Network Discovery: An Estimation Based Approach

Girish Chowdhary, Magnus Egerstedt, and Eric N. Johnson

Abstract— **We consider the unaddressed problem of network discovery, in which, an agent attempts to formulate an estimate of the global network topology using only locally available information. We show that under two key assumptions, the network discovery problem can be cast as a parameter estimation problem. Furthermore, we show that some form of excitation must be present in the network to be able to converge to a solution. The performance of two methods for solving the network discovery problem is evaluated in simulation.**

I. INTRODUCTION

Successful negotiation of real world missions often requires diverse teams to collaborate and synergistically combine different capabilities. The problem of controlling such networked teams has become highly relevant as advances in sensing and processing enable compact distributed systems with wide ranging applications, including networked Unmanned Aerial Systems (UAS), decentralized battlefield negotiation, decentralized smart-grid technology, and internet based social-networking (see for example [15], [11], [2], [5], and [14]). The development of these systems however, present many challenges as the presence of a central controlling agent with access to all the information cannot be assumed.

There have been significant advances in control of networked systems using information available only at the agent level, including reaching consensus in networked systems, formation control, and distributed estimation (see for example [15], [5]). The emphasis has been to rely only on local interactions to avoid the need for a central controlling agent. However, there are many applications where the knowledge of the global network topology is needed for making intelligent inferences. Inferences such as identifying the interactions between agents, identifying faulty or misbehaving agents, or identifying agents that enjoy high connectivity and are in a position to influence the decisions of the networked system. This information in turn, can allow agents to make intelligent decisions about how to control a network and how to build optimal networks in real-time. The key problem that needs to be addressed for enabling the needed intelligence is: How can an agent use only information available at the agent level to make global inferences about the network topology? We term this problem as Network Discovery, and formulate the problem in the framework of estimation theory.

The idea of using measured information to make inferences about the network characteristics was explored by Franceschelli et al. through the estimation of the eigenvalues of the network graph Laplacian [6]. They proposed a decentralized method for Laplacian eigenvalue estimation by providing an interaction rule that ensured that the state of the agents oscillate in such a manner such that the problem of eigenvalue estimation can be reduced to a problem of signal processing. The eigenvalues are then estimated using Fast Fourier Transforms. The Laplacian eigenvalues contains useful information that can be used to characterize the network, particularly the second eigenvalue of the Laplacian contains information on the connectivity of the network and how fast it can reach agreement. However, the knowledge of eigenvalues does not yield information about other details of the topology, including the degree of connectivity of individual agents and the graph adjacency matrix.

Agent level measurements of other agents' states was used by Franceschelli, Egerstedt, and Giua for fault detection through the use of motion probes [7]. The idea behind motion probes is that individual agents perform in a decentralized way a maneuver that leaves desirable properties of the consensus protocol invariant and analyze the response of others to detect faulty or malicious agents. This work emphasized the importance of excitation in the network states for network property discovery.

Muhammad and Jabdabaie have proposed using Gossip-like algorithms for minimizing communications overhead in discovering network properties through relayed information [11], while Abdolyusefi and Mesbahi have proposed a node knockout procedure [12] for identifying network topology. These algorithms rely on the internal communication in the network to relay relevant information to identify the network topology. There are various situations however, where such communication may not be possible or cannot be trusted. For example, communications based approach cannot work if some of the agents have become faulty, are unable to communicate, are maliciously relaying wrong information, or if the aim is to covertly discover the network topology of a (possibly unfriendly) network. In this paper, we do not assume access to the networks internal communication protocol, and concentrate on the development of network discovery algorithms that use only measured or sensed information at the agent level. Clearly the addition of communications would compliment any of the presented approaches.

Finally, we mention that the problem we are concerned with is quite different from that of distributed estimation (see for example reference [9] and the references therein). In distributed estimation the purpose is to reach consensus

This work was supported in part by NSF ECS-0238993 and NASA Cooperative Agreement NNX08AD06A. G. Chowdhary and A. Prof. Eric Johnson are with the school of aerospace engineering while Prof. Magnus Egerstedt is with the school of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta GA, Girish.Chowdhary@gatech.edu, magnus.egerstedt@ece.gatech.edu, Eric.Johnson@ae.gatech.edu

about the value of an external global quantity in a decentralized manner through distributed measurements over different agents, whereas here we are concerned with the estimation of internal network properties (particularly the rows of the graph Laplacian) through measurements.

