X_S)^{-1} X_S^T) w\|_\infty \leq \lambda \sqrt{2} \).
- \( C_4 \) – Size condition: \( \|(X_S^T X_S)^{-1} \text{sgn}(\beta_S)\|_\infty \leq 3 \).
- \( C_5 \) – Complementary size condition: \( \|X_S^T X_S (X_S^T X_S)^{-1} \text{sgn}(\beta_S)\|_\infty \leq \frac{1}{4} \).

Further, it trivially follows in this case that the set of non-zero elements of \( \hat{\beta} \) is \( S \), which guarantees that \( \hat{I} = I \). Our goal then is to consider the probability of each one of these conditions not being met under the assumptions of Theorem 1 and the proof of the theorem would then simply follow from the union bound.

A. Invertibility Condition

In order to establish the invertibility condition, we will make use of the following proposition from [17].

**Proposition 1** ([17]): Fix \( q = 2 \log (M(\tau + 1)) \) and define the block coherence
\[
\mu_B(X) \overset{\text{def}}{=} \max_{1 \leq i,j \leq M} \|X_i^T X_j - 1_{\{i=j\}}I\|_2.
\] (26)

Then, for \( E_q Z \overset{\text{def}}{=} \|E[Z]^q\|^{1/q} \) and \( \delta \overset{\text{def}}{=} k/M \), we have the following bound
\[
E_q \|X_B^T X_B - I\|_2 \leq 20\mu_B(X) \log (M(\tau + 1)) + \delta \|X\|_2^2 + 9\sqrt{\delta \log (M(\tau + 1)) (1 + \tau \mu(X))} \|X\|_2. \tag{27}
\]
We would like to bound (27) via bounds on \( \mu_B(X) \), \( \mu(X) \) and \( \|X\|_2 \). First, we can use the linear algebra fact \( \|\cdot\|_2 \leq \sqrt{\|\cdot\|_1 \cdot \|\cdot\|_\infty} \) on (26) to show that \( \mu_B(X) \leq (\tau + 1)\mu(X) \). Thus, we can rearrange the inequalities of (10) and (11) to obtain
\[
\mu(X) \leq \frac{1}{c(\tau + 1) \log (M(\tau + 1))},
\] (28)
\[
\|X\|_2^2 \leq \frac{1}{c\delta \log (M(\tau + 1))}, \quad \text{and}
\] (29)
\[
\mu_B(X) \leq \frac{1}{c \log (M(\tau + 1))}.
\] (30)
Substituting these inequalities into (27) and choosing \(c\) appropriately large yields \(\mathbb{E}_q \|X_B^T X_B - I\|_2 < \frac{1}{4}\).

Finally, notice that \(X_S\) is a submatrix of \(X_B\) and therefore we trivially have \(\|X_S^T X_S - I\|_2 \leq \|X_B^T X_B - I\|_2\). It can then be easily seen from the Markov inequality that

\[
\Pr(\|X_S^T X_S - I\|_2 > 1/2) \leq \frac{2^q (\mathbb{E}_q \|X_B^T X_B - I\|_2)^q}{(M(\tau + 1))^{-2\log 2}} \tag{31}
\]

where \((a)\) follows from the fact that \(\mathbb{E}_q \|X_B^T X_B - I\|_2 < \frac{1}{4}\). We have now established that \(\|X_S^T X_S\|_2 \in (1/2, 3/2)\) with high probability, which implies that

\[
\Pr(C_1^c) \leq \left( M(\tau + 1) \right)^{-2\log 2}. \tag{32}
\]

### B. Noise Stability

In order to establish the noise-stability condition, we first condition on \(C_1\) (the invertibility condition). Next, we denote the \(j\)-th column of \(X_S (X_S^T X_S)^{-1}\) by \(z_j\) and note that

\[
\| (X_S^T X_S)^{-1} X_S^T w \|_\infty = \max_{1 \leq j \leq |S|} |\langle z_j, w \rangle|. \tag{33}
\]

Further, since the noise vector \(w\) is distributed as \(\mathcal{N}(0, I)\), we also have that \(\langle z_j, w \rangle \sim \mathcal{N}(0, \|z_j\|_2^2)\).

