Fifty-third Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 29 - October 2, 2015

Compressive and Collaborative Mobile Sensing for Scalar Field Mapping in Robotic Networks

Minh T. Nguyen †, Hung M. La ‡ and Keith A. Teague †
† Oklahoma State University, Stillwater, OK 74078
‡ University of Nevada, Reno, NV 89557
Emails: {tuanminh.nguyen, keith.teague}@okstate.edu and hla@unr.edu

***Abstract*—In this paper, we propose a compressive and collaborative sensing (CCS) algorithm for distributed robotic networks to build scalar field map. A collaborative control law is utilized to steer the robots to move on the field while avoiding collision with each other and with obstacles. At each time instant, the robots collect, add measurements within their sensing range and exchange data with their neighbors to form compressive sensing (CS) measurements at each robot. After a certain times of moving and sampling, each robot can achieve that number of CS measurements to be able to reconstruct all sensory readings from the positions that the group of robots visited to build a scalar map. We further analyze and formulate the total communication power consumption associated with the number of robots, sensor communication range and provide suggestions for more energy saving.**

Keywords*—*Robotic networks, compressive sensing, scalar field mapping, collaborative control

I. Introduction

A. Motivation

Mobile sensor networks (MSNs) facilitate many existing application areas, such as monitoring temperature, humidity, acoustic, vibration or detecting target or special events (chemical leak, vehicle passing). The networks are combined from sensors, control algorithms and other dynamic factors which depend on specific purposes or application scenarios [1], [2].

Scalar field mapping is one of the most common applications for MSNs [3], [4]. In each map, a vector $\underline{\mathbf{X}} = [x_1\, x_2 \dots x_N]^T$ represents all the unknown sensory readings from N random positions in a sensing area needed to be observed. They are temperature, humidity, etc. and are often highly correlated. In general, the sensors are attached to mobile robots or vehicles and their movements follow a control algorithm [5]. All sensors need to move to visit the entire sensing area in order to build a full scalar map that cost a huge power consumption for either movements and communications [4], [6]. There exist many research results shown to improve control algorithms, physical sensors, data processing methods and communication connections for saving power for the networks [7], [8], [9], [10], [11].

Compressive sensing (CS) [12], [13], [14], [15] provides a novel framework to compress sparse or compressible signals that is perfectly applicable to the data collected from sensor networks. The technique offers to reconstruct all sensor readings based on a much smaller number of CS measurements ($\underline{\mathbf{Y}} = [y_1\, y_2 \dots y_M]^T$) compared to the number of positions (N). The CS measurements are collected as $\underline{\mathbf{Y}} = \Phi\underline{\mathbf{X}}$, where Φ is the measurement matrix, also called routing matrix in wireless networks that shows how a network collects data from a sensing area which is mentioned in this work.

In this paper, a team of robots are led by a collaborative control with virtual leaders to sample sparsely a sensing area. At instant time t, each robot adds all readings within its sensing range and shares the accumulated scalar value attached with the position indices to others. Each CS measurement is a sum or is collected from linear combinations of the sensory data measured by the group of robots. After M periods of time, at different positions, each robot achieves M CS measurements and reconstructs N sensory readings from the area. Different from other methods such as the consensus methods [16], [17] that usually require a connected network, our robot networks maybe disconnected sometimes due to the limited sensor communication range and obstacle avoidance on the field. This results in unequal CS measurements between each separated group of robots that does not affect the CS performance. Furthermore, the sensory readings transmitted between robots are added as a scalar value that saves a huge energy for communication compared to separated measurements transmitting in other methods [16], [17].

B. Related Work

Collaborative sensing in MSNs has been studied in [3], [4], [8], [9] with many applications such as target tracking, environmental mapping or monitoring. The distributed mobile robots are connected to share their information for specific purposes.

Upgrading from data collection methods using CS in wireless sensor networks [18], [19], [20], [21], recently, there have been some research studies that exploit the combination between the mobility of sensors and CS. Wang [22]

Minh T. Nguyen is a lecturer with Thai Nguyen University of Technology (TNUT), Thai Nguyen city, Vietnam. This work is supported by the Vietnamese Ministry of Education and Training (MOET) and School of Electrical and Computer Engineering (ECE), Oklahoma State University, University of Nevada, Reno, and NSF-NRI grant # 1426828.

978-1-5090-1824-6/15/$31.00 ©2015 IEEE

monitors vehicle networks based on CS to reduce the communication cost. Mostofi builds maps of obstacles/objects using robot networks [23] while the robots are deployed outside the areas. In [24], mobile sensors are deployed randomly in a sensing area to form CS measurements and to share the measurements with each other for CS recovery process at each sensor.

To our best knowledge, the existing work has not focused on exploring the collaboration and the mobility of robotic sensors to sample sparsely sensing areas and using CS to reconstruct all data at each distributed mobile robot. In summary, the contributions of our paper are

1) A new distributed compressive and collaborative sensing algorithm for MSNs to build scalar field maps ***at each mobile robot*** is proposed.
2) The total communication power consumptions for the network are analyzed and formulated.
3) Some important factors such as the number of mobile robots, the convergence time and the robot communication range are analyzed and simulated to minimize the network power consumption.

Unlike the existing consensus filter algorithms, our proposed framework does not require the sensor network to maintain connectivity all the time during its movement. This feature could enable real world applications of the complex field mapping and exploration where obstacles may exist and interrupt the network connectivity. Moreover, instead of full field coverage, the sensor network just needs to move and sample certain target locations on the field, but can efficiently reconstruct the map of the scalar field due to the high correlation of the sensory data.