In this paper We show that under two key assumptions the problem of network discovery can be related to that of parameter estimation. We also show that some form of excitation must be present or inserted into the system to be able to solve this problem using only local information. Furthermore, we propose and compare various methods that an agent can use for network discovery. We rely heavily on an algebraic graph theoretic representation of networked systems, where the network and its interconnections are represented through sets. We begin by showing that the problem of identifying a particular agents degree of connectivity and neighbors can be reduced to that of estimating that agent's linear consensus protocol. We then show that subject to two assumptions, namely static network, and complete availability of information, this problem can be cast as that of parameter estimation and propose three different methods to solve the problem online.

II. THE NETWORK DISCOVERY PROBLEM

Consider a network consisting of N independent agents enabled with limited communication capabilities and operating under a protocol to reach consensus [15]. We assume that the information available to an agent is composed entirely of what it can sense or measure. A network such as this is capable of representing a wide variety of decentralized networked dynamical systems, including a collaborating group of mobile robots, a power grid, or computer systems connected over ethernet. Such a network can be represented as a graph $G = V \times E$, with $V = 1, ..., N$ denoting the set of vertices or nodes of the network, and E denoting the set of edges $E \subset V \times V$, with the pair $(i, j) \in E$ if and only if the agents i can communicate with or otherwise sense the state of agent j. In this case, agent j is termed as a neighbor of agent i. The total number of all neighbors of an agent at time t is termed as its degree at time t. Let $Z_i \in \Re^n$ denote the state of the i^{th} agent, with $Z_i = \{z_1, z_2, z_3, ..., z_n\}$. The elements of Z_i can represent various physical quantities of interest, such as position, velocity, voltage etc. If the elements of the edge set (that is the pairs (i, j)) are unordered, the graph is termed as undirected. We will consider undirected graphs in this paper for ease of exposition, we note that an extension to the directed case is straightforward.

In the following, we will refer to the agent whose degree and neighbors are to be estimated as the *target agent*, while the agent that wish to estimate the network structure of the target agent as the *estimating agent)*. The problem of network discovery can now be formulated:

Problem 1 The Network Discovery Problem Use only the information available at estimating agent to determine the degree of the target agent and identify its neighbors.

Note that multiple target and estimating agents will be considered in future work. We now introduce a simplification in the notation, namely, when only one component of z_i is under consideration its identifying subscript will be dropped. Using this convention, let the vector $x = \{x_1, x_2, ..., x_N\} \in \Re^N$ contain the i^{th} element $z_i \in \Re$ of all agents. We assume that the dynamics of the target agent (agent i) is given by the following equation [5]:

$$\dot{x}_i(t) = \sum_{j \in N_i} \left[x_i(t) - x_j(t)\right], \tag{1}$$

where N_i is the set of agent i's neighbors and the mapping $y_i(t) = \sum_{j \in N_i} [x_i(t) - x_j(t)]$ denotes the un-weighted consensus protocol of agent i [15], [5]. The preceding equation basically states that $y_i = \dot{x}_i$, and we will often drop the subscript i on y for notational convenience. Let $\zeta \in \Re^l$ denote the vector containing the states of all of agent i's neighbors where $l < N$ denotes the degree of agent i. Note that with an arbitrary numbering of the agents, the state vector x can be written as $x = [\zeta, \xi]^T$, where $\xi \in \Re^{N-l}$ is the vector containing the states of all the agent's in the networks which are not agent i's neighbors. Therefore, y can be also expressed as: $y = W^T x$, where the vector $W \in \Re^N$ is the i^{th} row of the instantaneous graph Laplacian [5]. Taking advantage of this fact, we denote W as the Laplacian vector of agent i. Under conditions on connectivity of the network, the consensus protocol will result in $x \rightarrow \frac{1}{N}\mathbf{1}\mathbf{1}^T x(0)$, where $\mathbf{1} = [1, 1, 1, 1..1]^T \in \Re^N$ [5]. In this paper however, we are not concerned with the convergence properties of the consensus protocol. What we are concerned with, is the problem of estimating agent i's degree and neighbors (problem 1. Figure 1 depicts a network discovery scenario where the estimating agent can sense the states of the target agent and all of its neighbors, but not all of the agents in the network.