Finally, note that conditioned on \(C_1\), we have the upper bound

\[
\|z_j\|_2 \leq \|X_S (X_S^T X_S)^{-1}\|_2 \leq \|X_S\|_2 \| (X_S^T X_S)^{-1}\|_2 \leq \sqrt{2}.
\]

The rest of the argument now follows easily from bounds on the maximum of a collection of arbitrary Gaussian random variables. Specifically, it can be seen from the previous discussion and a real-valued version of [16, Lemma 6] that

\[
\Pr \left( \| (X_S^T X_S)^{-1} X_S^T w \|_\infty \geq \sqrt{2}t | C_1 \right) \leq \frac{2M e^{-t^2/2}}{\sqrt{2\pi t}}.
\]

We substitute \(t = \lambda/\sqrt{2}\) in the above expression to obtain

\[
\frac{2M e^{-\lambda^2/4}}{\sqrt{\pi \lambda}} = \frac{1}{M(\tau + 1) \sqrt{2\pi \log(M \sqrt{\tau + 1})}}.
\]

Summarizing, we have that the noise stability condition satisfies

\[
\Pr(C_2^c | C_1) \leq \frac{1}{M(\tau + 1) \sqrt{2\pi \log(M \sqrt{\tau + 1})}}. \tag{34}
\]
C. Complementary Noise Stability

In order to establish the complementary noise-stability condition, we use ideas similar to the ones used in the previous section. To begin with, we again condition on the event \( C_1 \) and use \( P_{X_S} \stackrel{\text{def}}{=} X_S (X_S^T X_S)^{-1} X_S^T \) to denote the orthogonal projector onto the column span of \( X_S \). Next, we use \( z_j \) to denote the \( j \)-th column of \( (I - P_{X_S}) X_S \) and note that

\[
\| X_S^T (I - P_{X_S}) w \|_\infty = \max_{1 \leq j \leq |S|} |\langle z_j, w \rangle|.
\]

Finally, given that \( P_{X_S} \) is a projection matrix and the columns of \( X \) have unit norm, we have that

\[
\| z_j \|_2 = \| (I - P_{X_S}) X_S e_j \|_2 \leq 1
\]

where \( e_j \) denotes the \( j \)-th canonical basis vector.

It is now easy to see that, since \( \langle z_j, w \rangle \) is also distributed as \( \mathcal{N}(0, \| z_j \|_2^2) \), we can make use of [16, Lemma 6] to obtain

\[
\Pr \left( \| X_S^T (I - P_{X_S}) w \|_\infty \geq t | C_1 \right) \leq \frac{2M(\tau + 1)e^{-t^2/2}}{\sqrt{2\pi} t}.
\]

We substitute \( t = \lambda/\sqrt{2} \) in the above expression to obtain

\[
\frac{2M(\tau + 1)e^{-\lambda^2/4}}{\sqrt{\pi} \lambda} \leq \frac{1}{M \sqrt{2\pi \log(M \sqrt{\tau} + 1)}}.
\]

Summarizing, we have that the complementary noise stability condition satisfies

\[
\Pr(C_3^c | C_1) \leq \frac{1}{M \sqrt{2\pi \log(M \sqrt{\tau} + 1)}}.
\]

D. Size Condition

In order to establish the size condition, we first write

\[
\|(X_S^T X_S)^{-1} \text{sgn}(\beta_S)\|_\infty \stackrel{(a)}{=} \|(X_S^T X_S)^{-1} - I\) \text{sgn}(\beta_S)\|_\infty + \|\text{sgn}(\beta_S)\|_\infty
\]

\[
= \|(X_S^T X_S)^{-1} - I)\text{sgn}(\beta_S)\|_\infty + 1
\]

where \((a)\) follows from the triangle inequality. Next, we once again use \( z_j \) to denote the \( j \)-th column of

\[
((X_S^T X_S)^{-1} - I),
\]

which simply implies that \( \|(X_S^T X_S)^{-1} - I)\text{sgn}(\beta_S)\|_\infty = \max_{1 \leq j \leq |S|} |\langle z_j, \text{sgn}(\beta_S) \rangle| \).

