Exploiting Sparse Dynamics For Bandwidth Reduction In Cooperative Sensing Systems

Harish Ganapathy, *Member, IEEE*, Constantine Caramanis, *Member, IEEE*, and Lei Ying, *Member, IEEE*

***Abstract*—Recently, there has been a significant interest in developing *cooperative sensing* systems for certain types of wireless applications. In such systems, a group of sensing nodes periodically collects measurements about the signals being observed in the given geographical region and transmits these measurements to a central node, which in turn processes this information to recover the signals. For example, in cognitive radio networks, the signals of interest are those generated by the primary transmitters and the sensing nodes are the secondary users. In such networks, it is critically important to be able to reliably determine the presence or absence of primary transmitters in order to avoid causing interference. The standard approach to transmitting these measurements from the sensor nodes to the fusion center has been to use orthogonal channels. Such an approach quickly places a burden on the control-channel-capacity of the network that would scale linearly in the number of cooperating sensing nodes. In this paper, we show that as long as one condition is satisfied: the *dynamics of the observed signals are sparse*, i.e., the observed signals do not change their values very rapidly in relation to the time-scale at which the measurements are collected, we can significantly reduce the control bandwidth of the system while achieving near full (linear) bandwidth performance.**

***Index Terms*—Compressed sensing, compressive sampling, cooperative sensing, null-space property, restricted isometry.**

I. Introduction

COOPERATIVE sensing is a promising technique that has received a lot of attention recently due to the necessity for reliable decision-making. Here, a group of wireless nodes collects measurements about some signals of interest. The observations are then reported to a central node, which in turn applies some appropriately-chosen decision rule to recover the signals. Such an architecture finds application in two closely-related and well-studied settings.

The first setting is (de-)centralized detection in sensor networks — a group of sensor nodes commanded by a fusion center — where the observed signals could be temperature or humidity readings, military targets, etc. Much work (see [1]–[4] and references therein), both theoretical and practical, has studied and optimized various aspects of the cooperative sensing architecture, e.g., quantization functions at the sensing nodes, the combining process and decision metric at the fusion center, the communication scheme between the sensing nodes and the fusion center, etc. This is a rich area of research and hence, our references are far from a comprehensive list. The second setting is a cognitive radio network where the observed signals are other unlicensed (secondary) or licensed (primary) transmitters. While the ideas in this paper are applicable to both the aforementioned settings, we review some literature from the latter as it is most relevant to the technical developments proposed here.

A. Spectrum Sensing in Cognitive Radio Networks

Cognitive radio technology, introduced by J. Mitola [5], is a promising solution to the growing scarcity of wireless spectrum, one that can potentially increase the spectrum utilization efficiency as recognized by the FCC [6], [7]. Traditionally, a portion of spectrum is allocated or licensed for use by a specific group of users by regulatory agencies. On the other hand, cognitive radio networks call for cognitive (unlicensed/secondary) users to operate on the same frequency band as the primary licensed users while attempting to access the spectrum *seamlessly*. In other words, the cognitive users need to adjust their operating parameters to guarantee minimal impact to the primary licensed users. For example, DARPA's *Next Generation* program [8] has been interested in developing spectrum sensing techniques that prevent interference to existing occupants of the frequency band. To increase spectral efficiency, the FCC recently opened up TV whitespaces (54 MHz-806 MHz) for unlicensed use [9]. While the ruling calls for access to a central database to determine TV band availability, the traditional use case for spectrum sensing, we believe it still plays an important role in providing acceptable quality-of-service (QoS) over unlicensed bands. For example, a database administrator could provide multiple classes of service to a *querying* secondary user where the higher class would be provided with a per-TV-band estimate of the interference statistics from other unlicensed users in the band. Spectrum sensing might be necessary in this context in order to collect such statistics.

Cooperative or collaborative sensing in the context of cognitive radio networks consists of a group of secondary users or specially-placed sensor nodes that collect measurements over some sensing time window about the activity of the primary user and transmit these measurements to a central node. The central node or fusion center may often be the cognitive base station. Cooperative sensing techniques can be classified appropriately based on the type of decision metric used at the fusion center.

Manuscript received September 21, 2012; revised January 31, 2013; accepted April 09, 2013. Date of publication April 26, 2013; date of current version June 25, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Walaa Hamouda. This research was supported in part by DTRA Grant HDTRA1-13-1-0030 and NSF Grant CNS-1262329. A preliminary version of this work with a subset of the results was presented at the Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, September 2010.

H. Ganapathy and C. Caramanis are with the Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712 USA (e-mail: harishg@utexas.edu; constantine@utexas.edu).

L. Ying is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85286 USA (e-mail: lei.ying.2@asu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2013.2260336

We focus on one of the simplest schemes that relies on energy detection. Most prior work on energy-detection-based cooperative sensing (see [10]–[14] and references therein), the signal is typically a single binary hypothesis modeling a system with one primary user that is either ON or OFF. In cooperative sensing, the measurements from each cognitive sensor node are typically linearly weighted and combined to form a decision statistic that is in turn compared against a threshold to produce a binary output. The paper by Ganesan and Li [10] constitutes one of the earliest contributions that establishes the gains in detection (of a single primary user) probability due to cooperation in the presence of channel fading. Peh *et al.* [11] and Lee [14] study the tradeoff between sensing time and throughput of the cognitive network since longer sensing times lead to higher primary detection probabilities but lower secondary throughputs. Quan *et al.* [12] further optimize the linear weighted combiner studied by Ganesan and Li [10] by choosing weights that maximize the detection probability. While the above authors designed *soft-combining* systems where the sensors typically report their measurements to the fusion center without any processing, Li *et al.* [13] consider *hard-combining* where local decisions are made at each sensor, and design a voting rule to produce the final binary decision. Finally, Mishra *et al.* [15] compare the performance of hard-/soft-combining and study the effects of correlated fading on the detection probability. We refer the reader to recent survey papers by Akyildiz *et al.* [16] and Yucek [17] for a more comprehensive list of references along with further discussion on the other types of cooperative spectrum sensing such as cyclostationary feature detection.

B. Our Contributions

In almost all of the reviewed literature, the standard approach to transmit the measurements from the sensor nodes to the fusion center has been to employ orthogonal channels. This means that the control bandwidth demanded by a standard cooperative sensing scheme scales linearly in the number of sensing nodes. This immediately places a significant burden on the bandwidth requirements of the network and might be impractical in some scenarios as recognized by Akyildiz *et al.* [16]. In this work, we focus on reducing the amount of control bandwidth required by considering the structure that may often exist in the *dynamics* of the observed signals. Thus, we immediately build on and extend past models by adding a temporal dimension to the observed signal and by considering multiple discrete signals instead of a single binary hypothesis.

The key idea that we exploit in this paper is the following: while the observed signals at a snapshot in time in general live in some arbitrary high dimension, often, there is a time-scale separation between the sensing time window and the behavior of the observed process. For example, TV transmitters would turn ON/OFF on a significantly slower time-scale (e.g., in the order of minutes) than sensor measurement windows (e.g., in the order of milliseconds). Under these circumstances, *the dynamics of the observed signal vector are likely to be sparse*. It has long been known, and recently popularized under the name of compressive sampling, that whereas N linear measurements are required to reconstruct a vector or signal in $\mathbb{R}^N$, if it is S-sparse (i.e., it has S non-zero coefficients) then under appropriate conditions on the measurements, $\mathcal{O}(S \log N)$ are enough [19], [20], [22]. By developing similar tools, and applying them on the dynamics, rather than the signal directly, we show that with greatly-reduced control bandwidth, our algorithms perform close to the linear control bandwidth case. The performance is measured in terms of the similarity between the recovered signal and the true signal.

To the best of our knowledge, this is the first work in the domain of wireless networks[1] to exploit *sparsity in the dynamics* of the observed signal. As this is likely much more prevalent than sparsity in the actual trajectory of the signal (of course, if the trajectory is sparse, then so are the dynamics) we expect this high-level idea to find broad application. More concretely, the main contributions in this paper are as follows:

(1) A first (to the best of our knowledge) application of compressive sampling to reduce the control bandwidth in a cooperative sensing system.

(2) A proof that path-loss matrices satisfy the *null space property* thereby allowing for efficient acquisition or sensing of the observed signal using standard convex programs such as ℓ_1-norm minimization and Lasso [25]. The proof technique is novel since path-loss matrices contain entries that have non-zero mean and are not independent, a scenario that has not been dealt with extensively in past literature on compressive sampling[2].

(3) Simulation results that establish the competitive performance of our algorithm in comparison to the full control bandwidth case.

The rest of this paper is organized as follows. In Section II, we introduce the system model for the cooperative sensing network under consideration. In Section III, we discuss the compressive sampling algorithm that enables cooperative sensing using significantly-reduced control overhead. We establish the "goodness" of path-loss-induced sensing matrices in Section IV. The complete cooperative sensing algorithm is presented in Section V. Simulation results establishing the competitive performance of the algorithm are contained in Section VI. Concluding remarks are made in Section VII.

Notation: x_{ij} denotes element (i, j) of matrix $\mathbf{X}$ while x_i denotes element i of vector $\mathbf{x}$. $(.)^T$ is the transpose operator. For $\mathbf{x} \in \mathbb{R}^N, \mathbf{x}_{\mathcal{A}}, \mathcal{A} \subseteq \{1, 2, \ldots, N\}$ denotes the vector $\mathbf{x}$ restricted to the entries in $\mathcal{A}$. For $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathbf{X}_{\mathcal{A}}$, $\mathcal{A} \subseteq \{1, 2, \ldots, m\}$ denotes the sub-matrix of $\mathbf{X}$ formed by the rows contained in $\mathcal{A}$. Finally, $\|\cdot\|_p$ is the p-norm operator on vectors.

II. System Model

We consider a network with N_s sensing nodes and a single fusion center operating in slotted-time. We introduce the necessary measurement and signal models below.

Signal generation: We assume the signal of interest is being generated by multiple physical entities (e.g., TV transmitters) that are each dropped uniformly on a circle of radius r_p centered at the origin. There are a total of N_p such signal generators that

[1] Sparse changes in the dynamics of the signal have been used in other areas such as image processing to achieve video compression [18].

[2] We do not use the terminology "compressed sensing" to avoid confusion with "cooperative sensing".

