Observability and Estimation in Distributed Sensor Networks

Meng Ji and Magnus Egerstedt

***Abstract*— A graph-based estimation method is proposed for distributed sensor networks. In particular, we study the problem of estimating a spatially distributed parameter. The proposed method is based on the consensus algorithm as well as basic linear control theory. As a consequence, observability of sensor networks is also investigated from the graph-theoretic point of view.**

***Index Terms*— Distributed sensor networks, networked systems, observability, equitable partitions, agreement dynamics, algebraic graph theory.**

I. Introduction

In recent years, distributed sensor networks (DSNs) have found a number of applications, ranging from field surveillance [1], environment monitoring [2], to geo-scientific exploration [3], [4]. A DSN typically consists of a collection of sensor nodes, each integrated with a microprocessor, a transceiver, and, sometimes, actuators. Thus, beyond sensing, the nodes are capable of simple information processing and short distance communication. The sensors are usually deployed over a large area, where the sensors collect certain information and report to one or more base stations for data synthesis and decision making.

Sensing with DSNs poses an interesting problem to the control and communication communities, because it requires the combination of well-coordinated sensing, communication, and control strategies to satisfy constraints on power and bandwidth usage [5]. The restricted range of transmission of the sensor nodes, and the fact that they are spread over a large area, make the distributed sensing more challenging.

Two types of coordination strategies have been proposed in the literature for information collection in DSNs, namely multi-hop [6] and mobile central nodes [7], [8]. In the multi-hop method, the sensor nodes transfer their data to the base station either directly or by ways of other nodes. In such a setting, an intermediate node acts as a router in that it sends its own data together with the data from other nodes (in separate packets) to the next node in the data link. The cost of communication (mainly power related) and the error accumulation are two major hurdles for the multi-hop method to overcome.

In the mobile central node(s) method, the central node(s) can move around and collect data from the nodes that are within transmission range. Having avoided the difficulties faced by the multi-hop method, the mobile sensor nodes method has its own limitation. Each mobile node can collect information from only a small portion of the network, and the size of the deployment area and its moving speed determine the update frequency. If the area is large and the speed is low, the frequency can be low enough to significantly impact the concurrency of the data.

Other related approaches can be found in [9] in the context of consensus filters, where the author reduced the distributed Kalman filter into two separate dynamical consensus problem. In [10], the authors provided a stochastic strategy to generate optimal sensor trajectories for the coverage problem of mobile sensor network. Estimation problems in a network with dropped packet was studied in [11], where an optimal strategy was presented. This strategy required every node in the network to store and process the packet they received.

The problem we are interested in this paper is the distributed sensing problem, where a network of sensors are deployed in the area-of-interest to monitor *spatially distributed* information, e.g., temperature, radioactivity, humidity, etc.. This mission is different from the sensor network consensus, where the network is monitoring a homogeneous environment, and the major mission of the network is to reach an agreement over the parameters being monitored. Here we are interested in the spatial distribution of the parameters instead of the average value, which makes the mission more challenging.

In this paper, we propose an estimation method based on the consensus algorithm and linear control theory. This method is promising because of its low communication cost and fast convergence rate. The remainder of this paper is organized as follows: In Section II we introduce some basic notation and preliminary results related to consensus seeking. In Section III, we propose our main estimation algorithm. In order to investigate the observability of the networks, we introduce equitable partitions of graphs in Section IV, and derive a necessary condition for the network to be observable in Section V.

II. Graph-Based Modeling and Consensus Seeking

Graphs are adopted as encodings of the limited information present in networked systems, where edges between nodes correspond to the ability of sharing information. In this section, we introduce some basic notation and concepts in graph-based modeling, and discuss some useful properties related to the matrix representations of graphs.

The authors are with the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA. Email: {mengji,magnus}@ece.gatech.edu.

Their work is supported by the U.S. Army Research Office through the grant # 99838.

A. Basic Notation and Concepts in Graph Theory

An (undirected) *graph* $\mathcal{G}$ is defined by a set $V = \{1, \dots N\}$ of *nodes* and a set $E \subset V \times V$ of *edges*. Two nodes i and j are *adjacent*, or *neighbors*, if $(i,j) \in E(\mathcal{G})$, and the neighboring relation is indicated with $i \sim j$, while $\mathcal{N}_{\mathcal{G}}(i) = \{j \in V(\mathcal{G}) : j \sim i\}$ collects all neighbors to the node i.

A *path* $i_0 i_1 \dots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1, \dots, L$, and a graph $\mathcal{G}$ is *connected* if there exists a path between any pair of distinct nodes.