The remainder of this paper is organized as follows. The background of CS and the collaborative control algorithm and the Problem Formulation are addressed in Section II. The distributed compressive and collaborative sensing (CCS) algorithm and some discussions are mentioned Section III. The communication power consumptions is analyzed and formulated heuristically in Section IV. Simulation results are shown in Section V and finally, conclusion and suggestions for the future work are presented in Section VI.

II. Background and Problem Formulation

A. Network Model

We assume a sensing area has N random positions corresponding to N unknown sensor readings for mapping. There are L distributed mobile robots working as a team in the area. The robots can connect to each other based on the communication range, denoted as R_c. The sensing range of the robots, denoted as R_s, can be selected smaller than R_c. In this paper we choose $R_s \leq \frac{1}{2}R_c$. Due to the limited power, we choose both R_s and R_c of the robots are much smaller than the sensing area. In order to lead the robots to measure data and avoid obstacles, a collaborative control algorithm [5], [25] is chosen with predefined virtual leaders to to sample the entire scalar field. The obstacles and the limitation of R_c cause the network not being always connected as shown in Figure 1 that may not affect the data collection method which will be discussed in the next sections.

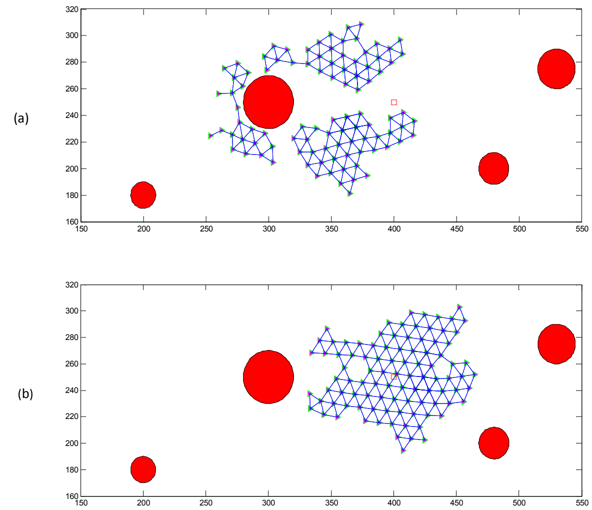

Fig. 1. Distributed mobile robots leading by the collaborative control algorithm in a sensing area can be (a) separated as three groups and (b) fully connected.

B. Distributed Collaborative Control Algorithm

As mentioned in our network model, we have a dynamic topology of an L-robot team based on an information graph $G(\boldsymbol{V}, \boldsymbol{E})$. The vertex set $\boldsymbol{V} = \{1, 2, ..., L\}$ present all the active mobile robots. The edge set $\boldsymbol{E} = \{(i, j) : i, j \in \boldsymbol{V}, i \neq j\}$ is defined by the communication links between robots based on a robot communication range R_c.

Let $\boldsymbol{p}_i, \boldsymbol{v}_i \in \Re^2$ denote the position and velocity vectors of the i^{th} robot. For simplicity, the kinematic motion of each robot is modeled as a particle:

$$\begin{cases} \dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i, \quad i = 1, 2, ..., L. \end{cases} \tag{1}$$

here $\boldsymbol{u}_i$ is the control input vector of robot ith. Equation (1) can be used to model mobile robots that have omni-directional motion capability such as the Rovio robots [26].

We define an actual neighborhood set of robot i at time t as follows:

$$N_i^a(t) = \left\{ j \in \boldsymbol{V} : \|\boldsymbol{p}_j - \boldsymbol{p}_i\| \leq R_c, j \neq i \right\}, \tag{2}$$

where $\|.\|$ is the Euclidean distance. The superscript a indicates the actual neighbors of robot i that is used to distinguish with virtual neighbors in the case of obstacle avoidance. The set of virtual neighbors of robot i at time t with K obstacles is defined as

$$N_i^v(t) = \{k \in \boldsymbol{V}_o : \|\boldsymbol{p}_{ik} - \boldsymbol{p}_i\| \leq R_0, \boldsymbol{V}_o = \{1, 2, ..., K\}\} \tag{3}$$

where R_0 is the obstacle detection range. $\boldsymbol{V}_o$ is a set of obstacles. $\boldsymbol{p}_{ik}$ is the position of robot i projecting on the k^{th} obstacle. For convenience from now on we just use N_i^a and N_i^v by dropping t. The virtual neighbors are used to generate the repulsive force to push the robots away from the obstacles.

Mobile robots in the team will work together with neighbors to form a certain geometry formation (e.g quasi lattice formation) while avoiding collision with each other and static/moving obstacles. The distributed collaborative control law $\boldsymbol{u}_i$ for each robot is introduced [25], [27], [28] consisting of three terms: formation control $\boldsymbol{f}_i^f$, obstacle avoidance $\boldsymbol{f}_i^o$, and navigation $\boldsymbol{f}_i^n$, namely,

$$\boldsymbol{u}_i = \boldsymbol{f}_i^f + \boldsymbol{f}_i^o + \boldsymbol{f}_i^n. \tag{4}$$

The term $\boldsymbol{f}_i^f$ is the pair-wise attractive/repulsive action function to allow robots to maintain a certain distance with its neighbors as well as avoid collision among them. This function is designed as:

$$\boldsymbol{f}_i^f = c_1^a \sum_{j \in N_i^a} \phi_a(\|\boldsymbol{p}_j - \boldsymbol{p}_i\|_\sigma) n_{ij} + c_2^a \sum_{j \in N_i^a} a_{ij}(p)(\boldsymbol{v}_j - \boldsymbol{v}_i),$$

where $\phi_a(.)$ is the action function defined in [25]. The σ - norm, $\|.\|_\sigma$, defined as $\|\boldsymbol{x}\|_\sigma = \frac{1}{\epsilon}[\sqrt{1+\epsilon\|\boldsymbol{x}\|^2} - 1]$ with $\epsilon > 0$, unlike the Euclidean norm $\|z\|$, is differentiable every where.