Now define \( A = (X_S^T X_S - I) \) and condition on the event \( C_1 \). Then it follows from the Neumann series (cf. [13, p. 2171]) that \( \| z_j \|_2 \leq 2 \| A e_j \|_2 \). Further, since \( X_S \) is a submatrix of \( X_B \), we have \( \| A e_j \|_2 \leq \| (X_B^T X_B - I) e_{j'} \|_2 \), where \( j' \) is such that the \( j' \)-th column of \( X_B \) matches the \( j \)-th column of \( X_S \).

Finally, define the diagonal matrix \( Q \stackrel{\text{def}}{=} \text{diag}(\delta_1, \ldots, \delta_M) \) with the “random activation variables” \( \{ \delta_i \} \) on the diagonal and define a new matrix \( R = Q \otimes I_{\tau + 1} \), where \( \otimes \) denotes the Kronecker product. Next,
use the notation $H \overset{\text{def}}{=} (X^T X - I)$ and notice that $\|(X^T B X - I) e_j\|_2 = \|R H e_j\|_2$, where $j''$ is such that the $j''$-th column of $X$ matches the $j$-th column of $X S$. In addition, note that $H = \begin{bmatrix} H_1 & H_2 & \ldots & H_M \end{bmatrix}$ has a block structure that can be expressed as

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & \ldots & H_{1,M} \\ H_{2,1} & H_{2,2} & \ldots & H_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ H_{M,1} & H_{M,2} & \ldots & H_{M,M} \end{bmatrix}$$  (39)

where $H_{i,j} = X_i^T X_j - 1_{i=j} I$, $1 \leq i, j \leq M$, and $H_i = [H_{i,1}^T \ldots H_{i,M}^T]^T$. We now define two blockwise norms on $H$ as follows: $\|H\|_{B,1} \overset{\text{def}}{=} \max_{1 \leq i \leq M} \|H_i\|_2$, and $\|H\|_{B,2} \overset{\text{def}}{=} \max_{1 \leq i, j \leq M} \|H_{i,j}\|_2$.

Then it follows from the preceding discussion and the structure of the block matrix $H$ that

$$\|z_j\|_2 \leq 2\|A e_j\|_2 \leq 2\|R H e_{j''}\|_2 \leq 2\|R H\|_{B,1}. \quad (40)$$

Our next goal then is to provide a bound on $\|R H\|_{B,1}$ and for this we resort to [17, Lemma 5].

**Proposition 2** ([17]): For $q \geq 2 \log M$ and $\delta = k/M$, we have that

$$\mathbb{E}_q\|R H\|_{B,1} \leq 2^{1.5} \sqrt{q} \|H\|_{B,1} + \sqrt{\delta} \|H\|_{B,1}. \quad (41)$$

Now notice from the definition of $H$ and $\|\cdot\|_{B,2}$ that $\|H\|_{B,2} \equiv \mu_B (X) \leq (\tau + 1) \mu (X)$. In addition, we have from the definition of $H$ and $\|\cdot\|_{B,1}$ that

$$\|H\|_{B,1} \overset{(b)}{\leq} \max_{1 \leq i \leq M} \|X^T X_i\|_2 + \|I_{\tau+1}\|_2 \overset{(c)}{\leq} \sqrt{1 + \tau \mu (X)} \|X\|_2 + 1 \quad (42)$$

where (b) follows from the definition of the spectral norm and the triangle inequality, while (c) mainly follows from the fact that $\|X_i\|_2 \leq \sqrt{1 + \tau \mu (X)}$ because of the Geršgorin disc theorem [18]. We can now fix $q = 2 \log M$ and make use of the above bounds to conclude from Proposition 2 that

$$\mathbb{E}_q\|R H\|_{B,1} \leq 4(\tau + 1) \mu (X) \sqrt{\log M} + \sqrt{\delta (1 + \tau \mu (X))} \|X\|_2 + \sqrt{\delta}. \quad (43)$$