Let $\mathcal{G} = (V, E)$ and $\mathcal{G}' = (V', E')$ be two graphs. We call $\mathcal{G}'$ a *subgraph* of $\mathcal{G}$ (and $\mathcal{G}$ a *supergraph* of $\mathcal{G}'$) if $V' \subseteq V$ and $E' \subseteq E$, and we denote this by $\mathcal{G}' \subseteq \mathcal{G}$. A subgraph $\mathcal{G}'$ is said to be *induced* from the host graph $\mathcal{G}$ if $E' = E \cap (V' \times V')$. In other words, an induced graph is obtained by deleting a subset of nodes and all the edges connecting to those nodes.

The *adjacency* matrix of a graph $\mathcal{A} \in \mathbb{R}^{N \times N}$, with N denoting the number of nodes, is defined by

$$[\,\mathcal{A}(\mathcal{G})\,]_{ij} = \begin{cases} 1 & \text{if} \quad (i,j) \in E(\mathcal{G}) \\ 0 & \text{otherwise.} \end{cases}$$

Let d_i be the degree of node i and let $\mathcal{D}(\mathcal{G}) = \mathbf{Diag}([d_i]_{i=1}^N)$ be the corresponding diagonal degree matrix, then the graph *Laplacian* matrix can be defined by

$$\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G}).$$

It is known that the Laplacian matrix is positive semidefinite, and that the multiplicity of the zero eigenvalue of the graph Laplacian is equal to the number of connected components of the graph [12]. Moreover, if the graph is connected, then the eigenvector associated with the zero eigenvalue is $\mathbf{1}$.

B. Consensus Seeking

The consensus algorithm plays an important role in our estimation method. The consensus problem is normally formulated in the context of networked systems whose constitutional elements are called agents. The goal is to reach an agreement over some state-related variables [13].

Given a networked system with N agents, we can model the network by a graph $\mathcal{G}$ where nodes represent agents and edges are inter-agent communication links.[1] Let $x_i(t) \in \mathbb{R}^d$ denote the state of agent i at time t, whose dynamics is described by the single integrator

$$\dot{x}_i(t) = u_i(t), \; i = 1, \dots, N, \tag{1}$$

with $u_i(t)$ being node i's control input. The state can be the position of robot i in a multi-robot system, or the state of an distributed estimation process associated with sensor node i in a DSN. [2]

Next, we allow agent i to have access to the relative state information with respect to its neighbors and use it to compute its control $u_i(t)$. For example, one can adopt the nearest-neighbor rule:

$$u_i(t) = -\sum_{j \sim i} (x_i(t) - x_j(t)). \tag{2}$$

If $d = 1$, i.e., the state is a scalar, (2) can be represented as the Laplacian dynamics of the form

$$\dot{x}(t) = -\mathcal{L}(\mathcal{G})x(t), \tag{3}$$

where $x(t) = [x(t)_1, \; x(t)_2, \dots, \; x(t)_N]^T$ denotes the aggregated state vector of the multi-agent system[3].

The localized rule in (2) has been shown to solve the rendezvous problem, which has attracted considerable attention [14]–[16]. Some other networked systems problems, e.g., formation control [17]–[19], consensus [20] or agreement [21], [22], flocking [23], [24], etc., also share the same distributive flavor with the rendezvous problem.

III. Graph-Based Estimation of Distributed Sensor Networks

Given a DSN with N sensors, let $\mathcal{G}_s = [V_s, E_s]$ denote the underlying *sensor graph*, where V_s is associated with the set of sensor nodes, and E_s represent the communication links among the nodes. Let $q_i \in \mathbb{R}$ be the sensor value of node i, $\; i = 1, \dots, N$. Next, let $x \in \mathbb{R}^N$ be the state of a distributed estimator, and x_i be the component associated with node i. Again, note that x_i is different form the real sensed value q_i in that x_i decided by a distributed estimation algorithm. It is x_i, not q_i, that is transmitted through the communication links and processed by other nodes, which is essential to our method. In other words, each node, say i, has three roles: First, it is a sensor that monitors the parameter value, q_i. Secondly, it is a processing unit of a distributed computing system, which updates its state, x_i, by processing the information gathered from its communication links. Thirdly, it is a transceiver that exchanges its state information with its neighbors $x_j, \; j \in \mathcal{N}_{\mathcal{G}_s}(i)$.