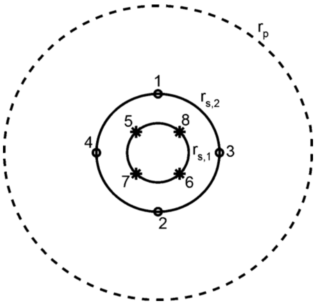


Fig. 1. Network with signal generators (not shown) uniformly distributed on the blue circle of radius r_p. There are $N_s = 8$ sensing nodes in the network equally-divided across two circles ($n_c = 2$) of radii $r_{s,1}$ and $r_{s,2}$ respectively. This gives rise to partitions $\mathcal{C}_1 = \{1, 2, 3, 4\}$ and $\mathcal{C}_2 = \{5, 6, 7, 8\}$. The sensing nodes are equally-spaced on each circle as shown.

are located at points $\{(r_p, \theta_j)\}_{j=1}^{N_p}$ where $\theta_j \sim U[0, 2\pi]$, $\forall j$. The vector signal emitted at time t is denoted by $\sqrt{s(t)} = [\sqrt{s_1(t)}\ \sqrt{s_2(t)} \ldots \sqrt{s_{N_p}(t)}]$ where $s_m(t)$ corresponds to the power emitted by physical entity or signal generator m and comes from some finite, discrete set.

Spatial distribution of sensing nodes : There are $N_s,\ N_s \geq N_p$ sensing nodes that are placed on a collection of n_c circles of radii $\{r_{s,1}, r_{s,2}, \ldots, r_{s,n_c}\}$, where N_s is such that $\frac{N_s}{n_c}$ is *even* . Circle c contains $\frac{N_s}{n_c}$ interfering receivers located at fixed points $\{(r_{s,c}, \theta_i)\}_{i=1}^{N_s}$ that are equally-spaced $\theta_i = \frac{2\pi n_c}{N_s}$, $i = 0, 1, \ldots, \frac{N_s}{n_c} - 1$, as shown in Fig. 1. We note that this would roughly be the case when N_s becomes large and the users are uniformly distributed on this collection of circles. The fusion center is located at some arbitrary point on the xy-plane.

The above spatial distribution model affords us analytical tractability. In the simulations section, we show that the proposed algorithms work even under a more general spatial model where both the sensing and signal nodes are scattered uniformly on a regular square grid. For the sake of the analysis, we also partition the sensing nodes according to the circle they belong to thus creating n_c partitions $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_c}\}$ such that $\bigcup_{i=1}^{n_c} \mathcal{C}_i = \{1, 2, \ldots, N_s\}$ and $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for $i \neq j$. Within each circle, the nodes are numbered or ordered in diametrically opposite pairs as shown in Fig. 1, a labelling rule that is feasible since $\frac{N_s}{n_c}$ is even. In other words, all pairs $(j, j+1) \in \mathcal{C}_i,\ j\ odd$, will correspond to a pair of diametrically opposite nodes on circle $\mathcal{C}_i$.

Channel model: The distance between the n-th signal generator at location (r_p, θ_n) and m-th sensing node at (r_m, θ_m) is given by $d_{mn} = \sqrt{r_m^2 + r_n^2 - 2r_m r_n \cos(\theta_m - \theta_n)}$, which induces the following path-loss gain[3]

$$\lambda_{mn} = \frac{1}{K + d_{mn}^2},\ K > 0 \tag{1}$$

between the same. This model is an approximation of the free space path-loss model (with path-loss coefficient two) [21], a choice that affords us analytical tractability while compromising very little on modeling accuracy. The composite channel gain between n-th signal generator at location (r_n, θ_n) and m-th sensing node at (r_p, θ_m) is given by

$$h_{mn}(t) = g_{mn}(t)\sqrt{\lambda_{mn}}, \tag{2}$$

where $g_{mn}(t) \sim \mathcal{CN}(0, 1)$ models small-scale Rayleigh fading. Distances do not change as a function of time since sensor nodes are deterministically placed and signal nodes are assumed fixed, once drawn from a uniform distribution.

Sensor measurement model: We adopt the sensing model in [11], [15], where each sensor collects a set of samples over a window of size W time slots. In particular, at time t sensor node m receives the signal

$$r_m(t) = \sum_n h_{mn}(t)\sqrt{s_n(t)} + v_m(t),$$

where $v_m(t)$ is additive noise. Each sensor then forms the following *measurement* over a W-window of samples,

$$z_m(k) = \frac{1}{W} \sum_{t=kW+1}^{(k+1)W} |r_m(t)|^2 - \mathbb{E}[|v_m(t)|^2]. \tag{3}$$

The sensors then send their measurements to the fusion center via orthogonal error-free control channels. The fusion center is charged with the task of recovering the observed signal $\{s_n(t)\}_n$. In this work, we do not explicitly account for the time incurred in transmitting measurements to the fusion center and for subsequent actions such as data transmissions by the secondary network in the case of cognitive radio, as we are primarily interested in reducing the control bandwidth between the sensors and the fusion center. Thus, the time index k iterates only across sensing windows.

Sparse dynamics: Due to mismatches between the time-scale of the signal generators and the sensor sample collection period, we assume that the vector $\mathbf{s}(t)$ exhibits the following behavior. The signal $s_n(t)$ remains constant through the k-th collection window $kW + 1 \leq t < (k+1)W$ and is denoted by $s_n(k)$. Furthermore, only a subset of signals change between the k-th and the $(k+1)$-th collection window, i.e., $|\mathbf{s}(k+1) - \mathbf{s}(k)|$ is a sparse vector. This model is effective when the signal values change on a slower time-scale in relation to the sensing timeline. This is indeed what one typically expects for systems that exhibit a time-scale separation between the sensing network and the signal emitting process. Under the sparse dynamics assumption and for W sufficiently large, we invoke the *Law of Large Numbers (LLN)* and write the measurements in (3) as

$$\mathbf{z}(k) = \mathbf{\Lambda}\mathbf{s}(k) + \boldsymbol{\beta}(k), \tag{4}$$

where $\lambda_{mn} = \mathbb{E}[|h_{mn}(t)|^2] = \frac{1}{K + d_{mn}^2}$ from (2); $\beta_m(k)$ is a noise term that models the inaccuracies of the LLN over finite averaging windows. It is well-known from the *Central Limit Theorem* that $\mathbb{E}[|\beta_m(k)|^2] = \theta\left(\frac{1}{W}\right)$, $\forall m$. The term $\boldsymbol{\beta}(k)$ is referred to as LLN-noise through the remainder of this paper. Next, in the context of the measurement model in (4), we introduce a signal recovery algorithm that will be developed further through the course of this paper and that exploits the sparse structure of the observed dynamics.

Recovery algorithm: We assume that a subset $Q \subseteq \{1, 2, \ldots, N_s\}$ of the measurements $\mathbf{z}_Q(k)$ are transmitted through $|Q|$ orthogonal control channels to the fusion center at the end of sensing window k. The fusion center must recover the signal $\mathbf{s}(k)$. The naïve approach to recovering $\mathbf{s}(k)$ would be to

[3] The analysis in this paper can be extended to cover other path-loss models.

transmit *all* the measured values, i.e., set $Q = \{1, 2, \ldots, N_s\}$, to the fusion center[4]. The above naïve solution would consume a control bandwidth of $\mathcal{O}(N_s)$. If the process $\mathbf{s}(k)$ is completely general, there is little that can be done to remedy this problem, and partial feedback (of only a subset of $\mathbf{z}$) will necessarily result in degraded performance. However, as discussed in the introduction, for networks where only a subset of signal generators (e.g., TV transmitters) change their state between two adjacent sensing windows, we show that it is possible to reduce the control bandwidth. Using ideas from subset selection and compressive sampling, the next section considers how sparse dynamics can be exploited in order to achieve near-optimal performance while reducing the bandwidth consumption.

III. Recovering Dynamics Through Compressive Sampling

In this section, we propose a compressive sampling approach to efficiently recover the signal $\mathbf{s}(k)$. Before we discuss the recovery algorithm, we take a short diversion into the topic of compressive sampling.

A. Compressive Sampling

The topic of compressive sampling has received tremendous interest in the recent years [19], [20], [22]. The theory essentially states that one can recover sparse data exactly, given an under-determined system of equations. The generic problem is the following: Given a signal $\mathbf{x} \in \mathbb{R}^N$, one receives $q \ll N$ linear, potentially noisy measurements: $\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{w}$. Here, $\mathbf{M} \in \mathbb{R}^{q \times N}$ encodes the measurement matrix and $\mathbf{w} \in \mathbb{R}^q$ denotes additive noise, usually of bounded norm.

For a general vector $\mathbf{x} \in \mathbb{R}^N$, N independent measurements are required to hope to reconstruct $\mathbf{x}$. When $q < N$, the problem therefore is under-determined. If $\mathbf{x}$ is sparse, however, in some settings the problem is no longer under-determined, and can be solved exactly by considering standard convex programs such as

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N} : \|\mathbf{M}\mathbf{x} - \mathbf{y}\|_2^2 - \xi\|\mathbf{x}\|_1 \text{ for some chosen } \xi > 0,$$

in the noisy case and

$$\begin{aligned} \arg\min: \quad & \|\mathbf{x}\|_1 \\ & \mathbf{M}\mathbf{x} = \mathbf{y} \end{aligned} \tag{5}$$

in the noiseless case. The former is the so-called *Lasso* [25] formulation for model selection (subset selection) while the latter ℓ_1-norm minimization problem is called *Basis Pursuit*. Many such results have appeared in the literature, e.g., [19], [20], [22], [29], under the umbrella of compressive sampling. Indeed, the results are attractive from an algorithmic perspective since the convex relaxation is easily solvable, with computation time that scales gracefully as the size of the problem increases, allowing the efficient solution of very large problems. The above convex programs succeed as long as the linear equations, or measurements, satisfy a property called *Null Space Property* (NSP), which essentially amounts to the statement that there are no very sparse vectors in the null-space of the measurement matrix $\mathbf{M}$. The theoretical connections between Lasso and Basis Pursuit have been well-analyzed by authors such as Tropp [24].