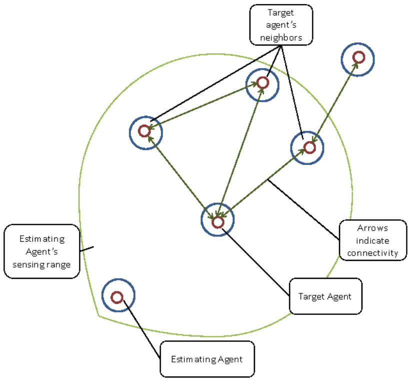
Target agent's neighbors
Estimating Agent's sensing range
Arrows indicate connectivity
Target Agent
Estimating Agent

Fig. 1. A depiction of the network discovery problem, where the estimating agent uses available measurements to estimate the neighbors and degree of the target agent. Note that the estimating agent can sense the states of the target agent and all of its neighbors, however, one agent in the target agent's network is out of the estimating agent's sensing range. Furthermore, note that multiple estimating agents and target agents may be present in a network.

III. POSING NETWORK DISCOVERY AS AN ESTIMATION PROBLEM

Obtaining a solution to problem 1 in the most general case can be a quite daunting task due to a number of reasons, including:

- The neighbors of the target agent may change with time,
- The estimating agent may not be able to sense information about all of target agent's neighbors,
- The target agent may be actively trying to avoid identification of its consensus protocol.

In order to progress, we will make the following simplifying assumption.

Assumption 1 Assume that the network edge set does not change for a predefined time interval $\Delta(t)$, that is the network is *slowly varying*.

The above assumption requires that within a time interval $\Delta(t)$, $W(t) = W$, that is the Laplacian vector $W(t)$ is time invariant for a predefined amount of time, which amounts to"slow" variation in the network topology. Such slowly varying networks can be used to model many real-world networked systems. This assumption allows us to cast the problem of network discovery as a problem of estimating the constant Laplacian vector of the target agent over a time interval. The Laplacian vector contains the information about the degree of agent i and its adjacency to other agents in the network, information that can be used to solve the network discovery problem. Let $\bar{x} \in \Re^k$ contain the measurements of the states of agents that are available to the estimating agent. Note that without loss of generality we can assume that the agents whose states the target agent can measure are bounded above by the total number of agents in the network, i.e. $k \leq N$; for if $k > N$, that is when not all agents whose measurements are available are part of the network, then we can always set $N = k$. Then, letting $\hat{W} \in \Re^k$ the following estimation model can be used for estimating W:

$$\nu(t) = \hat{W}^T(t)\bar{x}(t). \tag{2}$$

Recalling that $y(t) = W^T(t)x(t)$ the estimation error can be formulated as:

$$\epsilon(t) = \nu(t) - y(t) = \hat{W}^T(t)\bar{x}(t) - W^T x(t). \tag{3}$$

One way to approach the network discovery problem, is to design a weight law $\dot{\hat{W}}(t)$ such that $\epsilon(t) \to 0$ uniformly in finite time, that is $\epsilon(t)$ is identically equal to zero after some time T ($\epsilon(t) \equiv 0 \ \forall t > T$). The following proposition shows that when only a single estimating agent it used, then if the estimating agent cannot measure the states of all of the target agent's neighbors, then $\epsilon(t)$ cannot be identically equal to zero.

Proposition 1 Consider the estimation model of (2) and the estimation error ϵ of (3), and suppose $\bar{x}$ does not contain the state measurements of all of the target agent's neighbors, then $\epsilon(t)$ cannot be identically equal to zero.

Proof: Ignoring the irrelevant case when the target agent has no neighbors, let $\zeta \in \Re^m$ denote the vector containing all of target agent's neighbors. Then letting i denote the identifying subscript for the target agent, and deg_i denote the degree of i we have that $y(t) = \dot{x}_i(t) = [-1, -1, ..., deg_i, ..., -1]^T\zeta(t) = \check{W}^T\zeta(t)$. Therefore the vector $\check{W} \in \Re^m$ contains only nonzero elements. Let $\bar{x} \in \Re^k$, and assume that $k < m$ (the case when $k > m$ follows in a similar manner), furthermore, let $\zeta = [\bar{x}, \xi]$, with $\xi \in \Re^{m-k}$. Suppose *ad absurdum* $\epsilon(t)$ is identically equal to zero, then we have that:

$$\nu(t) - y(t) = [\hat{W}(t), 0..0]^T[\begin{array}{c}\bar{x}(t)\\ \xi(t)\end{array}] - \check{W}\zeta(t) = 0. \tag{4}$$