We can now substitute (28) and (29) into the above expression to obtain $\mathbb{E}_q\|R H\|_{B,1} \leq \gamma_0$ with

$$\gamma_0 \overset{\text{def}}{=} \frac{4}{c'} \sqrt{\log (M(\tau + 1))} + \frac{2}{c' \log (M(\tau + 1))} \sqrt{1 + \frac{1}{c' \log (M(\tau + 1))}}. \quad (44)$$

In order to establish the size condition, we now define the event $E = \{\max_{1 \leq j \leq |S|} \|z_j\|_2 < \gamma\}$ and make use of the Markov inequality along with (40) and the preceding discussion to obtain

$$\Pr(E^c) \leq \gamma^{-q} \left(\mathbb{E}_q \max_{1 \leq j \leq |S|} \|z_j\|_2\right)^q \leq \left( \frac{2}{\gamma} \mathbb{E}_q\|R H\|_{B,1} \right)^q \leq \left( \frac{2 \gamma_0}{\gamma} \right)^q.$$

January 10, 2011
DRAFT
Finally, we use \( Z \overset{\text{def}}{=} \max_{1 \leq j \leq |S|} |\langle z_j, \text{sgn}(\beta_S) \rangle| \) and conclude that
\[
\Pr(Z \geq t) \leq \Pr(Z \geq t | E) + \Pr(E^c) \overset{(d)}{\leq} 2Me^{-t^2/2\gamma^2} + (2\gamma_0/\gamma)^q
\]
where \((d)\) is a consequence of the Hoeffding inequality [19, Proposition 16] and the union bound. The condition is now established from (38) by setting \( t = 2 \) in the above expression. Further, set
\[
\gamma \leq \sqrt{\frac{2}{(1 + 2 \log 2) \log M}}
\]
which leads to \( 2Me^{-2/\gamma^2} \leq 2M^{-2\log 2} \) and
\[
\frac{\gamma_0}{\gamma} \leq \frac{2(\sqrt{1 + c'} + 2)}{0.9155c'} < 1/4.
\]
Therefore, we obtain that \( \Pr(E^c) \leq (1/2)^q \leq M^{-2\log 2} \) and thus we have that the size condition does not hold with probability at most
\[
\Pr(C_1^c | C_1) \leq 3M^{-2\log 2}.
\]

E. Complementary Size Condition

In order to establish the complementary size condition, we proceed similar to the case of the “size condition” and define \( z_j \) as \( z_j \overset{\text{def}}{=} (X^T_S X_S)^{-1}X^T_S X_S e_j \). It can then be easily seen that \( \|X^T_S X_S (X^T_S X_S)^{-1} \text{sgn}(\beta_S)\|_\infty = \max_{1 \leq j \leq |S^e|} |\langle z_j, \text{sgn}(\beta_S) \rangle| \). Now condition on the event \( C_1 \) and notice that \( \|z_j\|_2 \leq 2\|X^T_S X_S e_j\|_2 \), \( j = 1, \ldots, |S^c| \).

We now define \( X_{B'} \overset{\text{def}}{=} [X_i : i \in I^c] \) and consider the set of indices \( T_1 \overset{\text{def}}{=} \{ j' : X_S e_j' \in X_{B'} \} \). It is then easy to argue by making use of the notation developed in Section A-D that if \( j \in T_1 \) then
\[
\|X^T_S X_S e_j\|_2 \leq \max_{i \in I^c} \|X^T_i X_i\|_2 = \|X^T_{B'} X_{B'}\|_{B,1} \overset{(a)}{\leq} \|RH\|_{B,1}
\]
where \((a)\) follows from the fact that \( X^T_{B'} X_{B'} \) is a submatrix of \( RH \). We therefore have from the discussion following Proposition [2] and the Markov inequality that \( \forall j \in T_1 \) and for \( q = 2 \log M \) and \( \gamma > 0 \)
\[
\Pr(\|X^T_S X_S e_j\|_2 > \gamma) \leq \frac{[\|E_q\|\|RH\|_{B,1}]^q}{\gamma^q} \leq \left( \frac{\gamma_0}{\gamma} \right)^q.
\]