B. The Recovery Algorithm

Returning to our problem, we define $\Delta\mathbf{s}(k) = \mathbf{s}(k) - \mathbf{s}(k-1)$ and apply the model selection paradigm outlined above, to *the dynamics vector* $\Delta\mathbf{s}(k)$ rather than the signal vector itself. We can assume that at some initial time k_0, $\mathbf{s}(k_0)$ is known. At time k, we "query" a subset of sensors $\mathcal{Q}$ and receive measurements $\mathbf{z}_{\mathcal{Q}}(k) = \mathbf{\Lambda}_{\mathcal{Q}}\mathbf{s}(k) + \boldsymbol{\beta}(k)$. We can then construct the difference in measurements

$$\begin{aligned} \Delta\mathbf{z}_{\mathcal{Q}}(k) &= \mathbf{z}_{\mathcal{Q}}(k) - \mathbf{z}_{\mathcal{Q}}(k-1) \\ &= \mathbf{\Lambda}_{\mathcal{Q}}\Delta\mathbf{s}(k) + \boldsymbol{\beta}(k) - \boldsymbol{\beta}(k-1). \end{aligned}$$

Since the left hand side, $\Delta\mathbf{z}_{\mathcal{Q}}(k)$, is known, this falls precisely into the sparse recovery paradigm developed above. More concretely, let Q be the subset of queried users with size $|Q| = q,\ q\ even, q \le N_s$. Then, Q is chosen according to the following algorithm:

Algorithm 1: Protocol to choose query set Q of size $|Q| = q$

```
Set $Q = \emptyset$, $i = c = 0$.
while $c < \frac{q}{2}$ do
    Choose any pair of diametrically opposite receivers
    $(j, j+1)$ from circle $i+1$, i.e., $j \in \mathcal{C}_{i+1},\ j\ odd$.
    $Q = Q \cup \{j, j+1\}$.
    Set $\mathcal{C}_{i+1} = \mathcal{C}_{i+1} \setminus \{j, j+1\}$
    Increment $c = c + 1$ and $i = (i+1)(\bmod\ n_c)$.
end while
```

Note that the output query set is not unique since the choice of node-pairs is left open. Let $\mathcal{Q}$ be the set of all possible output query sets from Algorithm 1 above. For the example in Fig. 1, one possible output query set for $q = 4$ is $Q = \{1, 2, 7, 8\}$. It is necessary for a query set Q to be selected in this way for the sake of analytical tractability. As with the spatial distribution model, we adopt a more general querying model in our simulations section. Under such a querying model, it is of immediate interest to determine the smallest query size q (or control bandwidth) that the system requires in order to recover $\Delta\mathbf{s}(k)$ reliably using

$$\arg\min_{\mathbf{x}\in\mathbb{R}^{N_p}} : \|\mathbf{\Lambda}_{\mathcal{Q}}\mathbf{x} - \Delta\mathbf{z}_{\mathcal{Q}}(k)\|_2^2 - \xi\|\mathbf{x}\|_1. \tag{6}$$

As mentioned earlier, in this work, we do not consider the number of bits required to communicate $\mathbf{z}_Q(k)$ reliably to the fusion center as we are interested primarily in the scaling behavior of the control bandwidth.

Compressive sampling theory states that recovery of any S-sparse vector through Lasso or Basis Pursuit is possible in a noiseless setting if and only if the *sensing matrix* $\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$ satisfies the NSP [26] of order S. This property is defined in the next section. *We note that in our application, the sensing matrix is provided by the channel as opposed to traditional compressive sampling where the designer is often allowed to choose a convenient sensing mechanism*. In the next section, we present the main result of this paper, which states that

[4] The fusion center may recover the signal using standard tools such as least-squares.

path-loss matrices $\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$ make for good sensing matrices in the noiseless case, i.e., when $W \to \infty$. Recovery results that are proved for Basis Pursuit, which, in the absence of noise, carry over to Lasso [24]. While the theory is developed for the noiseless case with $W \to \infty$, our simulations consider finite averaging windows and demonstrate that sparse recovery still remains effective in this setting.

IV. NSP of Path-Loss Matrices

In this section, we establish that path-loss matrices $\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$ satisfy the *Null Space Property* (which will be defined shortly) when the control bandwidth obeys $q = \mathcal{O}(S\log N_p)$. Lemma 4, Lemma 5 along with Theorem 4 constitute the main results in this section.

A. Preliminaries

Let the support set of $\mathbf{x}$ be denoted by $\mathcal{S}$. A vector $\mathbf{x}$ is S-sparse if $|\mathcal{S}| \leq S$. We define the null space property from Gribonval *et al.* [27]. Given a matrix $\mathbf{M}$, let $\mathcal{N}(\mathbf{M})$ denote its null space.

Definition (Null space Property): A matrix $\mathbf{M}$ satisfies the null space property of order S if for all subsets $\mathcal{S} \subseteq \{1, 2, \ldots, N\}$ with $|\mathcal{S}| \leq S$, the following holds $\|\mathbf{v}_{\mathcal{S}}\|_1 \leq \|\mathbf{v}_{\mathcal{S}^c}\|_1,\ \forall \mathbf{v} \in \mathcal{N}(\mathbf{M}) \setminus \mathbf{0}$, where $\mathcal{S}^c = \{1, 2, \ldots, N\} \setminus \mathcal{S}$. Based on this property, the following recovery result [27] has appeared both implicitly and explicitly in works such as [26], [28].

Theorem 1: [27] Let $\mathbf{M} \in \mathbb{R}^{q \times N}$. Then, any S-sparse vector may be recovered by solving (5) iff $\mathbf{M}$ satisfies the NSP of order S. □

The NSP is typically quite difficult to prove directly leading to the development of sufficient conditions that are easier to establish. One such sufficient condition is the *Restricted Isometry Property* [29] that has become quite popular in recent years and is defined below.

Definition (Restricted Isometry Property): A $q \times N$ matrix $\mathbf{M}$ satisfies the Restricted Isometry Property (RIP) of order p if there exists $\epsilon_p(\mathbf{M}) \in (0, 1)$ such that for all $\mathbf{x} \in \mathbb{R}^N$,

$$(1 - \epsilon_p(\mathbf{M}))\|\mathbf{x}_{\mathcal{T}}\|_2^2 \leq \|\mathbf{x}_{\mathcal{T}}^T(\mathbf{M}^T)_{\mathcal{T}}\|_2^2 \leq (1 + \epsilon_p(\mathbf{M}))\|\mathbf{x}_{\mathcal{T}}\|_2^2 \tag{7}$$

holds for all sets $\mathcal{T}$ with $|\mathcal{T}| \leq p$. Here, $(\mathbf{M}^T)_{\mathcal{T}}$ is the submatrix of $\mathbf{M}^T$ formed by rows in $\mathcal{T}$.

Here, $\epsilon_p(\mathbf{M})$ is called the *Restricted Isometric Constant* of $\mathbf{M}$. The RIP essentially requires that all $q \times |\mathcal{T}|$ sub-matrices of $\mathbf{M}$ be well-conditioned. Under such a conditioning, perfect recovery of $\mathbf{x}$ is possible as stated in the following theorem.

Theorem 2: [34],[35] Let $\mathbf{M} \in \mathbb{R}^{q \times N}$. If $\mathbf{M}$ satisfies the RIP with $\epsilon_{2S}(\mathbf{M}) \leq 2\frac{(3-\sqrt{2})}{7} \approx 0.4531$, then every S-sparse vector $\mathbf{x} \in \mathbb{R}^N$ is the solution to the ℓ_1-norm minimization problem in (5). □

Thus, the RIP with a sufficiently small constant immediately implies the NSP in the context of ℓ_1-recovery. The approach we use to prove "goodness" of path-loss matrices $\mathbf{\Lambda}_Q$ is motivated by the following observation. In general, the null space of a product of two matrices $\mathbf{NM}$ contains the null space of $\mathbf{M}$ and therefore if $\mathbf{NM}$ satisfies the NSP, so does $\mathbf{M}$. This allows us to study the class of *linearly-processed* path-loss matrices $\mathbf{A}_Q = \mathbf{B}\,\underbrace{\mathbf{W}\mathbf{\Lambda}_Q}_{\mathbf{G}}$ where

$$\mathbf{W} = \mathrm{diag}\{\underbrace{\mathbf{J}\,\mathbf{J}\ldots\mathbf{J}}_{\frac{q}{2}\ \text{times}}\},\ \mathbf{J} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \tag{8}$$

and $\mathrm{diag}\{\cdot\}$ is the standard block-diagonal operator;

$$\mathbf{B} = \mathrm{diag}\left\{\frac{\beta_1}{\sqrt{\mathrm{Var}\{g_{11}\}}} \ldots \frac{\beta_q}{\sqrt{\mathrm{Var}\{g_{q1}\}}}\right\} \tag{9}$$

with $\beta_i \sim \text{Bernoulli}\left(\frac{1}{2}\right),\ \forall i$ and independent across i. The Bernoulli random variables have support $\{\pm 1\}$. We focus our attention on establishing the recovery properties of $\mathbf{A}_Q$ rather than $\mathbf{\Lambda}_Q$. We show that $\mathbf{A}_Q$ satisfies the RIP with $q = \mathcal{O}(S\log N_p)$ measurements and hence the NSP. The transformation $\mathbf{W}$ essentially subtracts rows of $\mathbf{\Lambda}_Q$ corresponding to *diametrically opposite* pairs of sensors in the *same* partition. Thus, the dimension of $\mathbf{G}$ is still $q \times N_p$. The transformation $\mathbf{B}$ weights and adds adjacent rows of $\mathbf{G}$.

According to our spatial distribution model, since the positions of the sensing nodes are fixed, the columns of $\mathbf{\Lambda}_Q$ (and hence $\mathbf{G}$) become stochastically independent since each interfering transmitter is independently thrown. At this point, we will rely heavily on recent results from Vershynin [30] and Adamczyk *et al.* [23] that deal with sensing matrices containing independent columns. Before we reproduce the RIP result [23], [30] for matrices with independent columns, we present a primer on sub-gaussian and sub-exponential random variables along with some useful results from non-asymptotic matrix theory.

B. Useful Concentration Inequalities

We refer the reader to the tutorial paper by Vershynin [30] for a great introduction to non-asymptotic matrix theory. Lemmas 1–3 below are well-known past results that are summarized in this paper [30]. The proofs are not reproduced due to lack of space.

Lemma 1: Let u be a random variable. The following properties are equivalent with parameters $K_i > 0,\ i = 1, 2, 3, 4$, differing from each other by at most an absolute constant factor.

(i) Tails: $\Pr(|u| > \gamma) \leq \exp(1 - \frac{\gamma^2}{K_2})$ for all $\gamma > 0$,

(ii) Moments: $(\mathbb{E}\left[|u|^p\right])^{\frac{1}{p}} \leq K_2\sqrt{p}$ for all $p \geq 1$,

(iii) Super-exponential moment: $\mathbb{E}\left[\exp\left(\frac{u^2}{K_3}\right)\right] \leq \exp(1)$. Moreover, if $\mathbb{E}[u] = 0$ then properties (i)–(iii) are also equivalent to the following property:

(iv) Moment generating function: $\mathbb{E}\left[\exp\left(u\gamma\right)\right] \leq \exp(\gamma^2 K_4)$ for all $\gamma \in \mathbb{R}$.

□

A random variable that satisfies the above property is called a *sub-gaussian* random variable. Such random variables are often characterized by the ψ_2-norm[5], which is defined as $\|u\|_{\psi_2} = \sup_{p \geq 1} \frac{(\mathbb{E}[|u|^p])^{\frac{1}{p}}}{\sqrt{p}}$. It follows that if the ψ_2-norm of u is finite, then u is a sub-gaussian random variable with $\|u\|_{\psi_2} = K_2$. This is in fact the case for bounded random variables with symmetric distributions.

[5] Alternate definitions of this norm have been adopted (such as in [31]) that are all equivalent within a constant factor.

Lemma 2: Let u be a symmetrically distributed, bounded random variable with $|u| \leq M,\ M > 0$. Then, u is a sub-gaussian random variable with $\|u\|_{\psi_2} \leq cM^2,\ c > 0$. □

In higher dimensions, a random vector $\mathbf{u}$ of dimension N is called sub-gaussian if $\mathbf{u}^T\mathbf{x}$ is sub-gaussian for every $\mathbf{x} \in \mathbb{R}^N$.