Our proposed estimation algorithm can then be described in the following way: At time zero, node i records its sensing value q_i and use it as the initial value for x_i. Then it updates its own state based on the local information, i.e., all the states of its neighbors and the state of itself. The central node(s) collects information from a subset of the nodes and, hopefully, can recover the original information at each node, i.e., q_i, from the observation. The algorithm can be formulated as

$$\frac{dx_i}{dt} = \sum_{j \in \mathcal{N}(i)} f(x_i(t) - x_j(t)), \; x_i(0) = q_i, \; i = 1, \dots, N, \tag{4}$$

where $\mathcal{N}_{\mathcal{G}_s}(i)$ is the neighborhood of node i. Based on this updating scheme, the central nodes retrieve observation

$$y(t) = g(x(t)), \tag{5}$$

[1] Throughout this paper, we assume that the sensor nodes do not move and that the communication links are stable, which means that the graph associated with the network is static.

[2] Note that in the case of DSN, we need to distinguish between the state of the estimation process and the actual value observed by a sensor node. The state of the estimation process is updated according to the dynamics in (1), where u_i is the information input to node i, while u_i may not be directly affected by the real sensor value at point i.

[3] Most results from this paper can be directly extended to higher dimensions using the Kronecker product

where $x(t) = [x_1(t), \ldots, x_N(t)]^T$, and $y(t) \in \mathbb{R}^m,\ m << N$, with m being the number of central nodes. Our goal here is to *uniquely* determine $q = [q_1, \ldots, q_N]^T$ from $y(t),\ t > 0$.

If we adopt the nearest neighbor rule (4) for $f(\cdot)$, we get

$$\frac{dx_i}{dt} = \sum_{j \in N(i)} (x_j - x_i),\ i = 1, 2, \ldots, N. \tag{6}$$

Using the graph-based control notation, and let $g(\cdots)$ be a linear transformation given by matrix C, (5) and (6) can be written as

$$\begin{cases} \dot{x} &= -\mathcal{L}(\mathcal{G}_s)x, \quad x(0) = q, \\ y &= Cx, \end{cases} \tag{7}$$

where $\mathcal{L}(\mathcal{G}_s)$ is the Laplacian matrix of the graph associated with the sensor network, and $C \in \mathcal{R}^{m \times N}$ is the observation matrix. Since the graph is assumed to be static, we will use $\mathcal{L}_s$ for $\mathcal{L}(\mathcal{G}_s)$ wherever it causes no confusion. From the properties of Laplacian matrix introduced in Section II, we can conclude that system (7) is stable, and $x_i(t)$ converges to $\sum_1^N q_j/N$, for all $i \in \{1, N\}$ if $\mathcal{G}$ is connected. Nonetheless, we know that if the system $(-\mathcal{L}_s, C)$ is observable, we can recover the initial state $x(0)$ from the observation $y(t)$. Note that, in the case of multiple central nodes, each central node collects raw information from a subset of sensors and sums them up together, an operation that equals that of multiplying x by each row of C. Then the summation results are reported to a single super node for the final information processing and decision making.

We now restrict C to be a $(0, 1)$ matrix such that an "1" in column j, row i means that there is a communication link between sensor j and central node i.

One can design different observers based on (7), and a simple way of recovering the information, assuming the parameter is static or changing very slowly, is to solve the system

$$y(t) = Ce^{-t\mathcal{L}_s}x(0),\ t \geq 0,$$

or in the discrete time form

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(T-1) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ -C\mathcal{L}_s \\ \vdots \\ C(-\mathcal{L}_s)^{T-1} \end{bmatrix}}_{\mathcal{O}_T} x(0),$$

where $\mathcal{O}_T$ becomes the observability matrix when $T = N$. Of course, more sophisticated estimation methods are available in the literature, but they are not needed for the developments of this paper.

As we can see, the proposed estimation scheme is relying on one crucial assumption: the system $(-\mathcal{L}_s, C)$ is observable. Now, let us revisit the standard observability theorem from control theory.

Lemma 3.1: For a LTI system

$$\begin{cases} \dot{x} &= Ax, \quad x(0) = x_0, \quad x \in \mathbb{R}^n \\ y &= Cx, \end{cases} \tag{8}$$

the following are equal:

- system (8) is observable,
- the rank of the observability matrix $(\mathcal{O}(n))$ is n,
- rank of $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ equals n for every eigenvalue λ of A.

We can always check the rank condition to determine the observability of the network. However, it becomes infeasible when the number of nodes becomes very large. Instead we want to be able to guarantee observability when we build the network, and, for this sake, we want to understand how the topology of the network affects the observability. In the following section, we will show how the observability property is related to the topology of the sensor network.

IV. Equitable Partitions of Graphs

Equitable partitions and interlacing theory play important roles when deriving necessary conditions for observability. In this section, we introduce some definitions and lemmas needed to support this development.