During the scalar field exploration, one may have obstacles distributed along the field, and the obstacle avoidance function needs to be designed to enable obstacle avoidance capability of mobile robots. With this purpose the term $\boldsymbol{f}_i^o$ in Equation (4) is the repulsive action function to allow robots to avoid obstacles, and it is designed as:

$$\boldsymbol{f}_i^o = c_1^o \sum_{j \in N_i^v} \phi_o(\|\boldsymbol{p}_{ik} - \boldsymbol{p}_j\|_\sigma) n_{ik} + c_2^o \sum_{j \in N_i^v} b_{ik}(p)(\boldsymbol{v}_{ik} - \boldsymbol{v}_j),$$

where $\phi_o(.)$ is a repulsive action function to allow robots to avoid obstacles defined in [25]. n_{ij} and n_{ik} are vectors along the line connecting $\boldsymbol{p}_j$ to $\boldsymbol{p}_j$ and $\boldsymbol{p}_{ik}$ to $\boldsymbol{p}_i$, respectively. Matrices $A = [a_{ij}]$ and $B = [b_{ik}]$ are the adjacency matrices defined by graph G [29] .

Since the scalar field is very large comparing to the limited sensing of each robot, all mobile robots need to move and collaborate with each other to sample certain locations and reconstruct the map of the field. To allow the sensor/robot network to travel to these certain locations, the navigation function is designed as

$$\boldsymbol{f}_i^n \quad = \quad -c_1^n(\boldsymbol{p}_i - \boldsymbol{p}_t) - c_2^n(\overline{\boldsymbol{p}}_{(N_i^a \cup \{i\})} - \boldsymbol{p}_t), \tag{5}$$

where c_1^n and c_2^n are positive constants, and $(\overline{\boldsymbol{p}}_{(N_i^a \cup \{i\})}$ is defined as $\overline{\boldsymbol{p}}_{(N_i^a \cup \{i\})} = \frac{1}{|N_i^a \cup \{i\}|} \sum_{i=1}^{|N_i^a \cup \{i\}|} \boldsymbol{p}_i$, where $|N_i^a \cup \{i\}|$ is the number of robots in robot i's local neighborhood including robot i itself. $\boldsymbol{p}_t$ is a target location that the robot network wants to move there to collect data. Certainly, the larger scalar field the more target locations are needed to allow the robot network to cover the field.

C. Compressive Sensing (CS)

1) Sparse signal presentation: Compressed sensing techniques provide us a sound approach to recover a sparse or compressible signal from undersampled random projections, also refer to as measurements [12], [13]. In order to apply CS, signals are supposed to be sparse or compressible. A vector signal $\underline{\mathbf{X}} \in R^N$ ($\underline{\mathbf{X}} = [x_1\, x_2 \ldots x_N]^T$) is k-sparse if it has k non-zero elements. In another case, $\underline{\mathbf{X}}$ may be dense (most elements of $\underline{\mathbf{X}}$ are non-zeros) but can be considered as sparse in Ψ domain if $\underline{\mathbf{X}} = \Psi\underline{\Theta}$ and $\underline{\Theta}$ is a k-sparse vector.

When employing CS to a sensor/robot network for monitoring purposes, vector $\underline{\mathbf{X}}$ represents all N unknown readings from the sensing area. The gain obtained from using CS is that the number of CS measurements required (M) are much less than the number of the sensory readings (N).

2) Signal sampling: Vector $\underline{\mathbf{Y}}$, called the measurement vector, contains data sampled from $\underline{\mathbf{X}}$ with length N; $\underline{\mathbf{Y}} \in R^M$ ($\underline{\mathbf{Y}} = [y_1\, y_2 \ldots y_M]^T$), and $M \ll N$. The random measurements are generated by $\underline{\mathbf{Y}} = \Phi\underline{\mathbf{X}}$ where $\Phi \in R^{M \times N}$ is often a full-Gaussian matrix or binary matrix [30], called the measurement matrix. $\underline{\mathbf{Y}} \in R^M$ is the measurement vector with $y_i = \sum_{j=1}^n \varphi_{i,j} x_j$, where $\varphi_{i,j}$ are all entries on the i^{th} row of the projection matrix Φ.

3) Signal recovery: The number of measurements required to reconstruct the original signals perfectly with high probability is $M = \mathcal{O}(k\, log\, N/k)$ following the l_1 optimization problem given by [14]. The unknown signals are reconstructed based on the CS measurements as follows

$$\hat{\underline{\Theta}} = arg\, min \parallel \underline{\Theta} \parallel_1,\ subject\, to \ \ \underline{\mathbf{Y}} = \Phi\Psi\underline{\Theta}, \tag{6}$$

where $||\underline{\Theta}||_1 = \sum_{i=1}^N |\theta_i|$ and $\underline{\mathbf{X}} = \Psi\underline{\Theta}$. The l_1 optimization problem can be solved with linear programming techniques such as Basis Pursuit (BP) [12]. In reality, noisy CS measurements should be considered while collecting as $\underline{\mathbf{Y}} = \Phi\underline{\mathbf{X}} + \underline{\mathbf{e}}$, with $||\underline{\mathbf{e}}||_2 < \epsilon$. The recovery algorithm is addressed as follows

$$\hat{\underline{\Theta}} = arg\, min \parallel \underline{\Theta} \parallel_1,\ subject\, to \ \ \parallel \underline{\mathbf{Y}} - \Phi\Psi\underline{\Theta} \parallel_2 < \epsilon. \tag{7}$$