Lemma 3: Let $\{u_i\}_{i=1}^p$ be a collection of independent, zero-mean, sub-gaussian random variables. Then, $\mathbf{u}$ is a sub-gaussian random vector with $\|\mathbf{u}\|_{\psi_2} = C \max_i\ \|u_i\|_{\psi_2}$ for some $C > 0$. □

Before we move on to proving the RIP (hence NSP) for matrix $\mathbf{A}_Q$, we require one more definition. A random vector $\mathbf{u}$ of dimension N is called *isotropic* if $\mathbb{E}[|\mathbf{u}^T\mathbf{x}|^2] = \|\mathbf{x}\|^2$ for all $\mathbf{x} \in \mathbb{R}^N$.

C. NSP of Linearly-processed Path-loss Matrices $\mathbf{A}_q$

We reproduce the recent RIP result [30], [31] concerning matrices with independent columns. We refer the reader to [30], [31] for the proof.

Theorem 3: Let $\mathbf{M} = [\mathbf{m}_1\, \mathbf{m}_2 \ldots \mathbf{m}_N]$ be a $q \times N$ random matrix whose columns are independent, isotropic and sub-gaussian with $\psi_{max,m} = \max_{i\in\{1,\ldots,N\}}\ \|\mathbf{m}_i\|_{\psi_2}$. Furthermore, let the columns satisfy $\|\mathbf{m}_i\|^2 = q$ almost surely for all i. Then, the normalized matrix $\frac{1}{\sqrt{q}}\mathbf{M}$ is such that if $q \geq C_{\psi_{max,m}}\varepsilon^{-2}S\log\left(\frac{\exp(1)N}{S}\right)$ for some $\varepsilon > 0$, then $\varepsilon_p\left(\frac{1}{\sqrt{q}}\mathbf{M}\right) \leq \varepsilon$ with probability at least $1 - 2\exp(-c_{\psi_{max,m}}\varepsilon^2 q)$. Here, $c_{\psi_{max,m}}$ and $C_{\psi_{max,m}}$ are positive scalars that depend only the worst-case sub-gaussian norm $\psi_{max,m}$. □

As mentioned earlier, the channel matrix $\mathbf{\Lambda} = [\boldsymbol{\lambda}_1\, \boldsymbol{\lambda}_2 \ldots \boldsymbol{\lambda}_{N_p}]$ contains independent columns since the positions of the sensing nodes are fixed. However, each column contains entries that are not centered, not isotropic and that are highly coupled. This is because all entries in $\boldsymbol{\lambda}_i$ are now completely determined by the position of signal generator i. For this reason, it is not immediately clear whether the columns are sub-gaussian.

To finally prove the NSP of $\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$, we show that after left-multiplication by matrices $\mathbf{B}$ and $\mathbf{W}$, the entries of matrix $\mathbf{A}_Q$ are sub-gaussian and satisfy the sufficient conditions in Theorem 3. The following lemmas and theorem constitute the main theoretical results of this paper.

Lemma 4: The matrix $\mathbf{A}_Q = \mathbf{B}\mathbf{W}\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$ of size $q \times N$ contains independent, isotropic, centered, sub-gaussian columns.

Proof: See Appendix A. □

Lemma 5: For matrix $\mathbf{A}_Q = \mathbf{B}\mathbf{W}\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$ of size $q \times N$, we have that $\|\mathbf{a}_i\|^2 = q$ almost surely.

Proof: See Appendix B. □

Theorem 4: $\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$ satisfies the NSP of order S almost surely when $q = \mathcal{O}(S\log N_p)$.

Proof: From Lemma 4, $\mathbf{A}_Q$ contains independent, isotropic, centered, sub-gaussian columns. From Lemma 5, $\mathbf{A}_Q$ is such that $\|\mathbf{a}_i\|^2 = q$ almost surely. Thus, the matrix $\mathbf{A}_Q$ satisfies the sufficient conditions of Theorem 3, which implies that $\mathbf{A}_Q$ satisfies the RIP and hence the NSP when $q = \mathcal{O}(S\log N_p)$. Since the null space of $\mathbf{A}_Q$ contains the null space of $\mathbf{\Lambda}_Q,\ Q \in \mathcal{Q}$, the result follows. □

Next, we discuss how this sparse recovery algorithm is incorporated into the development of a complete cooperative sensing algorithm. In particular, since we are estimating dynamics rather than the signal itself and due to the presence of LLN-noise, the real possibility of *error propagation* arises. This has not been heretofore addressed in the literature, to the best of our knowledge. In the next section, the algorithm introduces explicit steps to control this. The simulations in Section VI demonstrate the effectiveness of our approach.

V. Complete Cooperative Sensing Algorithm

The essential conclusion of the previous section is that the signal vector $\mathbf{s}(k)$ can be recovered by acquiring $\mathcal{O}(S\log N_p)$ noise-free measurements if the dynamics of $\mathbf{s}(k)$ are S-sparse.

The complete sensing algorithm is presented in Algorithm 2. Recall that we do not consider the actions taken post-sensing and the time taken (equiv. throughput *per* control channel) to transmit the measurements to the fusion center as we are only interested in scaling properties in this work. The algorithm operates in three modes; in *dynamics* mode, the algorithm retrieves signals dynamics incurring logarithmic control bandwidth and then estimates the recovery error. If the estimated recovery error is too high, the algorithm switches to *partial reset* mode in the next sensing window or slot where the fusion center retrieves the signal directly while incurring linear bandwidth. Following partial reset, the system returns to dynamics mode. This operation is overlaid by a periodic (with a pre-set period) *reset* mode where again the fusion center again retrieves the signal directly incurring linear bandwidth.

Algorithm 2: Sensing algorithm

```
Primary inputs: sensing window sizes $W$, number of sensors $N_s$, number of signal emitters $N_p$, sparsity factor $S$, error reset period $T_{err} > 0$
Initialize mode $[0] =$ "dynamics"
**for** each $k = 1, 2, \ldots,$ **do**
  **if** $k(\mathrm{mod}\ T_{err}) \not\equiv 0$ **then**
    During time slots $kW \leq t+1 < (k+1)W$, each sensor samples the signal of interest $W$ times and averages the same to obtain $z_m(k)$ as per (4).
    **if** mode[k-1] = "dynamics" **then**
      The fusion center queries $q = \Theta(S\log N_p)$ sensors indexed by set $\mathcal{Q}$ according to Algorithm 1 and computes the change in received signal levels $\Delta\mathbf{z}_Q(k)$.
      *Dynamics mode*: Once acquired, the fusion center estimates the change in measurements $\Delta\mathbf{s}(k) = \mathbf{s}(k) - \mathbf{s}(k-1)$ by solving the following optimization problem:
          $\Delta\tilde{\mathbf{s}}(k) = \arg\min_{\mathbf{x}\in\mathbb{R}^{N_p}} : \|\mathbf{\Lambda}_Q\mathbf{x} - \Delta\mathbf{z}_Q(k)\|_2^2 - \xi\|\mathbf{x}\|_1.$   (10)
      The fusion center sets $\tilde{\mathbf{s}}(k) = [\phi^\nu(\tilde{s}_1(k-1) + \Delta\tilde{s}_1(k)) \ldots \phi^\nu(\tilde{s}_{N_s}(k-1) + \Delta\tilde{s}_{N_s}(k))]^T$, where $\phi^\nu : \mathbb{R} \to \mathbb{R}$ is a suitably-chosen thresholding function with $\nu$ that is used to recover discrete sparse signals in noisy environments.
      **if** $T_{err} > 2$ **then**
        The fusion center then queries and acquires extra measurements from $q'$ additional sensors $Q' \subseteq$
```

```
            $\{1, 2, \ldots, N_s\} \setminus \mathcal{Q},\ |\mathcal{Q}'| = q'$. Next, the fusion
            center computes
                $e = \frac{1}{q'}\|\mathbf{\Lambda}_{Q'}\tilde{\mathbf{s}}(k) - \mathbf{z}_{Q'}(k)\|_1,$        (11)
            which essentially represents the average recovery
            error on the extra measurements.
12:         if $e \leq \epsilon$, i.e., the estimate is accurate, then
13:             Set $\tilde{\mathbf{s}}(k)$ to be the final estimate of the signal.
14:             Set $\mathrm{mode[k]}$ = "dynamics".
15:         else
16:             Set $\mathrm{mode[k]}$ = "partial reset".
17:         end if
18:       end if
19:     else if $\mathrm{mode[k-1]}$ = "partial reset" then
20:       Partial Reset Mode: The fusion center queries all $N_s$
          sensors and uses the measurements to directly
          recover the final estimate $\tilde{\mathbf{s}}(k)$ by solving
              $\mathbf{u}(k) = \arg\min_{\mathbf{x}\in\mathbb{R}^{N_p}} : \|\mathbf{\Lambda}_Q\mathbf{x} - \mathbf{z}_Q(k)\|_2^2 - \xi\|\mathbf{x}\|_1$        (12)
          and setting $\tilde{\mathbf{s}}(k) = [\phi^\nu(u_1(k)) \ldots \phi^\nu(u_{N_s}(k))]^T$.
21:       Set $\mathrm{mode[k]}$ = "dynamics".
22:     end if
23:   else if $k(\mathrm{mod}\ T_{err}) \equiv 0$ then
24:     During time slots $kW \leq t < (k+1)W$, each sensor
        forms $z_m(k)$ as per (4).
25:     Full Reset Mode: The fusion center then queries all
        sensors and uses the $N_s$ measurements to directly
        recover the final estimate $\tilde{\mathbf{s}}(k)$ by solving (12).
26:     Set $\mathrm{mode[k]}$ = "dynamics".
27:   end if
28: end for
```

In a typical sensing slot, the Lasso procedure in Step 8 recovers $\Delta\tilde{\mathbf{s}}(k)$. The recovery is almost surely exact when $\Delta\mathbf{s}(k)$ is S-sparse and when W is sufficiently large. However, as this might not always be the case in reality, error propagation is a critical aspect that needs to be managed when operating on the dynamics of the signal. The following remarks describe the steps taken by the algorithm to address the same:

- The system selects a parameter T_{err} that represents the number of sensing slots after which the fusion center must query *all* sensors and recover the absolute signal as opposed to the dynamics.
- During typical operation, i.e., when $k(\mathrm{mod}\ T_{err}) \not\equiv 0$, we estimate the accuracy of $\tilde{\mathbf{s}}(k)$ in Step 11 immediately following compressed sampling in Step 9. The technique is commonly referred to as cross-validation[32]. If the estimated error is larger than the threshold ϵ, we query*all* remaining N_s sensors and recover the absolute signal in the next time slot.
- Once more, due to the sources of noise described above, the vectors retrieved by Lasso in ((10) will almost never be truly sparse. Thus, we introduce a thresholding function $\phi^\nu(\cdot)$ to recover a discrete signal.