Finally, the argument involving \( j \in T_1^c \) is a little more involved but follows along similar lines. Specifically, fix any \( j \in T_1^c \) and define \( i' \in I \) to be such that \( X_S e_j \) is a column of \( X_{i'} \). Next, define \( \tilde{x}_{S \cap i'} \) to be the column of \( X_S \) that lies within the Toeplitz block \( X_{i'} \) and \( \tilde{X}_{S \setminus i'} \) to be the submatrix constructed
by removing the column \( \tilde{x}_{S \cap i'} \) from \( X_S \). Then, if we use the notation \( X_{B \setminus i'} \) \( \overset{\text{def}}{=} \) \( \{ X_i : i \in B \setminus \{ i' \} \} \), it can be verified that for any \( j \in T_1^c \) we have
\[
\| X_S^T X_{S'} e_j \|_2^2 = \| \tilde{X}_{S \cap i'}^T X_{S'} e_j \|_2^2 + \| \tilde{x}_{S \cap i'}^T X_{S'} e_j \|_2^2 \\
\leq \max_{i' \in I} \| X_{B \setminus i'}^T X_{S'} e_j \|_2^2 + \| X_{\tilde{S}}^T X_{S'} e_j \|_2^2 + \| \tilde{x}_{S \cap i'}^T X_{S'} e_j \|_2^2 \\
\leq \| RH \|_{B, 1}^2 + \mu^2(X)
\]
(51)
where \( (b) \) again makes use of the fact that the spectral norm of a matrix is an upper bound for the spectral norm of any of its submatrices. We therefore once again obtain from the discussion following Proposition \( 2 \) and the Markov inequality that \( \forall \; j \in T_1^c \) and for \( q = 2 \log M \) and \( \gamma > 0 \)
\[
\Pr(\| X_S^T X_{S'} e_j \|_2 > \gamma) \leq \Pr(\| RH \|_{B, 1} > \sqrt{\gamma^2 - \mu^2(X)}) \leq \left( \frac{\gamma_0}{\sqrt{\gamma^2 - \mu^2}} \right)^q.
\]
(52)
We can now define the event \( E = \{ \| X_S^T X_{S'} e_j \|_2 \leq \gamma \} \) and use the notation \( Z \overset{\text{def}}{=} \max_{1 \leq j \leq |S|} |z_j, \text{sgn}(\beta_S)| \) to conclude from (50) and (52) that
\[
\Pr(Z \geq t) \leq \Pr(Z \geq t | E) + \Pr(E^c)
\]
(53)
where \( (c) \) follows from [19, Proposition 16] and the union bound. The condition is now established by setting \( t = \frac{1}{4} \) in the above expression. Further, set
\[
\gamma \leq \sqrt{128(1 + 2 \log 2) \log(M(\tau + 1))}
\]
(54)
which yields \( 2M(\tau + 1)e^{-1/128\gamma^2} \leq 2(M(\tau + 1))^{-2 \log 2} \) and \( \frac{\gamma_0}{\sqrt{\gamma^2 - \mu^2}} \leq \frac{2\sqrt{1 + \tau + \frac{4}{\gamma^2 \mu^2}}}{\sqrt{0.0572 - 1/\gamma^2}} < 1/2 \). Therefore, we obtain that \( \Pr(E^c) \leq 2(\gamma_0/\sqrt{\gamma^2 - \mu^2})^q \leq 2(1/2)^q \leq 2(M(\tau + 1))^{-2 \log 2} \) and thus we have that the size condition satisfies
\[
\Pr(C_3^c | C_1) \leq 4(M(\tau + 1))^{-2 \log 2}.
\]
(55)
\[\Box\]

**F. Proof of Theorem 1**

The proof of Theorem 1 follows from the preceding discussion by taking a union bound over all the respective conditions and removing the conditionings: \( \Pr((C_1 \cap C_2 \cap C_3 \cap C_4 \cap C_5) \cap C_1) \leq \Pr(C_1^c) + \Pr(C_2 | C_1) + \Pr(C_3 | C_1) + \Pr(C_4 | C_1) + \Pr(C_5 | C_1) \). Consequently, we obtain that the probability of error is upper bounded by \( 5(M(\tau + 1))^{-2 \log 2} + 2M^{-1}(2\pi \log(M\sqrt{\tau + 1}))^{-1/2} \).
APPENDIX B

PROOF OF LEMMA \[21\]

In order to bound the sum of \[21\], we will use the following propositions.