D. Problem Formulation

As mentioned in the network model, given the unknown scalar field with N values needed to be observed and represented as $\underline{\mathbf{X}} \in R^N$ ($\underline{\mathbf{X}} = [x_1\, x_2 \ldots x_N]^T$). After a period of time T, the collaborative control leads the L connected mobile robots to sample almost the sensing area. We divide the period T into M time slots as $t = T/M$. The data collection processes are steps as follow

1) Collecting data: At time instant t, each mobile robot can measure several positions covered by its sensing range R_s. Due to the random distribution, on average, the number of positions sensed by a robot are

$$\gamma = \frac{N}{S}\pi R_s^2, \tag{8}$$

where S is the area of the sensing area. If there is no overlapped sensing area between L robots, at a time instant t, there are about γL positions sampled. Each mobile robot simply sums all sensory data as $\sum_{i=1}^{\gamma} x_i$ to be exchanged between connected robots. As shown in Figure 2, there are non-overlapped circular regions sensed by L mobile robots. There is no region sampled twice at the same time.

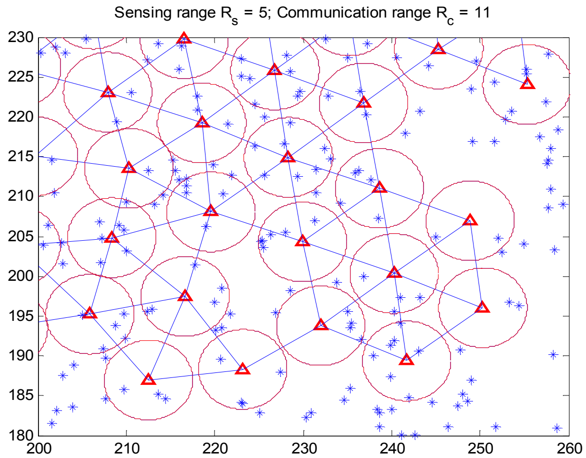

Fig. 2. An illustration of communication links (blue lines) and sensing range (within red circles) within a group of robots sampling in a small region of the sensing area

2) Exchanging the accumulated sensing information: Connected mobile robots exchange their accumulated data attached with the sensing position indices sampled by each robot through their neighbors based on the robot communication range R_c. After a certain numbers of sharing data, called convergence time, each mobile robot achieves one CS measurement which is a sum of all readings sampled by L mobile robots as

$$y_t = \sum_{j=1}^{L} \sum_{i=1}^{\gamma} x_i. \tag{9}$$

We have two definitions at this point:
Definition 1: Two robots are considered neighbors if the Euclidean distance between them is less than or equal to the communication range R_c.
Definition 2: Convergence time, denoted as I, is the average number of times each robot updates data from the neighbors until it achieves a CS measurement collected from all robots in the team-work. It measures of how fast a group of robots can create one CS measurement.

3) Moving for more CS measurements: Being led by the collaborative control, the group of L mobile robots move to different regions for sensing. At another time $(t + 1)$, one more CS measurement is created following step 1 and 2. After T or M periods of time ($T = Mt$), each robot achieves M CS measurements for the CS recovery process as $\underline{\mathbf{Y}} = [y_1\, y_2 \ldots y_M]$.

4) Data reconstruction at each distributed robot: Each robot implements the CS recovery algorithm to reconstruct all N readings to build a scalar map itself following Equation (6) or (7) with noiseless or noisy CS measurements, respectively.

E. Analysis of Measurement Matrix Φ

By adding the readings together, the measurement matrix created by collecting M CS measurements ($\underline{\mathbf{Y}} = \Phi\underline{\mathbf{X}}$) is a sparse binary matrix as shown in Equation (10). We need to ensure that this type of matrix can work as well as Gaussian matrices in which all entries are i.i.d. zero-mean random variables with variance $\frac{1}{M}$.

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & \ldots & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & \ldots & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 0 & 1 & 0 & 0 & 1 & \ldots & 0 \end{bmatrix}_{M\times N} \tag{10}$$

If L mobile robots are connected all the time of sampling, the matrix has the average row weight of γL. It means that the matrix has γL non-zeros elements ('1') on average in each row. The remainders of $(N-\gamma L)$ in each row are zeros.

Based on the way positions are indexed and the way mobile robots sampling data, the measurement matrix is asymptotic to random sparse binary one which is well studied in [30]. It has been shown that this type of matrix can satisfy the restricted isometry property (RIP) and therefore can be used as an energy-efficient measurement matrix. In the simulation section, we will be showing that our measurement matrix ($\Phi_{M\times N}$) can work as well as a Gaussian matrix for the CS recovery process. It means that we only need to sample the sensing area with a few robots at a time to form 1 CS measurement that does not affect CS performance.

III. The Compressive and Collaborative Mobile Sensing Algorithm

The proposed compressive and collaborative mobile sensing algorithm is summarized in Algorithm 1. At the *Initialization phase*, each robot stores its own data collected within its sensing range R_s which is added as only one scalar value. The *Exchange data phase* shows the collaboration between connected robots to build a CS measurement between them by exchanging data at a time instant t. The distributed robots only communicate to their neighbors by the time of this phase. Note that each robot only adds new data that it has not stored from robots identified by their indices. Based on these indices, a sparse binary measurement matrix is created at each robot for the CS recovery process.

In the *Moving phase*, the group of robots move following the algorithm as mention in Section II-B. Depending on the accuracy of the reconstructed map we are expecting, the robots move a certain number of times for those number of CS measurements. The number of measurements could be pre-defined for each robot. The *Data recovery phase* completes the task of building a map at each robot.