1	6			
2				
3				
4				
5				25

Fig. 2. Illustrative example of grid model used in simulations with $N = 25$.

VI. Numerical Results

In this section, we evaluate the performance of the proposed sensing algorithm using numerical experiments.

A. Simulation Setup

The simulation setting is described in the following:

Network geography- We consider a square region with side length 1 kilometer. We partition this square area into $N = 49$ micro-cells, each of dimension 143 m × 143 m, which is sufficiently small to ensure at most one signal generator or sensor per micro-cell. We index the micro-cells by i $(1 \leq i \leq N)$, counted column-wise as shown in Fig. 2. Each micro-cell contains one *potential* signal generator that is located at the center of the square micro-cell. The sensors are positioned one per micro-cell, i.e, $N_s = N = 49$.

Signal dimension and channel state knowledge- By virtue of the above geographical model where we have "discretized" the locations of the sensors and signal generators, the fusion center may now pre-compute $\mathbf{\Lambda}$ based on the grid. This removes the need for real-time knowledge about the positions of the signal generators. However, this comes at a price, namely, an increase in the dimension of the signal from N_p to N. Thus, the value $s_n(k),\ n = 1, 2, \ldots, N$, is non-zero if a signal generator is *present and active* at micro-cell n on the grid.

Signal activity- In our simulations, we assume that the signals are binary-valued, i.e., $s_n(k) \in \{0, P\},\ P > 0, \forall n, k$. The activity of each signal generator or micro-cell $s_n(k)$ is modeled as a Markovian ON-OFF process where the transition probability from 0 to P is $\eta_0 = \eta = 0.1$ and from P to 0 is $\eta_1 = \eta = 0.1$. The probability that the signal changes state from sensing window to the next can be computed as $\frac{2\eta_0\eta_1}{(\eta_0+\eta_1)} = \frac{2\times0.1\times0.1}{(0.1+0.1)} = 0.1$. Thus, on average 10% or roughly $S_{ave} = 49 \times 0.1 \approx 5$ signals change their state, which reflects the sparse dynamics. As per our system model, we assume that P is known and thus, we are only interested in estimating the support at time k. Henceforth, the true support at time k be given by $\mathcal{P}(k) = \{n : n = 1, 2, \ldots, N,\ s_n(k) = P\}$, while the estimated support by Algorithm 2 is given by $\hat{\mathcal{P}}(k) = \{n : n = 1, 2, \ldots, N,\ \hat{s}_n(k) = P\}$.

Post-recovery thresholding- We use a simple per-element thresholding function

$$\phi^\nu(x) = \begin{cases} \mathrm{sign}(x)P, & |x| \geq \nu \\ 0, & \text{else} \end{cases} \tag{13}$$

to recover the actual sparse entries themselves. Note that we have to account for negative values for x since $\mathbf{u}(k)$ represents differences in (10). Based on empirical evidence, we select $\nu = \frac{P}{2}$ as this value is seen to maximize the performance metric that is discussed next.

Performance metric- To measure the signal recovery accuracy of our algorithm, we use set similarity. Given two sets $\mathcal{A}, \mathcal{B} \subseteq \{1, 2, \ldots, N\}$, the similarity is defined as

$$\psi(\mathcal{A}, \mathcal{B}) = \frac{1}{N} \left[|\mathcal{A} \cap \mathcal{B}| + |\mathcal{A}^c \cap \mathcal{B}^c| \right]. \tag{14}$$

In the context of our sensing problem, i.e., setting $\mathcal{A} = \mathcal{P}(k)$ and $\mathcal{B} = \hat{\mathcal{P}}(k)$, we see that $\psi(\cdot)$ essentially counts the fraction of elements that are correctly recovered at time k and may alternatively be interpreted as one minus the normalized (by $\frac{1}{N}$) hamming distance between the recovered and the true signal. Now since we are dealing with a stochastic, time-varying system, we are interested in the long-term average similarity that is given by

$$\bar{\psi} = \frac{1}{T} \sum_{k=1}^{T} \psi(\mathcal{P}(k), \hat{\mathcal{P}}(k)), \tag{15}$$

where T is the length of our simulation counted as the total number of sensing windows. We select $T = 10000$ for our simulations.

Other algorithm parameters- For the initial set of experiments with a sparsity factor of 10%, the number of measurements for recovery and error measurement are set to $q = 1.5 S_{ave} \log N = \frac{2\eta^2}{(\eta_0 + \eta_1)} \log N = \eta \log N = 29$ and $q' = 5$ with associated query sets[6] $Q = \{32, 40, 22, 34, 35, 6, 3, 16, 11, 30, 45, 33, 7, 38, 42, 28, 17, 41, 47, 14, 46, 8, 5, 48, 29, 21, 25, 37, 31\}$ and $Q' = \{49, 27, 26, 43, 19\}$. Later, when we explore the performance of our algorithm under different sparsity factors, we choose a different number of measurements. The regularization parameter is set to $\xi = 0.01$. The remaining parameters such as W, ε and T_{err} are varied in our experiments.

Control bandwidth- Note that the parameters q, q', ε, T_{err} and N have a direct bearing on the bandwidth consumed by our algorithm. To analyze the bandwidth consumed in Algorithm 2, we basically count the fraction of time spendt in each of the modes in Steps 8, 20 and 25. For a total communication duration of T sensing windows, i.e., $k = 1, 2, \ldots, T$, let the time spent in dynamics mode be given by T_d. Then, the bandwidth consumed, is given by

$$B = \begin{cases} \frac{T_d}{T} q + N(1 - \frac{T_d}{T}), & T_{err} = 2 \\ \frac{T_d}{T}(q + q') + N(1 - \frac{T_d}{T}), & \text{otherwise} \end{cases} \tag{16}$$

and the savings over full (linear) control bandwidth can be expressed as $\Delta B = 1 - \frac{B}{N}$. The control bandwidth, in addition to the similarity measure in (14), is an important evaluation metric, one that forms the core motivation of this paper.

Through the remainder of this section, we explore the many interesting tradeoffs that are inherent in our proposed sensing design in the context of the above parameters.

B. Simulation Results

Having described our simulation setup in detail, we now proceed to analyze the performance of our algorithm. Since the “optimal” naïve algorithm in itself does not achieve perfect recovery due to LLN noise, we adopt the following methodology for comparison. We fix a sensing window size W and determine the similarity performance of the naïve algorithm with full linear bandwidth, i.e., set $T_{err} = 1$ in Algorithm 2 with $q = N$. We then run and tune Algorithm 2 over the parameters ε and $T_{err} > 1$ while fixing the same sensing window size W. We record the similarity and bandwidth performance of Algorithm 2 over the entire space of parameters (ε, T_{err}). We repeat this process over many sensing window sizes $W \in \{50, 100, 250, 500, 750, 2000, 5000, 10000\}$. We show that with careful tuning, Algorithm 2 consumes considerably lesser bandwidth while achieving an average similarity that is within 5% of the optimal full linear bandwidth approach in the worst-case.

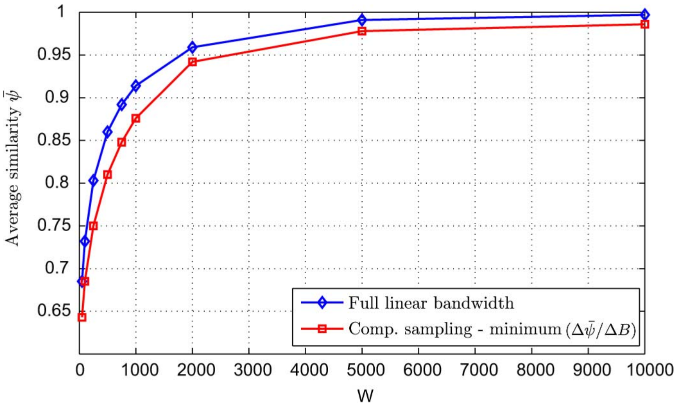


Fig. 3. Average similarity $\bar{\psi}$ versus window size W with sparsity factor of 10%.

In Fig. 3, we plot the similarity performance of the linear bandwidth algorithm over many sizes of sensing windows for a sparsity factor of 10% or $\eta = 0.1$. We see that at around $W = 10000$, we achieve near-perfect recovery. The exact similarity values are available in Table I. Next, we proceed to tune Algorithm 2 with the goal of achieving an average similarity that is “close” to the naïve algorithm while consuming lesser bandwidth. In particular, we study the accuracy-bandwidth performance of our approach while varying the error threshold ε and the reset period T_{err} over the space $(\varepsilon, T_{err}) \subseteq \{0.1P, 0.5P, P\} \times \{2, 3, 4, 5, 8\}$. This amounts to a total of $3 \times 6 = 18$ configurations per window size W. The results are tabulated in Table I. Note that in Table I, we have introduced the following additional metric that succinctly summarizes the dual requirements of accuracy and bandwidth reduction:

$$\frac{\Delta\bar{\psi}}{\Delta B}, \tag{17}$$

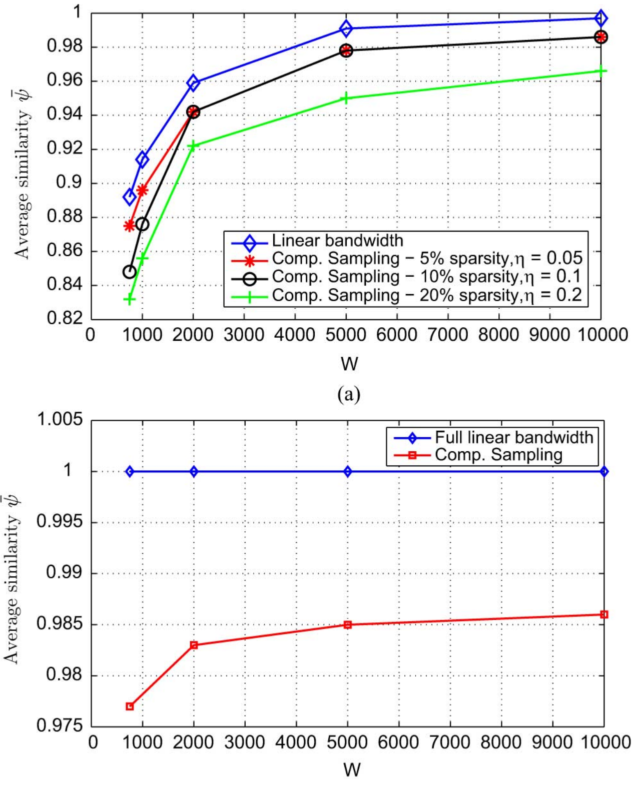
where $\Delta\bar{\psi} > 0$ is the difference in long-term average similarities between the naïve and compressive-sampling-based algorithm. To compare the two approaches, for each W, we select the configuration that minimizes (17), thereby balancing accuracy vs. bandwidth savings, as our competing configuration. As an example, the smallest value of (17) for $W = 50$ is highlighted in Table I. Thus, we can construct a similarity vs. window size curve for our compressive sampling algorithm in Fig. 3.