**Proposition 3:** For \(x,y \in \text{GF}(2^m)\) and \(i = 1,2,\ldots\)

\[
x^{2^i+1} + y^{2^i+1} = (x + y)^{2^i+1} + \sum_{j=0}^{i-1} (xy)^{2^i}(x + y)^{2^i-2^{j+1}+1}
\]

**Proof:** We prove this by induction and application of \((x + y)^{2^i} = x^{2^i} + y^{2^i}\). That is, we first note that the lemma holds for \(i = 1\) and, assuming true for \(i\), we have

\[
(x + y)^{2^{i+1}+1} = (x + y)^{2^i}(x + y)^{2^i+1}
\]

\[
= (x + y)^{2^i} [x^{2^i+1} + y^{2^i+1} + \sum_{j=0}^{i-1} (xy)^{2^i}(x + y)^{2^i-2^{j+1}+1}]
\]

\[
= (x^{2^i} + y^{2^i}) (x^{2^i+1} + y^{2^i+1}) + \sum_{j=0}^{i-1} (xy)^{2^i}(x + y)^{2^{i+1}-2^{j+1}+1}
\]

\[
= x^{2^{i+1}+1} + y^{2^{i+1}+1} + (xy)^{2^i}(x + y) + \sum_{j=0}^{i-1} (xy)^{2^i}(x + y)^{2^{i+1}-2^{j+1}+1}.
\]

Incorporating the middle term in the sum completes the proof. ■

**Proposition 4** ([\[11\] pp.278–9]): The quadratic polynomial \(x^2 + fx + g\) with coefficients in \(\text{GF}(2^m)\) and \(f \neq 0\) has two distinct roots in \(\text{GF}(2^m)\) if \(\text{Tr}(g/f^2) = 0\) and no roots in \(\text{GF}(2^m)\) if \(\text{Tr}(g/f^2) = 1\).

**Proposition 5:** (a) The cardinality of \( \{g \in \text{GF}(2^m) : \text{Tr}(\alpha g) = c_1\} \) is \(2^{m-1}\) for \(\alpha \in \text{GF}(2^m), \alpha \neq 0\) and \(c_1 \in \text{GF}(2)\). (b) The cardinality of \( \{g \in \text{GF}(2^m) : \text{Tr}(\alpha g) = c_1, \text{Tr}(\beta g) = c_2\} \) is \(2^{m-2}\) for \(\beta \in \text{GF}(2^m), \beta \neq \alpha, \beta \neq 0\) and \(c_1, c_2 \in \text{GF}(2)\).

**Proof:** Let \(\{\eta_i\}_{i=1}^m\) and \(\{\lambda_i\}_{i=1}^m\) be dual bases of \(\text{GF}(2^m)\) [\[11\] p.117] and consider \(\alpha\) and \(g\) in these bases respectively as \(\alpha = a_1\eta_1 + \cdots + a_m\eta_m\) and \(g = \gamma_1\lambda_1 + \cdots + \gamma_m\lambda_m\) for \(a_i, \gamma_i \in \text{GF}(2)\). Then \(\text{Tr}(\alpha g) = a_1\gamma_1 + \cdots + a_m\gamma_m = c_1\) is a restriction of a single degree of freedom in selecting \(\{\gamma_i\}_{i=1}^m\).

Similarly, \(\text{Tr}(\beta g) = c_2\) restricts an additional degree of freedom. ■

We are now ready to bound the sum given by \([21]\). We will begin with a simple (yet required) case which illustrates our use of Proposition \([5]\). Suppose the non-zero vector \(\alpha\) is zero everywhere but at \(\alpha_0\). In this case we have

\[
S = \sum_{x \in \text{GF}^*(2^m)} (-1)^{\text{Tr}(\alpha_0 x)} = \sum_{x \in \text{GF}(2^m)} (-1)^{\text{Tr}(\alpha_0 x)} - 1
\]
where we’ve completed the sum to be over all of \( GF(2^m) \). By Proposition 5 (a), \( \text{Tr}(\alpha_0 x) = 1 \) for precisely precisely half the \( 2^m \) terms of the sum. Thus, the sum is 0 and, in this case, \(|S| = 1\). For the remainder of the proof, we will assume that \( \alpha_i \) is non-zero for some \( i \geq 1 \).