In some cases, mobile robots maybe disconnected from the others to avoid obstacles. It means that the CS measurements collected at that time are not contributed by all L robots and are different. This results in the measurement matrix that the corresponding rows may have different row weights. As analyzed in [30] this does not affect the CS recovery algorithm. By applying CS, each mobile robot only has to collect M measurements to build a scalar field map ($M \ll N$). This reduces significantly consumed power not only for communications but also for robot movements.

Algorithm 1: Distributed Compressive and Collaborative Mobile Sensing Algorithm

1. Initialization phase:
- $\underline{\mathbf{X}} = [x_1\, x_2 \ldots x_N]^T$ represents N unknown values.
- $\underline{\mathbf{Y}} = [y_1\, y_2 \ldots x_M]^T$ represents CS measurements created at each robot for the CS recovery process.
- $\underline{\mathbf{S}} = [s_1\, s_2 \ldots s_L]$ represents data stored at L robots. In this phase, each robot stores its own readings as $s_j = \sum_{i=1}^{\gamma} x_i$.
- Convergence time at each robot
  $I_S = [I_{S1}\, I_{S2} \ldots I_{SL}] = zeros(1, L)$;
- A communication range R_c is chosen for all L robots.
- A neighborhood matrix Q is created based on connections among L robots as

$$Q[i,j] = \begin{cases} 1, & if\ r(i,j) \leq R_c \\ 0, & others \end{cases} \quad (11)$$

2. Exchange data phase between L robots to form 1 CS measurement at each robot:
for *k = 1 to L* **do**
  - Finding neighbors of each robot
    $nei = find\,(\,Q(k,:) == 1)$;
  - A robot adds all received data together including its own readings
    $s(k) = s(k) + s(nei)$;
  - The convergence time at each robot is counted as
    $I_S(k) = 1$;
end
- The "new data" delivered to robots is recognized by the attached sensing positions.
while *No new data received* **do**
  for *i = 1 to L* **do**
    if *New data received from any neighbor (j)* **then**
      - Add new data to the previous stored data as
      $s(i) = s(i) + s(j)$;
      - Forward new s(j) to its nei;
      - Convergence time at each robot increases as
      $I_S(i) = I_S(i) + 1$;
    end
  end
  - Repeat updating new data at each robot
end
- The first CS measurement is created at robot i^{th} as
  $y_1 = s_i$;
- The convergence time is calculated as
  $I = mean(I_S)$;
3. Moving phase
- The mobile robots move following the collaborative algorithm in Equation (4).
- After M periods of time, each robot collects and stores M measurements required.
4. Data recovery phase
All unknown values ($\underline{\mathbf{X}}$) are reconstructed based on the CS measurements ($\underline{\mathbf{Y}}$) stored at each robot following Equation (6) or (7).

IV. Communication power consumption analysis

The total consumed power for robot communications contains main elements: the consumed power for communications within all robot neighborhoods denoted as P_{nei}, the convergence time, denoted as I, and the number of measurements required denoted as M. In order to obtain the number of CS measurements, each mobile robot has to go through M time slots to collect data. The more the number of measurements, the more accurate the reconstructed data. So, the total power consumption can be calculated as

$$P_{total} = P_{nei} \times I \times M. \quad (12)$$

P_{nei} represents the total consumed power to transmit data between L neighborhoods that can be calculated as

$$P_{nei} = \omega \times R_c^{\alpha} \times L, \quad (13)$$

where ω represents the number of neighbors of each robot that forms the number of communications in each neighborhood. R_c is the robot communication range. The path loss exponent could be between $2 \leq \alpha \leq 4$ in free space and multi-path fading channels, respectively [31]. For simplicity, we choose $\alpha = 2$ throughout this paper. Hence

$$P_{total} = \omega R_c^2 LIM. \quad (14)$$

The convergence time I generally depends on both the robot density and the communication connections. We prefer to have small number of robots deployed in the area that can provide the small value of I. Since the robots working under the collaborative control algorithm are organized as a lattice network, the number of connections between them are limited. This does not change I much as we change communication range R_c. The convergence time can be considered as a constant with a fixed number of robots. This is shown in Figure 5 in the simulation section.

V. Simulation results

In this section, we chose a fixed dimension square sensing area 300×300 square units. We also assume that there are about 5000 positions need to be observed in the area. These positions are generated randomly and uniformly in the area. We deploy about 50 or 100 mobile robots led by the collaborative control algorithm. The real sensor readings to build the scalar map are collected from Sensorscope: Sensor Networks for Environmental Monitoring [32]. Without loss of generality, it is assumed that the power for transmitting 1 unit of data is 1 unit of power.

The result of the 50 robots flocking/moving together based on the control law (4) is presented in Figure 3. Parameters for the control law (4) are: $c_1^a = 30$, $c_2^a = 2\sqrt{c_1^a}$, $c_1^o = 1500$, $c_2^0 = 2\sqrt{c_1^o}$, $c_1^n = 1.1$ and $c_2^n = 2\sqrt{c_1^n}$. Three virtual target locations (see Equ. 5) selected for $\boldsymbol{p}_t$ are: [70 50], [250 70] and [250 250] to navigate the robots to visit entire scalar field. As can be seen 50 robots can move together to cover the scalar field while collecting CS measurements. The robots can form a quasi-lattice network formation and avoid collision with each other, and they may get disconnected sometimes when avoiding obstacles.