Definition 4.1: An r-partition π of $V(\mathcal{G})$, with cells $C_1, \ldots, C_r$, is said to be *equitable* if each node in C_j has the same number of neighbors in C_i, for all i, j. We denote the cardinality of the partition π with $r = |\pi|$.

Let b_{ij} be the number of neighbors in C_j of nodes in C_i. The directed graph with the r cells of π as its nodes and b_{ij} edges from the ith to the jth cells of π is called the *quotient* of $\mathcal{G}$ over π, and is denoted by $\mathcal{G}/\pi$. An obvious, trivial equitable partition is the n-partition, $\pi = \{\{1\}, \{2\}, \ldots, \{n\}\}$. If a partition contains at least one cell with more than one node, we call it a *nontrivial equitable partition* (NEP), and the *adjacency matrix of the quotient* is given by

$$\mathcal{A}(\mathcal{G}/\pi)_{ij} = b_{ij}.$$

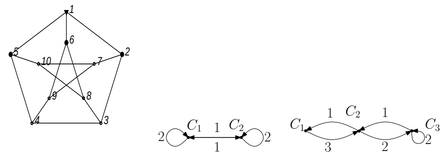
The equitable partition can be derived from graph automorphisms. For example, in the so-called Peterson graph, shown in Figure 1(a), one equitable partition π_1 (Figure 1(b)) is given by the two orbits of the automorphism groups, namely the 5 inner vertices and the 5 outer vertices. The adjacency matrix of the quotient is given by

$$\mathcal{A}(\mathcal{G}/\pi_1) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

The equitable partition can also be introduced by the equal distance partition. Let $C_1 \subset V(\mathcal{G})$ be a given cell, and let $C_i \subset V(\mathcal{G})$ be the set of vertices at distance $i-1$ from C_1. C_1 is said to be $completely\ regular$ if its distance partition is equitable. For instance, every vertex in the Peterson graph is completely regular and introduces the partition π_2 as shown in Figure 1(c). The adjacency matrix of this quotient is given by

$$\mathcal{A}(\mathcal{G}/\pi_2) = \begin{bmatrix} 0 & 3 & 0 \\ 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix}.$$

Partitions over a graph can be represented by vectors and



(a) (b) (c)

Fig. 1. Example of equitable partitions on (a) the Peterson graph $\mathcal{G} = J(5,2,0)$ and the quotients: (b) the NEP introduced by the automorphism is $\pi_1 = \{C_1^1, C_2^1\}, C_1^1 = \{1,2,3,4,5\}, C_2^1 = \{6,7,8,9,10\}$, and (c) the NEP introduced by equal-distance partition is $\pi_2 = \{C_1^2, C_2^2, C_3^2\}, C_1^2 = \{1\}, C_2^2 = \{2,5,6\}, C_3^2 = \{3,4,7,8,9,10\}$.

matrices.

Definition 4.2: A *characteristic vector* $p_i \in \mathbb{R}^n$ of a nontrivial cell C_i has 1's in the positions associated with C_i and 0's elsewhere. A *characteristic matrix* $P \in \mathbb{R}^{n\times r}$of a partition π of $V(\mathcal{G})$ is a matrix with the characteristic vectors of the cells as its columns.

Lemma 4.3: ([12] Lemma 9.3.1) Let P be the characteristic matrix of an equitable partition π of the graph $\mathcal{G}$, and let $\hat{\mathcal{A}} = \mathcal{A}(\mathcal{G}/\pi)$. Then $\mathcal{A}P = P\hat{\mathcal{A}}$ and $\hat{\mathcal{A}} = P^{+}\mathcal{A}P$, where $P^{+} = (P^TP)^{-1}P^T$ is the pseudo-inverse of P.

Lemma 4.4: ([12] Lemma 9.3.2) Let $\mathcal{G}$ be a graph with adjacency matrix $\mathcal{A}$, and let π be a partition of $V(\mathcal{G})$ with characteristic matrix P, then π is equitable if and only if the column space of P is $\mathcal{A}$-invariant.

Lemma 4.5: Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and let S be a subspace of $\mathbb{R}^n$. Then $S^{\perp}$ is A-invariant if and only if S is A-invariant.
The proof of this well-known fact can for example be found in [25].

Remark 4.6: Let $\mathcal{R}(\cdot)$ denote the range space. Suppose $|V(\mathcal{G})| = n$, $|C_i| = n_i$ and $|\pi| = r$, then we can find an orthogonal decomposition for $\mathbb{R}^n$ as

$$\mathbb{R}^n = \mathcal{R}(P) \oplus \mathcal{R}(Q), \tag{9}$$

where the matrix Q satisfies $\mathcal{R}(Q) = \mathcal{R}(P)^{\perp}$, such that its columns together with those of P form a basis for $\mathbb{R}^n$. Following Lemma 4.5, $\mathcal{R}(Q)$ is also $\mathcal{A}-$invariant.