[6] The query set was generated uniformly at random. Our experiments showed that the exact choice of sets did not affect the results much for the simulation setup under consideration.

TABLE I
ACCURACY-BANDWIDTH RESULTS FOR ALGORITHM 2

	$W = 50$	$W = 100$	$W = 250$
$(\varepsilon/P, T_{err})$	$(\bar{\psi},\ \Delta B,\ \Delta\bar{\psi}/\Delta B)$	$(\bar{\psi},\ \Delta B)$	$(\bar{\psi},\ \Delta B)$
(0.1,1)	(0.685,0,-)	(0.732,0)	(0.803,0)
(0.1,2)	**(0.643,0.204,0.204)**	(0.685,0.204)	(0.75,0.204)
(0.1,3)	(0.658,0.102,0.265)	(0.701,0.102)	(0.769,0.102)
(0.1,4)	(0.643,0.153,0.275)	(0.685,0.153)	(0.755,0.153)
(0.1,5)	(0.652,0.123,0.268)	(0.694,0.122)	(0.765,0.123)
(0.1,8)	(0.645,0.153,0.261)	(0.684,0.153)	(0.75,0.153)
(0.5,3)	(0.656,0.107,0.271)	(0.697,0.111)	(0.761,0.118)
(0.5,4)	(0.642,0.153,0.281)	(0.683,0.154)	(0.751,0.155)
(0.5,5)	(0.649,0.13,0.277)	(0.688,0.134)	(0.754,0.143)
(0.5,8)	(0.643,0.154,0.273)	(0.681,0.156)	(0.743,0.16)
(1,3)	(0.641,0.144,0.306)	(0.677,0.154)	(0.737,0.168)
(1,4)	(0.632,0.17,0.312)	(0.665,0.179)	(0.723,0.19)
(1,5)	(0.631,0.17,0.318)	(0.661,0.181)	(0.714,0.197)
(1,8)	(0.626,0.183,0.322)	(0.652,0.195)	(0.699,0.21)
	$W = 500$	$W = 750$	$W = 1000$
$(\varepsilon/P, T_{err})$	$(\bar{\psi},\ \Delta B)$	$(\bar{\psi},\ \Delta B)$	$(\bar{\psi},\ \Delta B)$
(0.1,1)	(0.86,0)	(0.892,0)	(0.914,0)
(0.1,2)	(0.81,0.204)	(0.848,0.153)	(0.876,0.204)
(0.1,3)	(0.827,0.102)	(0.864,0.103)	(0.889,0.104)
(0.1,4)	(0.811,0.153)	(0.849,0.153)	(0.872,0.153)
(0.1,5)	(0.82,0.123)	(0.856,0.124)	(0.881,0.124)
(0.1,8)	(0.81,0.153)	(0.846,0.153)	(0.875,0.153)
(0.5,3)	(0.814,0.13)	(0.848,0.14)	(0.871,0.147)
(0.5,4)	(0.803,0.161)	(0.837,0.165)	(0.861,0.168)
(0.5,5)	(0.806,0.151)	(0.836,0.161)	(0.859,0.169)
(0.5,8)	(0.798,0.167)	(0.829,0.174)	(0.854,0.182)
(1,3)	(0.79,0.177)	(0.827,0.183)	(0.851,0.189)
(1,4)	(0.773,0.198)	(0.806,0.204)	(0.831,0.208)
(1,5)	(0.765,0.204)	(0.794,0.214)	(0.82,0.218)
(1,8)	(0.744,0.221)	(0.774,0.227)	(0.795,0.236)
	$W = 2000$	$W = 5000$	$W = 10000$
$(\varepsilon/P, T_{err})$	$(\bar{\psi},\ \Delta B)$	$(\bar{\psi},\ \Delta B)$	$(\bar{\psi},\ \Delta B)$
(0.1,1)	(0.959,0)	(0.991,0)	(0.997,0)
(0.1,2)	(0.942,0.204)	(0.978,0.204)	(0.986,0.204)
(0.1,3)	(0.947,0.11)	(0.98,0.165)	(0.986,0.178)
(0.1,4)	(0.943,0.155)	(0.977,0.21)	(0.984,0.217)
(0.1,5)	(0.943,0.133)	(0.977,0.195)	(0.984,0.209)
(0.1,8)	(0.942,0.155)	(0.975,0.216)	(0.983,0.228)
(0.5,3)	(0.916,0.166)	(0.963,0.188)	(0.979,0.195)
(0.5,4)	(0.911,0.185)	(0.956,0.206)	(0.973,0.215)
(0.5,5)	(0.907,0.189)	(0.95,0.216)	(0.969,0.228)
(0.5,8)	(0.901,0.203)	(0.94,0.231)	(0.96,0.243)
(1,3)	(0.903,0.197)	(0.959,0.203)	(0.977,0.204)
(1,4)	(0.889,0.219)	(0.946,0.227)	(0.969,0.229)
(1,5)	(0.876,0.231)	(0.936,0.241)	(0.962,0.244)
(1,8)	(0.851,0.248)	(0.912,0.261)	(0.943,0.265)

From Fig. 3, we see that as the window size increases, the performance of the compressive sampling algorithm approaches that of full linear bandwidth. Furthermore, we see that over all $W \in \{50, 100, 250, 500, 750, 2000, 5000, 10000\}$, we are at worst within 5% of over the naïve algorithm, albeit while consuming significantly less bandwidth. In particular, from Fig. 3, we observe that at moderately large values of W such as $W = 2000$, we can achieve within 2% of the optimal accuracy while consuming at least 15% lesser bandwidth. As we move to larger values $W = 10000$, we are now within 1% of the optimal accuracy while consuming at least 20% less bandwidth. The exact bandwidth values can be found in Table I. Recognizing that different applications may give rise to various alternatives to (17), e.g., bandwidth savings might be more less important than accuracy, one may also consider the configuration that yields the second smallest value of (17) as the competing configuration.



(a)

(b)

Fig. 4. Average similarity $\bar{\psi}$ versus window size W for (a) different sparsity factors and (b) an alternate signal model.

From Table I, we see that the bandwidth savings when considering the second smallest values are around 10%–15%.

Next, we study the performance of our algorithm under different sparsity factors and signal models. In Fig. 4(a), we plot similarity results for different sparsity factors of 5% and 20%, i.e., $\eta \in \{0.05, 0.2\}$. The number of measurements are set to $(q, q') = (15, 5)$ for the 5% case and $(q, q') = (29, 5)$ for the 20% case. As with the earlier experiment, for each W, we record the accuracy-bandwidth values of the (ε, T_{err})-configuration that minimizes (17). For the 5% case, by reducing the number of observations, we are now able to save around 34% in bandwidth (uniformly across all W) while maintaining roughly the same level of similarity performance as compared to the case with the 10% sparsity factor. On the other hand, with a sparsity factor of 20%, by retaining the same number of measurements at $(q, q') = (29, 5)$, we lose at most 5% in average similarity across the window sizes of interest. The bandwidth reduction in this case is around 20% uniformly across all W.

In our final experiment, we modify the signal activity model to the following: We choose five $(\sim 10\%)$ signal generators uniformly at random and flip their states. Note that this model ensures that the sparsity factor is *exactly* 10% whereas the previous model guaranteed this sparsity factor in an expected sense. The results of this experiment are presented in Fig. 4(b). We clearly see that the linear bandwidth achieves perfect recovery as early as $W = 750$. The performance of compressive sampling is significantly better too, with a bandwidth saving of around 20% uniformly across all W. The results in Figs. 3 and 4(b) clearly establish the impact of the sparse dynamics model on the performance of the recovery algorithm.

Thus, by exploiting the structure in the emitted signal, we are able to achieve significant savings in control bandwidth over the naïve algorithm while not overly compromising on recovery accuracy. Recognizing that the number of parameters that control the algorithm is quite large, we believe that with more extensive tuning/optimization of the various components, e.g., error measurement budget q', further savings might be possible, especially as the scale or dimension of the problem grows. Finally, we note that the proposed compressive sampling approach has been applied to (even) more general spatial distribution models such as one where both the sensor and signal nodes are thrown uniformly at random on a square area, in the absence of a grid. The results are as promising but not presented here due to lack of space.

VII. Conclusions

In this paper, we apply ideas from the field of compressive sampling to reduce the control bandwidth in cooperative sensing systems. We are particularly interested in settings where the dynamics of the signals being observed are sparse. This is indeed what one typically expects for systems that exhibit a time-scale separation between the sensing network and the signal emitting process. We prove theoretical RIP-/NSP-based recovery results for path-loss matrices in networks with circular geometries. We then develop a complete cooperative sensing algorithm that addresses key issues such as error propagation that are certain to arise in practice. We demonstrate that our approach can provide significant bandwidth savings over the naïve linear bandwidth case under more general spatial distributions. Extending the theory to cover these general spatial distributions remains an open problem that deserves attention.

Appendix A
Proof of Lemma 4

Recall that $\mathbf{\Lambda}_{\mathcal{C}_i}$ represents the sub-matrix of $\mathbf{\Lambda}_Q$ containing the rows specified in $\mathcal{C}_i$. The entries $\{\lambda_{ij}\}$ of sub-matrix $\mathbf{\Lambda}_{\mathcal{C}_i}$ are identically distributed. This follows from the observation that any two sensor nodes on the circle of radius $r_{s,i}$ will perceive the same distribution of signal nodes that are uniformly distributed on a circle. Recall that the transformation $\mathbf{G} = \mathbf{W}\mathbf{\Lambda}_Q$ essentially subtracts rows of $\mathbf{\Lambda}_Q$ corresponding to diametrically opposite users on each circle. Thus, the columns of $\mathbf{G}$ are independent and all entries are centered.

The next step is to show that the entries $g_{ij} = \lambda_{ij} - \lambda_{(i+1)j}$ are symmetric. This is does not follow immediately from the fact that λ_{ij} and $\lambda_{(i+1)j}$ are identically distributed when indices i and $i+1$ come from the same partition since they are not independent. Hence, we appeal to the concept of exchangeable random variables, which is defined below for the special case of a pair of random variables.