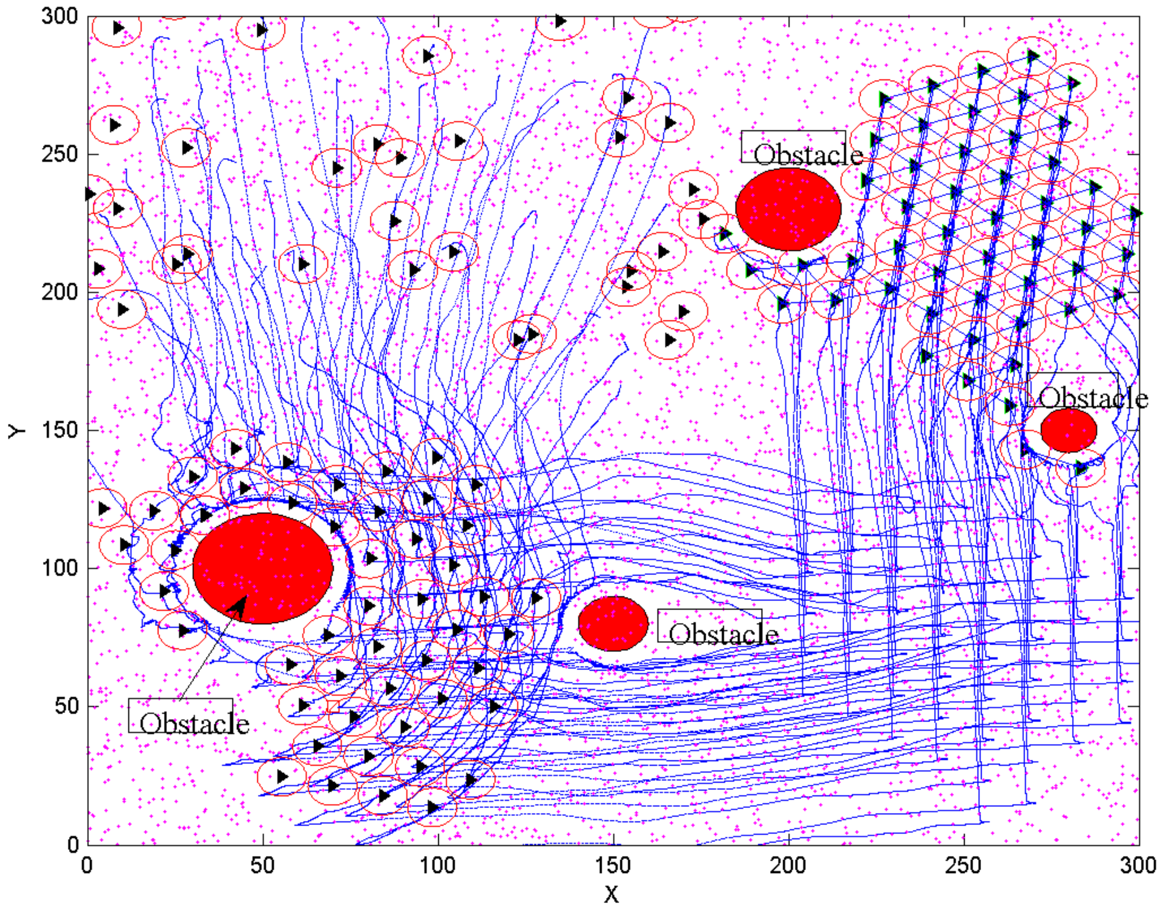

Fig. 3. The large scalar field has its size of 300×300 with 5000 random sample locations plotted in magenta color. Since the field is much larger than the sensing range of each robot/mobile sensor ($R_s = 5$), the group of 50 robots (black triangles) with sensing range that covers circular regions in the red circles have to move together along the field and take measurements. Three target locations for the navigation function defined in Equation (5) as $p^1 = [70\ 50]^T$, $p^2 = [250\ 70]^T$ and $p^3 = [250\ 250]^T$ are pre-determined to allow the mobile robot network to cover the field.

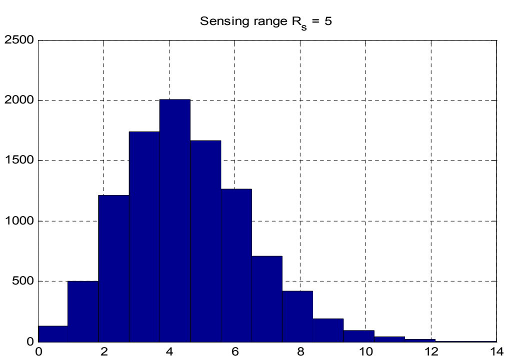

Fig. 4. Histogram of number of positions sampled by each robot.

As we chose the sensing range $R_s = 5$, Figure 4 shows the histogram of the number of positions sampled by each sensor in the area. These numbers will provide the number of positions sampled by L sensors at each sampling time. In the figure, the average value equals the one calculated from Equation 8 as $\gamma = \frac{5000}{300\times300}3.14 \times 5^2 = 4.36$.

Figure 5 depicts two convergence times (I) for both 50-robot and 100-robot networks versus different communication ranges. As mentioned, when we change the communication range R_c, these convergence times can be considered as constants. The smaller convergence time corresponding to the group of 50 robots should be chosen for energy saving. Figure 6 depicts the the total power consumption for all communications in the 50-robot network to collect and share $M = 800$ CS measurements versus different communication range R_c. The total power consumption is also calculated versus different number of measurements with a fixed $R_c = 10$, as shown in Figure 7.

As mentioned in [30], a sparse binary matrix should work as well as the full Gaussian measurement matrix which corresponds to the way of sampling all positions for each CS measurement. The normalized reconstruction errors ($\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_2}{\|\mathbf{X}\|_2}$) are calculated for comparison when both types of the matrices are used in the recovery process. In Figure 8 we show two sparse binary matrices created by two teams of robots, $L = 50$ and $L = 100$, compared to a full Gaussian matrix in a CS recovery process. The figure shows that either 50 or 100 can work as well as the case M CS measurements are collected from all positions ($N = 5000$) corresponding to the full Gaussian matrix.