Unlike matrix P, Q is derived from the nullspace of P and can be constructed in different ways. One possible choice of such a Q is the $n \times (n-r)$ matrix with r column blocks $Q = [Q_1, Q_2, \ldots, Q_r]$, where $Q_i \in \mathbb{R}^{n\times(n_i-1)}$ corresponds to C_i. Moreover, each column sums to zero in the positions associated with C_i and has zeros in the other positions. In other words,

$$Q_i = \begin{bmatrix} \mathbf{0} \\ \tilde{Q}_i \\ \mathbf{0} \end{bmatrix}_{n\times(n_i-1)}.$$

In Q_i, the upper and lower parts are zero matrices with appropriate dimensions (possibly empty).

One possible choice of Q matrix is using the orthonormal basis of $\mathcal{R}(P)^{\perp}$. We denote this matrix as $\bar{Q}$. If we define

$$\bar{P} = P(P^TP)^{-\frac{1}{2}}. \tag{10}$$

Note that the invertibility of P^TP follows from the fact that the cells of the partition are nonempty [4]. Moreover, it satisfies that $\bar{P}^T\bar{Q} = \mathbf{0}$ and $\bar{Q}^T\bar{Q} = I_{n-r}$. In other words,

$$T = [\bar{P} \mid \bar{Q}] \tag{11}$$

is a matrix whose columns are defined on an orthonormal basis of $\mathbb{R}^n$ based on the equitable partition π, and $\bar{P}$ and $\bar{Q}$ have the same column spaces as P and Q respectively.

V. Observability Analysis of DSN

Based on the equitable partition of a graph, we can further diagonallize the Laplacian matrix associated with the graph. Moreover, a similarity transformation can also be derived for the observability decomposition.

A. NEP-related Diagonalization of $\mathcal{L}(\mathcal{G})$

Lemma 5.1: If a graph $\mathcal{G}$ has a nontrivial equitable partition (NEP) π with characteristic matrix P, then the corresponding adjacency matrix $\mathcal{A}(\mathcal{G})$ is similar to a diagonal matrix

$$\bar{\mathcal{A}} = \begin{bmatrix} \mathcal{A}_P & \mathbf{0} \\ \mathbf{0} & \mathcal{A}_Q \end{bmatrix},$$

where $\mathcal{A}_P$ is similar to the adjacency matrix $\hat{\mathcal{A}} = \mathcal{A}(\mathcal{G}/\pi)$ of the quotient.

Proof: Let the matrix $T = [\bar{P} \mid \bar{Q}]$ be the orthonormal matrix with respect to π, as defined in (11). Now, let

$$\bar{\mathcal{A}} = T^T\mathcal{A}T = \begin{bmatrix} \bar{P}^T\mathcal{A}\bar{P} & \bar{P}^T\mathcal{A}\bar{Q} \\ \bar{Q}^T\mathcal{A}\bar{P} & \bar{Q}^T\mathcal{A}\bar{Q} \end{bmatrix}. \tag{12}$$

Since $\bar{P}$ and $\bar{Q}$ have the same column spaces as P and Q, respectively, they inherit the $\mathcal{A}-$invariance property, i.e.,

$$\mathcal{A}\bar{P} = \bar{P}B \quad \text{and} \quad \mathcal{A}\bar{Q} = \bar{Q}E.$$

for some matrices B and E. Moreover, since their column spaces are orthogonal complements of each other, one has

$$\bar{P}^T\mathcal{A}\bar{Q} = \bar{P}^T\bar{Q}E = \mathbf{0}$$

and

$$\bar{Q}^T\mathcal{A}\bar{P} = \bar{Q}^T\bar{P}B = \mathbf{0}.$$

In addition, by letting $D_P^2 = P^TP$, we obtain

$$\begin{aligned} \bar{P}^T\mathcal{A}\bar{P} &= D_P^{-1}P^T\mathcal{A}PD_P^{-1} \\ &= D_P(D_P^{-2}P^T\mathcal{A}P)D_P^{-1} \\ &= D_P\hat{\mathcal{A}}D_P^{-1}, \end{aligned} \tag{13}$$

and therefore the first diagonal block is similar to $\hat{\mathcal{A}}$. ■

Lemma 5.2: Let P be the characteristic matrix of a NEP in $\mathcal{G}$. Then $\mathcal{R}(P)$ is K-invariant, where K is any diagonal block matrix of the form

$$\begin{aligned} K &= \mathbf{Diag}([\underbrace{k_1,\ldots,k_1}_{n_1},\underbrace{k_2,\ldots,k_2}_{n_2},\ldots,\underbrace{k_r,\ldots,k_r}_{n_r}]^T) \\ &= \mathbf{Diag}([k_i\mathbf{1}_{n_i}]_{i=1}^r), \end{aligned}$$

[4]In fact, P^TP is a diagonal matrix with $(P^TP)_{ii} = |C_i|$.