Considering the square of (21), by using the linearity of the trace we have

\[
S^2 = \sum_{x \in GF^*(2^m)} \sum_{y \in GF^*(2^m)} (-1)^{\text{Tr}(x+y)} \alpha_0 (x+y) + \sum_{i=1}^{t} \alpha_i (x^{2^i + 1} + y^{2^i + 1})
\]

\[
= 2^m - 1 + \sum_{x \in GF^*(2^m)} \sum_{y \in GF^*(2^m) \setminus \{x\}} (-1)^{\text{Tr}(x+y)} \alpha_0 (x+y) + \sum_{i=1}^{t} \alpha_i (x^{2^i + 1} + y^{2^i + 1})
\]

\[
= 2^m - 1 + \sum_{x \in GF^*(2^m)} \sum_{y \in GF^*(2^m) \setminus \{x\}} (-1)^{\text{Tr}(x+y)} \alpha_0 ((x+y)^{2^i + 1} + \sum_{j=0}^{t-1} (x^{2^i} + y^{2^i}) (x+y)^{2^i - 2^{i+1} + 1})
\]

In the last equality we have used Proposition 3 so that we may apply the change of variables given by \( f = x + y \) and \( g = xy \). To justify this substitution we note that

\[
\{(x,y) : x \in GF^*(2^m), y \in GF^*(2^m), y \neq x\} = \{(f,g) : f \in GF^*(2^m), g \in GF^*(2^m), \text{Tr}(g/f^2) = 0\}.
\]

To see this, consider quadratics \((z+x)(z+y) = z^2 + f z + g\) with non-zero roots. The first set generates all quadratics with two solutions by enumerating the roots while, by Proposition 4, the second set generates the same quadratics by enumerating the coefficients. Since with this substitution both \((x,y)\) and \((y,x)\) map to \((f,g)\) we account for the extra factor of 2 below. We now have

\[
S^2 = 2^m - 1 + 2 \sum_{f,g \in GF^*(2^m)} (-1)^{\text{Tr}(f + \sum_{i=1}^{t} \alpha_i f^{2^i + 1} + \sum_{j=0}^{t-1} g^{2^i} f^{2^i - 2^{i+1} + 1})}
\]

\[
= 2^m - 1 + 2 \sum_{f \in GF^*(2^m)} (-1)^{\text{Tr}(f + \sum_{i=1}^{t} \alpha_i f^{2^i + 1})} \sum_{g \in GF^*(2^m) \setminus \{f\}} (-1)^{\text{Tr}(\sum_{i=1}^{t} \sum_{j=0}^{t-1} \alpha_i g^{2^i} f^{2^i - 2^{i+1} + 1})}
\]

\[
= 2^m - 1 + 2 \sum_{f \in GF^*(2^m)} (-1)^{\text{Tr}(f + \sum_{i=1}^{t} \alpha_i f^{2^i + 1})} \left[ \sum_{g \in GF^*(2^m) \setminus \{f\}} (-1)^{\text{Tr}(\sum_{i=1}^{t} \sum_{j=0}^{t-1} \alpha_i g^{2^i} f^{2^i - 2^{i+1} + 1})} - 1 \right]
\]

\[
\leq 3(2^m - 1) + 2 \sum_{f \in GF^*(2^m)} (-1)^{\text{Tr}(f + \sum_{i=1}^{t} \alpha_i f^{2^i + 1})} \sum_{g \in GF^*(2^m) \setminus \{f\}} (-1)^{\text{Tr}(\sum_{i=1}^{t} \sum_{j=0}^{t-1} \alpha_i g^{2^i} f^{2^i - 2^{i+1} + 1})}
\]

(56)

where, in the third equality, we’ve completed the sum in \( g \) to include \( g = 0 \). The resulting subtraction creates a sum over \( f \) which we trivially bound by \( 2^m - 1 \). Turning our attention to the innermost sum.
over $g$, we will show that the sum is either 0 or $2^{m-1}$. Further, we will bound the number of $f$ for which it is not zero.