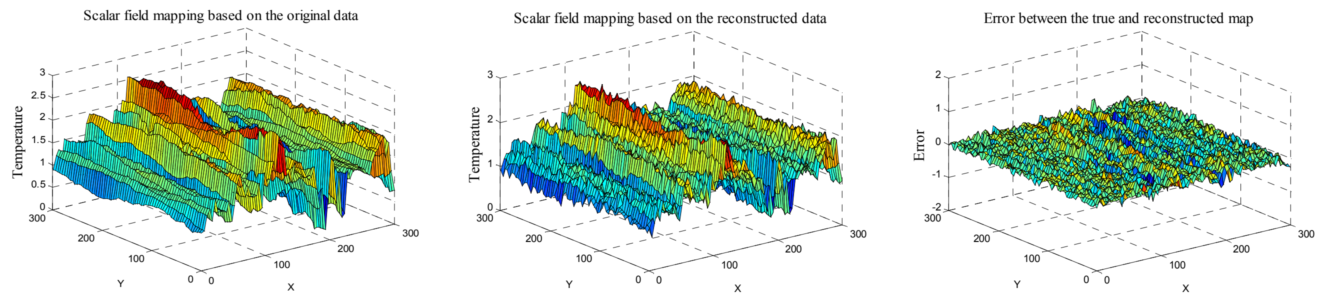

Fig. 9. Comparison of the true data and the reconstructed data and the reconstruction error when each robot stores $M = 800$ CS measurements.

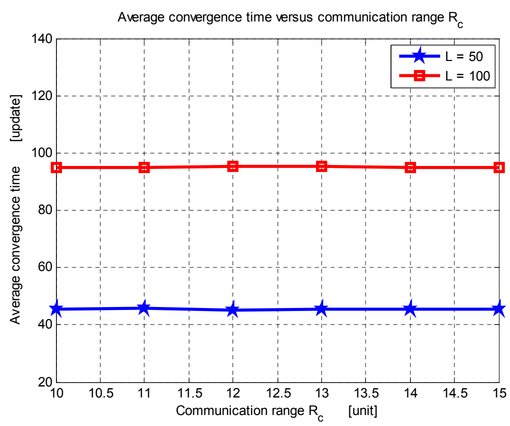

Fig. 5. The average convergence time versus different communication ranges in two team networks, 50-robot and 100-robot.

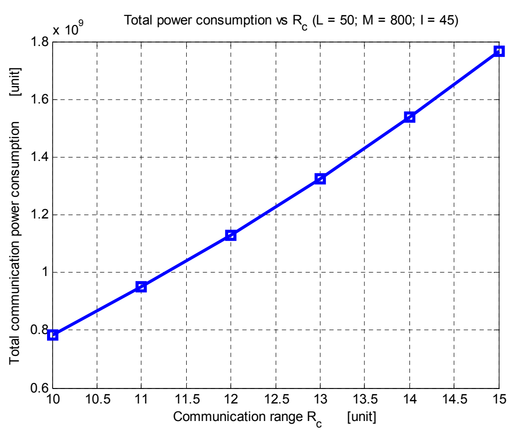

Fig. 6. Total power consumption for communication in the 50-robot network with different communication range R_c.

Figure 9 depicts the true and the reconstructed scalar maps in 3-D and the error map representing the difference between the true values and the reconstructed data as ($\underline{\mathbf{X}} - \hat{\underline{\mathbf{X}}}$). The error map looks quite uniform all over the sensing area.

* ***Discussion***: By combining CS and the collaboration, the group of robots can save a huge energy for transmitting data in the network as shown in Table I. Besides, the robots can also save energy from their motion and be flexible in connection between each other.

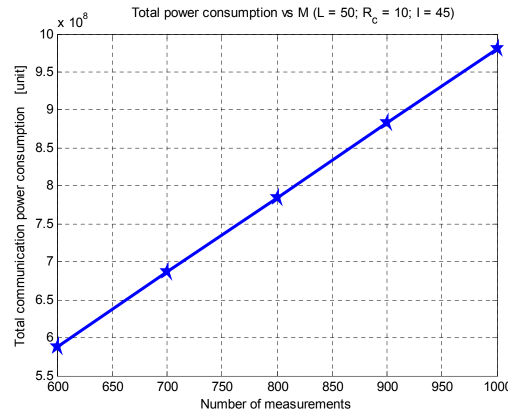

Fig. 7. Total power consumption for communication in the 50-robot network with different number of measurements M.

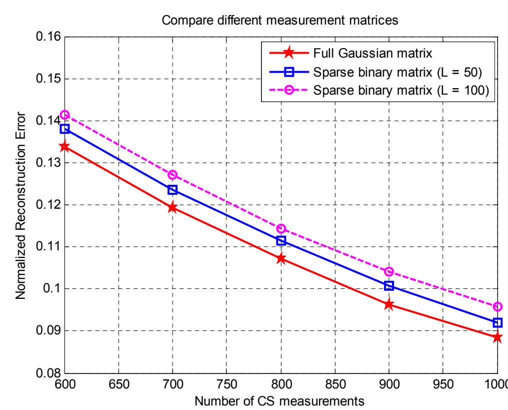

Fig. 8. Compare the reconstruction errors provided by three measurement matrices: full dense Gaussian and two sparse binary matrices created by 50 and 100 mobile robots, respectively.

TABLE I. COMPARISON BETWEEN USING CS AND NOT USING CS IN COMMUNICATION

	Number of robots	Number of positions to be observed	Samples sent from each robot	Total measurements sent
Apply CS	50	5000	1	$50 \times 800 = 40,000$
Not apply CS	50	5000	4.36	$50 \times 5000 = 250,000$

VI. Conclusion and Future work

In this paper we proposed a compressive and collaborative sensing algorithm for distributed mobile robots collecting data in MSNs to build scalar maps themselves. The novel combination is that the robots can save energy for both their motion and communication. Instead of sampling and sending a hug data among robots in order to build a map at each robot, a certain number of CS measurement are created to be able to reconstruct data needed. The communication power consumption and the coverage are analyzed and formulated heuristically. Simulation results are provided. We suggest that the smallest communication range and the smallest number of robots can minimize the total power consumption for the networks.