Two random variables X and Y are called exchangeable if their joint cumulative distribution function (cdf) is symmetric, i.e., if $F_{X,Y}(x,y) = F_{X,Y}(y,x)$.It is known the difference of two identically distributed, exchangeable random variables is indeed symmetric [33]. Thus, we only need to establish this property for the case when λ_{ij} and $\lambda_{(i+1)j}$ come from the same circle, say $\mathcal{C}_c$, since the definition of $\mathbf{G}$ in (8) precludes any other possibility. To establish that λ_{ij} and $\lambda_{(i+1)j}$ are exchangeable, we compute the joint cdf $F_{\lambda_{ij},\lambda_{(i+1)j}}(x,y)$, when i and $i+1$ come from the same circle c, as follows

$$\begin{aligned} F_{\lambda_{ij},\lambda_{(i+1)j}}(x,y) &= \Pr(\lambda_{ij} \le x, \lambda_{(i+1)j} \le y) \\ &= \Pr(d_{ij} \ge \frac{1}{x} - K, d_{(i+1)j} \ge \frac{1}{y} - K) \\ &= \Pr\Bigg(d\left(\theta_i,\theta_j\right) \ge \left[\frac{1}{x} - K\right]_+, d\left(\theta_{i+1}\theta_j\right) \ge \\ &\qquad \left[\frac{1}{y} - K\right]_+\Bigg) \text{ where}[u]_+ = \max\{u,0\} \\ &= \frac{1}{2\pi} V(\mathcal{R}(\theta_j,\theta_i,x,y)), \end{aligned}$$

where we have modified the distance function as

$$\begin{aligned} d^2\left(\theta_m,\theta_n\right) = r_{s,1}^2 + r_p^2 \\ -2r_p r_{s,1} \cos\left(\theta_m - \theta_n\right),\ i = 1,\ldots,\alpha(k), \end{aligned} \tag{18}$$

for notational convenience through the remainder of the proof. The set $\mathcal{R}(\theta_j,\theta_i,x,y)$ is defined as $\mathcal{R}(\theta_j,\theta_i,x,y) = \left\{\theta_j : d\left(\theta_j,\theta_i\right) \ge \left[\frac{1}{x} - K\right]_+,\ d^2\left(\theta_i,\theta_+\frac{2\pi q}{N_s}\right) \ge \left[\frac{1}{y} - K\right]_+\right\}$ and $V(\mathcal{A})$ denotes the volume of the set $\mathcal{A}$. The volume of set $\mathcal{R}(\theta_j,\theta_i,x,y)$ can be expressed as a sum of the volumes corresponding to smaller sets. To that effect, we define $\mathcal{R}_s(\theta,a) = \left\{\theta_j : d\left(\theta_j,\theta\right) \le \left[\frac{1}{a} - K\right]_+\right\}$ and can thus write

$$\begin{aligned} &\mathcal{V}(\mathcal{R}(\theta_j,\theta_i,x,y)) \\ &\quad = 2\pi r_p - \left[\mathcal{V}(\mathcal{R}_s(\theta_i,x)) + \mathcal{V}\left(\mathcal{R}_s\left(\theta_i + \frac{2\pi q}{N_s}, y\right)\right)\right. \\ &\qquad\qquad \left. -\mathcal{V}\left(\mathcal{R}_s\left(\theta_i,x\right) \cap \mathcal{R}_s\left(\theta_i + \frac{2\pi q}{N_s}, y\right)\right)\right]. \end{aligned}$$

From this characterization, we can immediately conclude that $\mathcal{V}(\mathcal{R}(r_p,\theta_j,r_{s,c},\theta_i,x,y)) = \mathcal{V}(\mathcal{R}(\theta_j,\theta_i,y,x))$, which gives our the desired result that λ_{ij} and $\lambda_{(i+1)j}$ are exchangeable. It is clear from the above arguments that exchangeabilty essentially follows from the uniform distribution of the signal emitters.

The symmetry of g_{ij} is crucial for our next step where we argue that columns of $\mathbf{A}_Q = \mathbf{BG}$ remain independent (denoted by $\perp$). By definition, $a_{ij} = \frac{1}{\sqrt{Var\{g_{i1}\}}}\beta_i g_{ij}$. Thus, to prove that the columns of $\mathbf{A}_Q = \left[\mathbf{a}_1\ \mathbf{a}_2 \ldots \mathbf{a}_{N_p}\right]$ are independent, we need to show that $a_{ij} \perp \mathbf{a}_k$ for any arbitrary i,j and $k \neq j$. Since the Bernoulli random variables are independent across rows and since $g_{ij} \perp g_{mk},\ k \neq j,\ m \neq i$, we clearly have that $a_{ij} \perp a_{mk},\ k \neq j,\ m \neq i$. Thus, we only need to establish that $a_{ij} \perp a_{ik}$ for $k \neq j$ or equivalently that $\beta_i g_{ij} \perp \beta_i g_{ik}$. But this follows from the symmetry of g_{ij} which means that knowledge of $\beta_i g_{ij}$ reveals no information about the random variable β_i. Thus, the columns of $\mathbf{A}_Q$ are independent. In addition, they are identically distributed and hence we can now focus on studying the properties of column $\mathbf{a}_1$ without loss of generality.

The random vector $\mathbf{a}_1$ is isotropic since $\mathbb{E}[a_{i1}^2] = \frac{1}{\mathbb{E}[g_{i1}^2]}\mathbb{E}\left[g_{i1}^2\right] = 1$ and $\mathbb{E}[a_{i1}a_{k1}] =$

$\frac{1}{\sqrt{\mathbb{E}[g_{i1}^2]}}\frac{1}{\sqrt{\mathbb{E}[g_{k1}^2]}}\mathbb{E}\left[\beta_i\beta_k\right]\mathbb{E}\left[g_{i1}g_{k1}\right] = 0$ for $i \neq k$, implying that $\mathbb{E}[|\mathbf{a}_1^T\mathbf{x}|^2] = \sum_i \mathbb{E}[a_{i1}^2]x_i^2 + \sum_{i\neq k}\mathbb{E}[a_{i1}a_{k1}]x_ix_k = \|\mathbf{x}\|^2$. To prove that $\mathbf{a}_1$ is sub-gaussian, we first condition on the position of the first signal emitter (r_p, θ_1). This will allow us to apply Lemma 1 and Lemma 2 since g_{i1} is now completely known thereby making $\mathbf{a}_{i1}$ a collection of independent random variables. The elements a_{i1} are symmetric and bounded with $|a_{i1}| \leq \frac{|g_{i1}|}{\sqrt{\mathbb{E}[g_{i1}^2]}}$ when conditioned on position (r_p, θ_1).

Lemma 6: There exists $M_p > 0,\ p = 0, 2, \ldots, q-1$ such that $\mathbb{E}[g_{i1}^2] = M_p,$ for $p\frac{N_s}{q} < i \leq (p+1)\frac{N_s}{q},\ \forall k$.

Proof: By definition, the distribution of g_{i1} for $p\frac{N_s}{q} < i \leq (p+1)\frac{N_s}{q}$ depends on the distance between the corresponding diametrically opposite sensors on circle $(p+1)$ along with the distribution of the first interfering user. Since this distance always remains the same independent of i and k, the result follows. □

Hence, by Lemma 2 and since $|g_{i1}| \leq 1$, a_{i1} is sub-gaussian with $\|a_{i1}\|_{\psi_2|(r_p,\theta_1)} \leq \frac{1}{M_*^2}$ where $M_* = \min_{p=0,\ldots,q-1} M_p$. Here, we have introduced notation $\|\cdot\|_{\psi_2|(r_p,\theta_1)}$ to indicate explicitly that we have conditioned on the location of the first interfering user. Now, from Lemma 1, we conclude that $\mathbf{a}_1$ is a sub-gaussian vector when conditioned on the location of the first signal node with $\|\mathbf{a}_1\|_{\psi_2|(r_p,\theta_1)} \leq 2C,\ C > 0$. However, since the sub-gaussian norm $\|\mathbf{a}_1\|_{\psi_2|(r_p,\theta_1)}$ computed above is independent of (r_p, θ_1), this implies that $\mathbf{a}_1$ is a sub-gaussian vector with $\|\mathbf{a}_1\|_{\psi_2} \leq 2C$. This can be seen by applying the Law of Total Probability to the definition of sub-gaussianity in Lemma 1. □

Appendix B
Proof of Lemma 5

To show almost-sure convergence of the norm of $\mathbf{a}_1$, we need to prove that $\exists c^* > 0$ such that

$$\Pr\left(\lim_{k\to\infty}\left|\frac{1}{k}\sum_{i=1}^{k} a_{i1}^2 - c^*\right| = 0\right) = \Pr\left(\lim_{k\to\infty}\left|\frac{1}{k}\sum_{i=1}^{k} g_{i1}^2 - c^*\right| = 0\right) = 1$$

where the probability is computed over the random location $(r_p, \theta),\ \theta \sim U[0, 2\pi]$. We instead prove the following more general statement that $\exists c^* > 0$ such that

$$\lim_{k\to\infty}\left|\frac{1}{k}\sum_{i=1}^{k}\frac{g_{i1}^2}{\mathbb{E}[g_{i1}^2]} - c^*\right| = 0 \text{ for all } (r_p, \theta),\ \theta \in [0, 2\pi]. \tag{19}$$

Note that (19) is a completely deterministic convergence statement in contrast to the earlier probabilistic statement. From the proposed query protocol in Algorithm 1 , we see that the number of sensors selected for feedback is a monotonically increasing function of k for all circles. This means that we can study the convergence of the norm for any one circle (a sub-vector of $\mathbf{a}_1$) and draw conclusions about the norm concentration of the entire vector $\mathbf{a}_1$. Consider the first partition of sensors on circle $\mathcal{C}_1$ and let the size of this partition be $\alpha(k)$. Then, we see that $\alpha(k) \to \infty$ as $k \to \infty$ and hence, we can shift our focus to proving the property

$$\lim_{k\to\infty}\left|\frac{1}{\alpha(k)}\sum_{i=1}^{\alpha(k)}\frac{g_{i1}^2}{\mathbb{E}[g_{i1}^2]} - c^*\right| = 0,\ \forall(r_p, \theta_1),\ \theta_1 \in [0, 2\pi], \tag{20}$$

where in fact $c^* = 1$. Recall that the squared-distance between a signal generator located at (r_p, θ_1) and sensor node i on $\mathcal{C}_1$ is given by $d^2(\theta_i, \theta_1)$ where $d(\cdot)$ is defined in (18) and where $\theta_i = \frac{2\pi iq}{k}$. Now let $f(x, y) = \left(\frac{1}{K+x} - \frac{1}{K+y}\right)^2,\ x, y > 0$ where K is given in the definition (1). Then, g_{i1}^2 can equivalently be expressed as $g_{i1}^2 = f\left(d^2(\theta_i, \theta_1), d^2\left(\theta_i + \frac{\pi}{2}, \theta_1\right)\right)$. Now, we proceed to prove (20) by observing that

$$\begin{aligned} &\lim_{k\to\infty}\tfrac{1}{\alpha(k)}\textstyle\sum_{i=1}^{\alpha(k)} g_{i1}^2 \\ =\ &\lim_{k\to\infty}\tfrac{1}{\alpha(k)}\textstyle\sum_{i=1}^{\alpha(k)} f\left(d^2(\theta_i, \theta_1), d^2\left(\theta_i + \tfrac{2\pi q}{k}, \theta_1\right)\right)\cdot \\ =\ &\tfrac{1}{2\pi}\textstyle\int_0^{2\pi} f\left(d^2(\theta_i, \theta_1), d^2\left(\theta_i + \tfrac{\pi}{2}, \theta_1\right)\right) d\theta_i. \end{aligned} \tag{21}$$