To separate $g$, we can use the linearity of the trace and $\text{Tr}(x) = \text{Tr}(x^{2^{-j}})$ for each $j$ and rewrite the exponent of the inner sum as $\text{Tr}\left[\left(\sum_{i=1}^{t} \sum_{j=0}^{i-1} \alpha_i^{2^{-j}} f^{2^{i-j}+2^{j-2}}g\right)\right] = \text{Tr}(\Gamma_f g)$ where we’ve introduced $\Gamma_f \in \text{GF}(2^m)$ to simplify notation. Suppose, for a fixed $f$, there exists some $g$ with $\text{Tr}(g/f^2) = 0$ such that $(-1)^{\text{Tr}(\Gamma_f g)} = -1$. Then we must have $\Gamma_f \neq 1/f^2$ and $\Gamma_f \neq 0$. In this case, Proposition 5 tells us that the inner sum of (56) evaluates to 0 since part (a) gives the size of the sum while (b) shows precisely half the terms take value (-1). Thus, we are interested in when $(-1)^{\text{Tr}(\Gamma_f g)}$ maps all of the subset $\{g \in \text{GF}(2^m) : \text{Tr}(g/f^2) = 0\}$ to 1.

When $(-1)^{\text{Tr}(\Gamma_f g)}$ is a trivial map of the subset, we have $\{g : \text{Tr}(\Gamma_f g) = 0\} \supseteq \{g : \text{Tr}(g/f^2) = 0\}$ which provides two cases. The first is that $\Gamma_f = 0$ and the above inclusion is strict. In second case, when $\Gamma_f \neq 0$, the sets have same cardinality by Proposition 5 and, thus, the two sets are equal. In this case, by $\text{Tr}[\text{Tr}(f + 1/f^2)]g] = 0 \ \forall g$, the non-degeneracy of the trace [20] Proposion 28.87 tells us $\Gamma_f = 1/f^2$. In both cases, Proposition 5 gives the size of the inner sum of (56) as $2^{m-1}$. The task now becomes to bound the number of $f$ for which each of these cases occur.

$\Gamma_f = 0$ defines the following polynomial in $f$:

$$0 = \sum_{i=1}^{t} \sum_{j=0}^{i-1} \alpha_i^{2^{-j}} f^{2^{i-j}+2^{j-2}} = \sum_{i=1}^{t} \sum_{j=0}^{i-1} \alpha_i^{2^{i-j}-1} f^{2^{i-1}+j-1+2^j-1-2^0}$$

where, in the second equality, we’ve used $(x+y)^{2^{i-1}} = x^{2^{i-1}} + y^{2^{i-1}}$ to ensure the powers of $f$ are positive integers. The degree of this polynomial is at most $2^{2^{t-1}} + 2^{t-1} - 2^t$ and thus we at most $2^{2^{t-1}} + 2^{t-1} - 2^t$ roots at which $\Gamma_f = 0$.

The case for $f^2 \Gamma_f = 1$ is similar and follows the same steps on a slightly different polynomial. In this case we find there are at most $2^{2^{t-1}} + 2^{t-1}$ values of $f$ for which $\Gamma_f = 1/f^2$. Combining the two cases, we find that there are at most $2^{2^t}$ values of $f$ for which $(-1)^{\text{Tr}(\Gamma_f g)}$ is a trivial map over the sum. Returning to (56), we’ve found the sum in $f$ have terms with values of either 0 or $\pm 2^{m-1}$ with the non-zeros terms occurring at most $2^{2t}$ times. Thus,

$$S^2 \leq 3(2^m - 1) + 2 \times 2^{2t} \times 2^{m-1} \leq 2^{m+2t+1}$$

Taking the root gives the result.

**REFERENCES**