$k_i \in \mathbb{R}$, $n_i = |C_i|$ is the cardinality of the cell, and $r = |\pi|$ is the cardinality of the partition. Consequently,

$$\bar{Q}^T K \bar{P} = \mathbf{0},$$

where $\bar{P} = P(P^T P)^{-\frac{1}{2}}$ and $\bar{Q}$ is chosen in such a way that $T = [\bar{P} \mid \bar{Q}]$ is a orthonormal matrix.

Proof: We note that

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_r \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & \dots & p_r \end{bmatrix},$$

where $P_i \in \mathbb{R}^{n_i \times r}$ is a row block which has 1's in column i and 0's elsewhere. On the other hand p_i is a characteristic vector representing C_i, which has 1's in the positions associated with C_i and zeros otherwise. With a little bit of calculation we can find

$$KP = \begin{bmatrix} k_1 P_1 \\ k_2 P_2 \\ \vdots \\ k_r P_r \end{bmatrix} = \begin{bmatrix} k_1 p_1 & k_2 p_2 & \dots & k_r p_r \end{bmatrix} = P\hat{K},$$

where $\hat{K} = \mathbf{Diag}([k_1, k_2, \dots, k_r]^T) = \mathbf{Diag}([k_i]_{i=1}^r)$, which shows that $\mathcal{R}(P)$ is K-invariant. Since $\mathcal{R}(\bar{Q}) = \mathcal{R}(P)^\perp$, by Lemma 4.5 $\bar{Q}$ is K-invariant as well and

$$\bar{Q}^T K \bar{P} = \bar{Q}^T \bar{P} \hat{K} = \mathbf{0}.$$

■

By the definition of the equitable partition, the subgraph induced by a cell is regular and every node in the same cell has the same number of neighbors outside the cell. Therefore, the nodes belonging to the same cell have the same degree, and thus by Lemma 5.2, $\mathcal{R}(\bar{Q})$ and $\mathcal{R}(P)$ are $\mathcal{D}$-invariant, where $\mathcal{D}$ is the degree matrix given by

$$\mathcal{D} = \mathbf{Diag}([d_i \mathbf{1}_{n_i}]_{i=1}^r),$$

with $d_i \in \mathbb{R}$ denoting the degree of each nodes in cell. Since the graph Laplacian satisfies $\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$, Lemmas 5.1 and 5.2 imply that $\mathcal{R}(\bar{Q})$ and $\mathcal{R}(P)$ are $\mathcal{L}$-invariant. Thereby, we have following corollary.

Corollary 5.3: Given the same condition as in Lemma 5.1, $\mathcal{L}$ is similar to a diagonal block matrix

$$\bar{\mathcal{L}} = T^T \mathcal{L} T = \begin{bmatrix} \mathcal{L}_P & \mathbf{0} \\ \mathbf{0} & \mathcal{L}_Q \end{bmatrix}, \tag{14}$$

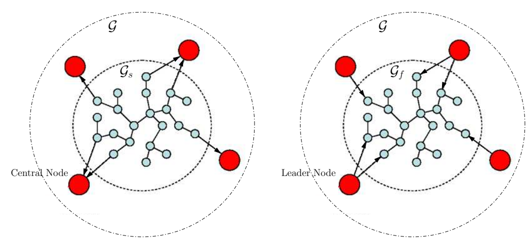
where $\mathcal{L}_P = \bar{P}^T \mathcal{L} \bar{P}$ and $\mathcal{L}_Q = \bar{Q}^T \mathcal{L} \bar{Q}$, and $T = [\bar{P} \mid \bar{Q}]$ defines a orthonormal basis for $\mathbb{R}^n$ with respect to π.

As (14) defines a similarity transformation, it follows that $\mathcal{L}_P$ and $\mathcal{L}_Q$ carry all the spectrum information of $\mathcal{L}$, i.e., they share the same eigenvalues as $\mathcal{L}$.