Since we claim that $\epsilon(t)$ is identically equal to zero, then in the nontrivial case (i.e. $\zeta(t) \neq 0$) we must have that $[\hat{W}(t), 0..0] - \check{W} = 0$, for all $t > T$ in order to satisfy (4). Therefore $\check{W}$ must contain $m - l$ zero elements, which contradicts the fact that $\check{W}$ contains only nonzero elements. Hence, if $\bar{x}$ does not contain the state measurements of all of the target agent's neighbors, then $\epsilon(t)$ cannot be identically equal to zero. ■

Remark 1 Note that in the above proof we ignored the case when $\zeta(t)$ is identically equal to zero. If $\zeta(t)$ is identically equal to zero then the states of all agents have converged to the origin, an unlikely prospect, considering the consensus law only guarantees $x \to span(\mathbf{1})$. Another unlikely but interesting case arises when $\zeta(t)$ is such that $[\hat{W}(t), 0..0] - \check{W} \perp \zeta(t) \ \forall t > T$. In both these cases, one can argue that $\zeta(t)$ does not contain sufficient excitation, and proposition 1 becomes irrelevant. The importance of excitation in the states for solving the network discovery problem is explored further in section III-A.

Remark 2 Proposition 1 formalizes a fundamental obstruction to obtaining a solution to the problem of network discovery: If the estimating agent cannot measure or otherwise know the states of the target agent's neighbors, then an estimation based approach with only one estimating agent cannot be used to solve the network discovery problem. Future work will consider multiple estimating agents.

Therefore, we have shown that in order to use the estimation model of (2) to solve the network discovery problem with one estimating agent, the following assumption must be satisfied:

Assumption 2 Assume that the estimating agent can measure or otherwise perceive the position of all of the target agent's neighbors.

The following theorem shows that if a weight update law $\dot{\hat{W}}(t)$ exists such that $\epsilon(t)$ can be made identically equal to zero, then a solution to the network discovery problem (problem 1) can be found.

Theorem 1 Suppose assumption 2 is satisfied, and $x(t)$ is not identically equal to zero, then finding a weight update law $\dot{\hat{W}}(t)$ such that $\epsilon(t)$ becomes identically equal to zero (that is $\epsilon(t) = 0 \ \forall t > T$), is equivalent to finding a solution to the network discovery problem for the case of static networks (assumption 1).

Proof: Suppose there exists a weight update law $\dot{\hat{W}}(t)$ exists such that $\epsilon(t)$ becomes identically equal to zero. Since assumption 2 holds, we can arbitrarily reorder the states such that $\bar{x} = [\zeta, \xi]$, where ξ denote the states of the agents which are not neighbors of the target agent, hence we have:

$$\nu - y = \hat{W}^T(t)\bar{x}(t) - [W, 0..0]^T[\begin{array}{c}\zeta\\ \xi\end{array}] = 0. \tag{5}$$

Letting $\tilde{W} = \hat{W} - [W, 0..0]$, we have:

$$\nu(t) - y(t) = \tilde{W}(t)\bar{x}(t) = 0. \tag{6}$$

Since $x(t)$ is assumed to be not identically equal to zero, in the nontrivial case we must have that $\tilde{W}(t) = 0 \ \forall t > T$. Therefore it follows that $\hat{W} = [W, 0..0]$ contains the Laplacian vector of the target agent, which is sufficient to identify the degree and neighbors of the target agent. ■

Remark 3 As in the proof of proposition 1, an interesting but unlikely case arises when $\tilde{W}(t) \perp \bar{x}(t)$. Once again this relates to a notion of sufficient excitation in the system states and is further explored in section III-A.

To simplify the notation, we can let $\bar{x} = x$. Due to theorem 1, this is equivalent to saying that for the purpose of the network discovery problem, the network can be assumed to be made of only the agents that either interact with the target agent or are visible to the estimating agent. Hence, this change in notation does not affect the structure of the problem, except that we now have $\epsilon(t) = \nu(t) - y(t) = \hat{W}^T(t)x(t) - W^T x(t) = \tilde{W}x$, which is simpler to deal with. In this case, the Laplacian vector of the target agent W will contain zero elements corresponding to agents that the target agent is not connected to.

Through the above discussion ,we have essentially shown that subject to assumption 1 and 2 the network discovery problem can be cast as the following simpler parameter estimation problem:

Problem 2 Let an estimation model for the network discovery problem be given by (2), and the estimation error be given by (3). Design an update law $\dot{\hat{W}}$ such that $\hat{W}(t) \to W$ as $t \to \infty$.

Various approaches have been proposed for online parameter estimation, in the following we will highlight three such approaches.