This conclusion follows since the expression on the left is essentially the Riemann sum of the integral on the right. This means that for any given $\varepsilon > 0,\ \exists\ k_\varepsilon$ such that when $k \geq k_\varepsilon$, we have that $\left|\frac{1}{\alpha(k_\varepsilon)}\sum_{i=1}^{\alpha(k_\varepsilon)} g_{i1}^2 - \frac{1}{2\pi}\int_0^{2\pi} f\left(d^2(\theta_i, \theta), d^2\left(\theta_i + \frac{\pi}{2}, \theta\right)\right) d\theta_i\right| < \varepsilon$. Next, we calculate

$$\mathbb{E}[g_{i1}^2] = \frac{1}{2\pi}\int_0^{2\pi} f\left(d^2(\theta_i, \theta), d^2\left(\theta_i + \frac{\pi}{2}, \theta\right)\right) d\theta. \tag{22}$$

By substituting (18) in (21) and (22), we see that the variable of integration may be switched and hence

$$\begin{aligned} &= \frac{1}{2\pi}\int_0^{2\pi} f\left(d^2(\theta_i, \theta), d^2\left(\theta_i + \frac{\pi}{2}, \theta\right)\right) d\theta \\ &= \frac{1}{2\pi}\int_0^{2\pi} f\left(d^2(\theta_i, \theta), d^2\left(\theta_i + \frac{\pi}{2}, \theta\right)\right) d\theta_i, \end{aligned} \tag{23}$$

which implies that $\left|\frac{1}{\alpha(k_\varepsilon)}\sum_{i=1}^{\alpha(k_\varepsilon)} g_{i1}^2 - \frac{1}{2\pi}\int_0^{2\pi} f\left(d^2(\theta_i, \theta), d^2\left(\theta_i + \frac{\pi}{2}, \theta\right)\right) d\theta_i\right| = \left|\frac{1}{\alpha(k_\varepsilon)}\sum_{i=1}^{\alpha(k_\varepsilon)} g_{i1}^2 - \mathbb{E}[g_{i1}^2]\right| < \varepsilon$. Convergence in (20) follows since $\mathbb{E}[g_{i1}^2]$ is bounded below by Lemma 6 and the fact that

$$\begin{aligned} \left|\frac{1}{\alpha(k)}\sum_{i=1}^{\alpha(k)}\frac{g_{i1}^2}{\mathbb{E}[g_{i1}^2]} - 1\right| &= \frac{1}{\mathbb{E}[g_{i1}^2]}\left|\frac{1}{\alpha(k)}\sum_{i=1}^{\alpha(k)} g_{i1}^2 - \mathbb{E}[g_{i1}^2]\right| \\ &< \frac{1}{\mathbb{E}[g_{i1}^2]}\varepsilon \text{ when } k \geq k_\varepsilon. \end{aligned}$$

□

References

[1] P. K. Varshney, *Distributed Detection and Sata Fusion*. New York, NY, USA: Springer-Verlag, 1997.

[2] J.-J. Xiao and Z.-Q. Luo, "Universal decentralized detection in a bandwidth-constrained sensor network," *IEEE Trans. Signal Proc.*, vol. 53, no. 8, pp. 2617–2624, Aug. 2005.

[3] M. K. Banavar, C. Tepedelenlioglu, and A. Spanias, "Estimation over fading channels with limited feedback using distributed sensing," *IEEE Trans. Signal. Proc.*, vol. 58, pp. 414–425, Jan. 2010.

[4] J.-F. Chamberland and V. V. Veeravalli, “Asymptotic results for decentralized detection in power constrained wireless sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 22, pp. 1007–1015, Aug. 2004.

[5] J. Mitola and G. Q. Maguire, “Cognitive radios: making software radios more personal,” *IEEE Pers. Commun.*, vol. 6, pp. 1318–1318, Aug. 1999.

[6] National Telecommunications and Information Administration FCC Frequency Allocation Chart 2003 [Online]. Available: http://www.ntia.doc.gov/osmhome/allochrt.pdf

[7] Federal Communications Commission Spectrum Policy Task Force Report of the Spectrum Efficiency Working Group, Technical Report 02–135, 2002.

[8] F. W. Seelig, “A description of the XG demonstrations at Fort A.P. Hill,” in *Proc. IEEE Int. Symp. New Frontiers Dyn. Spectrum Access Netw. (DySPAN)*, Apr. 2007, pp. 1–12.

[9] “FCC frees up vacant TV airwaves for ‘Super-Wi-Fi’ technologies,” FCC [Online]. Available: http://www.fcc.gov

[10] G. Ganesan and Y. Li, “Cooperative spectrum sensing in cognitive radio networks,” in *Proc. IEEE Int. Symp. New Frontiers Dyn. Spectrum Access Netw. (DySPAN)*, Baltimore, MD, USA, Nov. 2005, pp. 137–143.

[11] E. C. Y. Peh, Y.-C. Lian, Y. L. Quan, and Y. Zeng, “Optimization of cooperative sensing in cognitive radio networks: A sensing-throughput tradeoff view,” *IEEE Trans. Veh. Technol.*, vol. 58, pp. 5294–5299, Nov. 2009.

[12] Z. Quan, S. Cui, and A. H. Sayed, “Optimal linear cooperation for spectrum sensing in cognitive radio networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 2, pp. 28–40, Jun. 2008.

[13] W. Zhang, R. K. Mallik, and K. B. Letaief, “Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks,” *IEEE Trans. Wireless Commun.*, vol. 8, pp. 5761–5766, 2009.

[14] W. Lee, “Optimal spectrum sensing framework for cognitive radio networks,” *IEEE Trans. Wireless Commun.*, vol. 7, pp. 3845–3857, Oct. 2008.

[15] S. M. Mishra, A. Sahai, and R. W. Brodersen, “Cooperative sensing among cognitive radios,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Istanbul, Turkey, Jun. 2006, pp. 1658–1663.

[16] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, “Cooperative spectrum sensing in cognitive radio networks: A survey,” *Elsevier J. Physical Commun.*, vol. 4, pp. 40–62, Mar. 2011.

[17] T. Yucek, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE Commun. Surveys Tuts.*, vol. 11, pp. 116–130, 2009.

[18] I. E. G. Richardson*, H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. Hoboken, NJ, USA: Wiley, 2003.

[19] E. Candes and M. Wakin, “An introduction to compressive sampling,” *IEEE Signal Process. Mag.*, vol. 25, pp. 21–30, Mar. 2008.

[20] R. Baraniuk, “Compressive sensing,” *IEEE Signal Process. Mag.*, vol. 24, pp. 118–121, Jul. 2007.

[21] S. T. Rappaport*, Wireless Communications–Principles and Practice*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002, ch. 4, pp. 138–139.

[22] D. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[23] R. Adamczak, A. E. Litvak, A. Pajor, and N. Tomczak-Jaegermann, “Restricted isometry property of matrices with independent columns and neighborly polytopes by random sampling,” arXiv:0904.4723v1, 2009.

[24] J. A. Tropp, “Just relax: Convex programming methods for identifying sparse signals,” *IEEE Trans. Inf. Theory*, vol. 51, pp. 1030–1051, Mar. 2006.

[25] R. Tibshirani, “Regression shrinkage and selection via the Lasso,” *J. Roy. Statist. Soc. Ser. B*, pp. 267–288, 1996.

[26] A. Cohen, W. Dahmen, and R. DeVore, “Compressed sensing and best k-term approximation,” *J. Amer. Math. Soc.*, vol. 22, pp. 211–231, 2009.

[27] R. Grobinval and M. Nielson, “Sparse representations in unions of bases,” *IEEE Trans. Inf. Theory*, vol. 49, pp. 3320–3325, Dec. 2003.

[28] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 2845–2862, 2001.

[29] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Trans. Inf. Theory*, vol. 51, pp. 4203–4215, Dec. 2005.

[30] R. Vershynin, Y. Eldar, and G. Kutyniok, “Introduction to the non-asymptotic analysis of random matrices,” in *Compressed Sensing: Theory and Application*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[31] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, “Reconstruction and subgaussian operators in asymptotic geometric analysis,” *Geom. Funct. Anal.*, pp. 1248–1282, 2007.

[32] R. Ward, “Compressed sensing with cross validation,” *IEEE Trans. Inf. Theory*, vol. 55, pp. 5773–5782, Dec. 2009.

[33] J. P. Romano and A. F. Siegel*, Counterexamples in Probability and Statistics*. Belmonth, CA, USA: Wadsworth, 1986.

[34] M. E. Davies and R. Gribonval, “Restricted isometry constants where ℓ^p sparse recovery can fail for $0 \ll p \leq 1$,” *IEEE Trans. Inf. Theory*, vol. 55, pp. 2203–2214, May 2009.

[35] S. Foucarta and M.-J. Lai, “Sparsest solutions of underdetermined linear systems via ℓ_q -minimization for $0 < q < 1$,” *Appl. Comput. Harmonic Anal.*, vol. 26, pp. 395–407, May 2009.

Harish Ganapathy (M’12) received the B.S. and M.S. degrees in electrical engineering from the State University of New York, Buffalo, NY, USA, in May 2003 and May 2005, respectively. He received his Ph.D. in electrical engineering from The University of Texas, Austin, TX, USA, in August 2011.

He is currently a Systems Engineer with Qualcomm Inc., Santa Clara, CA, USA. His research interests lie broadly in optimization as applied to wireless networks, including both physical layer and networking aspects.

Constantine Caramanis (M’06) received the Ph.D. in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2006.

He has since been on the faculty of the Department of Electrical and Computer Engineering at The University of Texas, Austin, TX, USA. His current research interests include robust and adaptable optimization, machine learning, and high-dimensional statistics, with applications to large scale networks.

Dr. Caramanis received the NSF CAREER Award in 2011.

Lei Ying (M’08) received the B.E. degree from Tsinghua University, Beijing, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2003 and 2007, respectively.

He was an Assistant Professor in the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA, from January 2008 to August 2012. He currently is an Associate Professor in the School of Electrical, Computer and Energy Engineering at Arizona State University, Tempe, AZ. His research interest is broadly in the area of stochastic networks, including big data and cloud computing, cyber security, P2P networks, social networks and wireless networks.

Dr. Ying is an Associate Editor of the IEEE/ACM Transactions on Networking. He won the Young Investigator Award from the Defense Threat Reduction Agency in 2009 and the NSF CAREER Award in 2010.