B. Necessary Condition for Observability

Given a DSN with N sensors and m central nodes, let $\mathcal{G}_s$ be the underlying sensor graph. We can expand the sensor graph by including the central nodes and the links connected to them, and denote the resulting graph as the *augmented sensor network* graph $\mathcal{G}$. Notice that $\mathcal{G}$ is nothing more than the graph obtained by adding the central nodes onto the sensor graph $\mathcal{G}_s$ and adding edges onto $\mathcal{G}_s$, which represent the information flows from sensor nodes to the central nodes.

Remark 5.4: The sensor graph $\mathcal{G}_s$ plays the similar role of the follower graph, $\mathcal{G}_f$, in the leader-follower structure discussed in our previous work [26][5]. The difference is that in the DSN the information flows from the sensor nodes to the central nodes, as shown in Figure 2(a), while in the leader-following structure, the information flows from the leaders to the followers, as shown in Figure 2(b). This similarity reminds us the duality between controllability and observability, i.e., if $(-\mathcal{L}_s, C)$ is observable then $(-\mathcal{L}_s^T, C^T)$ is controllable. One can also find this duality between the following theorem and Theorem 4.5 of [26], where we address the controllability problem of the leader-follower structure.



(a) (b)

Fig. 2. Shown in the picture are (a) the graph associate with a DSN ($\mathcal{G}_s$), and the augmented graph $\mathcal{G}$, and (b) the interaction graph $\mathcal{G}$ of the leader-follower structure with the same topology as the DSN in (a). Note the follower graph $\mathcal{G}_f$ is similar to $\mathcal{G}_s$, but the information flows are from the leader to the follower.

Theorem 5.5: Given a DSN with connected graph $\mathcal{G}_s$ and the augmented sensor graph $\mathcal{G}$, the system (7) is not complete observable if there exist NEPs on $\mathcal{G}$ and $\mathcal{G}_s$, say π and π_s, such that all the nontrivial cells of π are contained in π_s, i.e. $\exists\ \pi$ and π_s, such that $|C_i| = 1,\ \forall\ C_i \in \pi \backslash \pi_s$. Moreover, $T_s = [\bar{P}_s \mid \bar{Q}_s]$ is a similarity transformation that gives us the observability decomposition of the system.

Proof: Let P_s be the characteristic matrix corresponding to the NEP π_s in $\mathcal{G}_s$. Since $\mathcal{G}_s$ is a connected graph, from Corollary 5.3, we know that $\mathcal{L}_s$ is similar to a diagonal matrix by the transformation T_s, i.e.,

$$\bar{\mathcal{L}}_s = T_s^T \mathcal{L}_s T_s = \begin{bmatrix} \mathcal{L}_{sP} & \mathbf{0} \\ \mathbf{0} & \mathcal{L}_{sQ} \end{bmatrix}. \tag{15}$$

Next, we want to show that the same transformation on C can give us desired block structure. In other words, we want

[5]In the leader-follower structure, some agents in a multi-agent group assume the leaders' role and the rest of the group are followers. The leaders, with enhanced capabilities, are driven by some exogenous signals, while the follower stick to the consensus algorithm.

to show that

$$\bar{C} = CT_s = [C_o, \mathbf{0}]. \quad (16)$$

From Lemma 4.4, we know that $\bar{Q}_s$ have zero column sum and that the columns could have nonzero elements only in the rows associated with nontrivial cells. On the other hand, row i of C denotes how the sensor nodes are connected to central node i, thus it would have a 1 in j-th element if there is an edge between central node i and sensor node j. Since, under π, all the nontrivial cells are in $\mathcal{G}_s$, and all the central nodes are trivial partitions, i.e., each leader is either connected to all the nodes in C_i, for some $i = 1, 2, \ldots, r_1$, or none. Thus columns of C^T are in the column space of P_s. Therefore, we have

$$C\bar{Q}_s = \mathbf{0},$$

and the transformed system can be given as

$$\begin{cases} \dot{v} &= -\bar{\mathcal{L}}_s\ v, \qquad v(0) = T_s^T q, \\ y &= \bar{C}\ v, \end{cases} \quad (17)$$

where $v = T_s^T x$ is the transformed state vector, and $\bar{\mathcal{L}}_s$ and $\bar{C}$ are transformed matrices given in (15) and (16) respectively. Furthermore we can state that

$$\dot{v}^o = -\mathcal{L}_{sP}\ v^o \quad (18)$$

$$\dot{v}^{uo} = -\mathcal{L}_{sQ}\ v^{uo} \quad (19)$$

$$y = C_o v^o \quad (20)$$

where $v^o = \bar{P}_s^T x$ is the observable part of the transformed state, while $v^{uo} = \bar{Q}_s^T x$ is the unobservable part of the transformed state. ■