In future work, we find the boundary for either the number of robots and the communication range. This is directly related to the boundary of the sparsity of the measurement matrix that we can explore for further energy saving.

References

[1] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, pp. 301–311, Feb 2013.

[2] C. Tunca, S. Isik, M. Donmez, and C. Ersoy, "Distributed mobile sink routing for wireless sensor networks: A survey," *Communications Surveys Tutorials, IEEE*, vol. 16, pp. 877–897, Second 2014.

[3] H. M. La and W. Sheng, "Distributed sensor fusion for scalar field mapping using mobile sensor networks," *Cybernetics, IEEE Transactions on*, vol. 43, pp. 766–778, April 2013.

[4] H. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 45, pp. 1–12, Jan 2015.

[5] H. M. La, R. Lim, and W. Sheng, "Multirobot cooperative learning for predator avoidance," *Control Systems Technology, IEEE Transactions on*, vol. 23, pp. 52–63, Jan 2015.

[6] K. M. Lynch, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Trans. on Robotics*, vol. 24, pp. 710–724, Jun. 2008.

[7] J. Cortes, "Distributed Kriged Kalman filter for spatial estimation," *IEEE Trans. on Auto. Control*, vol. 54, no. 12, pp. 2816–2827, 2009.

[8] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, 2009.

[9] F. Zhang and N. Leonard, "Cooperative filters and control for cooperative exploration," *Automatic Control, IEEE Transactions on*, vol. 55, pp. 650–663, March 2010.

[10] R. Graham and J. Cortes, "Spatial statistics and distributed estimation by robotic sensor networks," *American Control Conf.*, pp. 2422–2427, 2010.

[11] S. Li, R. Kong, and Y. Guo, "Cooperative distributed source seeking by multiple robots: Algorithms and experiments," *Mechatronics, IEEE/ASME Transactions on*, vol. 19, pp. 1810–1820, Dec 2014.

[12] D.L.Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, pp. 1289 – 1306, 2006.

[13] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, pp. 489 – 509, Feb. 2006.

[14] R. Baraniuk, "Compressive sensing [lecture notes]," *Signal Processing Magazine, IEEE*, vol. 24, pp. 118 –121, July 2007.

[15] T. Do, L. Gan, N. Nguyen, and T. Tran, "Fast and efficient compressive sensing using structurally random matrices," *Signal Processing, IEEE Transactions on*, vol. 60, pp. 139–154, Jan 2012.

[16] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005*, pp. 1859–1864 vol. 3, June 2005.

[17] E. Kokiopoulou and P. Frossard, "Distributed classification of multiple observation sets by consensus," *Signal Processing, IEEE Transactions on*, vol. 59, pp. 104–114, Jan 2011.

[18] M. T. Nguyen, "Minimizing energy consumption in random walk routing for wireless sensor networks utilizing compressed sensing," in *System of Systems Engineering (SoSE), 2013 8th International Conference on*, pp. 297–301, June 2013.

[19] M. T. Nguyen and N. Rahnavard, "Cluster-based energy-efficient data collection in wireless sensor networks utilizing compressive sensing," in *Military Communications Conference, MILCOM 2013 - 2013 IEEE*, pp. 1708–1713, Nov 2013.

[20] M. T. Nguyen and K. Teague, "Tree-based energy-efficient data gathering in wireless sensor networks deploying compressive sensing," in *Wireless and Optical Communication Conference (WOCC), 2014 23rd*, pp. 1–6, May 2014.

[21] M. T. Nguyen and K. Teague, "Neighborhood based data collection in wireless sensor networks employing compressive sensing," in *Advanced Technologies for Communications (ATC), 2014 International Conference on*, pp. 198–203, Oct 2014.

[22] H. Wang, Y. Zhu, and Q. Zhang, "Compressive sensing based monitoring with vehicular networks," in *INFOCOM, 2013 Proceedings IEEE*, pp. 2823–2831, April 2013.

[23] Y. Mostofi, "Cooperative wireless-based obstacle/object mapping and see-through capabilities in robotic networks," *Mobile Computing, IEEE Transactions on*, vol. 12, pp. 817–829, May 2013.

[24] M. T. Nguyen and K. A. Teague, "Random sampling in collaborative and distributed mobile sensor networks utilizing compressive sensing for scalar field mapping," in *System of Systems Engineering Conference (SoSE), 2015 10th*, pp. 1–6, May 2015.

[25] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[26] http://www.wowwee.com/en/support/rovio.

[27] H. M. La and W. Sheng, "Flocking control of a mobile sensor network to track and observe a moving target," *Proc. of the 2009 IEEE Inter. Conf. on Robotics and Automation*, pp. 3129–3134, 2009.

[28] H. M. La and W. Sheng, "Dynamic targets tracking and observing in a mobile sensor network," *Robotics and Autonomous Sys.*, pp. 996–1009, 2012.

[29] F. Bullo, J. Cortes, and S. Martinez, "Distributed control of robotic networks," *Ser. Applied Mathematics Series. Princeton University Press*, 2009.

[30] R. Berinde, A. C. Gilbert, P. Indyk, H. J. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," *CoRR*, vol. abs/0804.4666, 2008.

[31] T. S. Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall, 2 ed., Jan. 2002.

[32] http://lcav.epfl.ch/op/edit/sensorscope en.