Remark 5.6: If we let $x^\pi = P_s^+ x$, where $P_s^+ = (P_s^T P_s)^{-1} P_s^T$ is the pseudo inverse of P_s, and multiply both side of (18) by $(P_s^T P_s)^{-\frac{1}{2}}$ we get

$$\begin{aligned} \dot{x}^\pi &= P_s^+ \mathcal{L}_s P_s P_s^+ x \quad (21) \\ &= \hat{\mathcal{L}}_s P_s^+ x \\ &= \hat{\mathcal{L}}_s x^\pi. \end{aligned}$$

To this end, we have derived the sufficient condition for a DSN not to be completely controllable. The converse of this theorem gives us an necessary condition for the DSN to be fully observable. We state this as a corollary:

Corollary 5.7: Given a connected sensor graph $\mathcal{G}_s$ and its augmented graph $\mathcal{G}$, a necessary condition for $(-\mathcal{L}_s, C)$ to be observable is that no NEPs π and π_s exist over $\mathcal{G}$ and $\mathcal{G}_s$ such that π and π_s share all nontrivial cells.

REFERENCES

[1] P. Ting and R. Iltis, "Multitarget motion analysis in a dsn," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, no. 5, pp. 1125–1139, Sept.-Oct. 1991.

[2] S. M. Brennan, A. M. Mielke, and D. C. Torney, "Radioactive source detection by sensor networks," *IEEE Transactionas on Nuclear Science*, vol. 52, no. 3, June 2005.

[3] J. Chen, K. Yao, and R. Hudson, "Source localization and beamforming," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 30–39, Mar. 2002.

[4] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 41–49, Aug 2004.

[5] V. Gupta, "Distributed estimation and control in networked systems," Ph.D. dissertation, California Institute of Technology, June 2006.

[6] Z. Jin and R. M. Murray, "Multi-hop relay protocols for fast consensus seeking," in *45th IEEE Conference on Decision and Control*, San Diego, California, 13-15, Dec, 2006.

[7] H. Qi, S. Iyengar, and K. Chakrabarty, "Multiresolution data integration using mobile agents in distributed sensor networks," *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, vol. 31, no. 3, pp. 383–391, Aug 2001.

[8] J. Kurhinen, "Mp2p network in collecting data from sensor networks," in *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, June 2006, pp. 26–29.

[9] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference*, Dec. 2005.

[10] V. Gupta, D. Jeffcoat, and R. M. Murray, "On sensor coverage by mobile sensors," in *Proceedings of the IEEE Conference on Decision and Control*, Dec 2006.

[11] V. Gupta, A. F. Dana, J. P. Hespanha, and R. M. Murray, "Data transmission over networks for estimation," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, (MTNS 06)*, June 2006.

[12] C. Godsil and G. Royle, *Algebraic graph theory*. Springer, 2001.

[13] W. Ren and R. W. Beard, "A survey of consensus problems in multi-agent coordination," in *American Control Conference*, Portland, OR, June 2005, pp. 1859–1864.

[14] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Contr.*, vol. 48, no. 6, pp. 988–1001, June 2003.

[15] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in d dimension," *IEEE Trans. Robot. Automat.*, vol. 51, no. 8, pp. 1289–1298, 2006.

[16] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Trans. Automat. Contr.*, vol. 49, no. 4, pp. 622–629, 2004.

[17] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automat. Contr.*, vol. 49, pp. 1465–1476, Sept 2004.

[18] R. W. Beard, J. R. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Trans. Contr. Syst. Technol.*, vol. 9, no. 6, pp. 777–790, Nov. 2001.

[19] J. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc.IEEE Int. Conf. Robot. Automat.*, Leuven, Belgium, May 1998, pp. 2864–2869.

[20] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Contr.*, vol. 49, pp. 1520–1533, Sept 2004.

[21] M. Mesbahi, "State-dependent graphs," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii USA, Dec. 2003, pp. 3058–3063.

[22] R. Olfati-Saber and R. M. Murray, "Agreement problems in networks with directed graphs and switching toplogy," in *Proceedings of the 42^{nd} IEEE Conference on Decision and Control 2003*, vol. 4, Maui, Hawaii USA, Dec. 2003, pp. 4126–4132.

[23] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part II : Dynamic topology," in *Proceedings of the 42^{nd} IEEE Conference on Decision and Control*, 2003, pp. 2016–2021.

[24] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Contr.*, vol. 51, no. 3, pp. 401–420, March 2006.

[25] D. Luenberger, *Optimization by vector space methods*. Wiley, 1969.

[26] M. Ji and M. Egerstedt, "A graph-theoretic characterization of controllability for multi-agent systems," accepted by American Control Conference, New York, NY, July 2007.