A. Instantaneous Gradient Descent Based Approach

In this simplest and most widely studied approach for parameter estimation $\hat{W}$ is updated in the direction of maximum reduction of the instantaneous quadratic cost $V(\epsilon(t)) = \frac{1}{2}\epsilon^2(t)$. That is, letting Γ be a positive learning rate we have $\dot{\hat{W}} = -\Gamma\frac{\partial V}{\partial \hat{W}}$. This results in the following update law:

$$\dot{\hat{W}}(t) = -\Gamma x(t)\epsilon(t). \tag{7}$$

The convergence properties of the gradient descent based approach have been widely studied, it is well known that for this case a necessary and sufficient condition for ensuring $\hat{W} \to W$ as $t \to \infty$ exponentially is a condition on Persistency of Excitation (PE) in $x(t)$ [1],[13],[16]. Various equivalent definitions of excitation and the persistence of excitation of a bounded signals exist in the literature [1],[13]; we will use the definitions proposed by Tao in [16]:

Definition 1 A bounded vector signal $x(t)$ is persistently exciting if for all $t > t_0$ there exists $T > 0$ and $\gamma > 0$ such that

$$\int_t^{t+T} x(\tau)x^T(\tau)d\tau \geq \gamma I. \tag{8}$$

Note that definition 1 requires that the matrix $\int_t^{t+T} x(\tau)x^T(\tau)d\tau$ be positive definite over all predefined finite time intervals. If a signal satisfies this condition over only one such interval, it is called as exciting, but not persistently exciting. As an example, consider that in the two dimensional case, vector signals containing a step in every component are exciting, but not persistently exciting; whereas the vector signal $x(t) = [\sin(t), \cos(t)]$ is persistently exciting. Hence, in order to ensure that $\tilde{W} \to 0$, we must ensure that the system states $x(t)$ are persistently exciting. However, there is no guarantee that the network state vector $x(t)$ would be exciting if the network is only running the consensus protocol of (1). For example, the following fact shows that if the initial state of the network happens to be an eigenvector of the graph Laplacian, then the system states are not persistently exciting.

Fact 1 The solution $x(t)$ to the differential ($\dot{x}(t) = -Lx(t)$), where L is the graph Laplacian, need not be persistently exciting for all choices of $x(0)$.

Proof:

Let $x(0)$ and $\lambda \in \Re$ be such that $Lx(0) = \lambda x(0)$, that is let $x(0)$ be an eigenvector of L. Then we have $x(t) = e^{-\lambda t}x(0)$, hence

$$\int_t^{t+T} x(\tau)x^T(\tau)d\tau = \int_t^{t+T} e^{-2\lambda t}x(0)x^T(0), \tag{9}$$

which is at-most rank 1, and hence not positive definite over any interval. ■

Therefore, an external forcing term will be needed to enforce PE in the system. The consensus protocol can then be written as:

$$\dot{x}_i(t) = \sum_{j \in N_i} \left[x_i(t) - x_j(t) + f(x_i(t), t)\right], \tag{10}$$

where $f(x_i(t), t)$ is a known bounded mapping $\Re^2 \to \Re$ used to insert excitation into the system. In its most simplest form $f(x_i(t), t)$ can simply be a random sequence of numbers, or it could be an elaborate periodic pattern (such as in [7]) which is known over the network.

We evaluate the performance of this algorithm through simulation on a network containing 9 nodes with each of the nodes updated by (10), for solving the network discovery problem. It is assumed that $f(x_i(t), t)$ is a known Gaussian random sequence with an intensity of 0.01 and that $y_i(t) = \dot{x}_i(t) - f(x_i(t), t)$ can be measured. Note that the chosen $f(x_i(t), t)$ does introduce persistent excitation. The agents are arbitrarily labeled, and the third agent is picked as the estimating agent, and it estimates the consensus protocol for the second agent (which is the target agent). The Laplacian vector for the target agent is given by $W = [0, -3, 1, 0, 0, 1, 1, 0, 0]$, and its consensus protocol will have the form $y_i = W^T x$. The target agent has 3 neighbors (i.e. degree of i is 3), they are agent $3, 6$, and 7. Figure 2 shows the performance of the gradient descent algorithm for the network under consideration with $\Gamma = 10$. It can be seen that the algorithm is unsuccessful in estimating the Laplacian vector for W by the end of the simulation, even when persistent excitation is present. Increasing the learning rate Γ may slightly speed up the convergence, however the key condition required is that the $x(t)$ remain persistently exciting such that the scalar γ in definition 1 is large. That is, the convergence is dependent not only on the existence of excitation, but also on its magnitude.

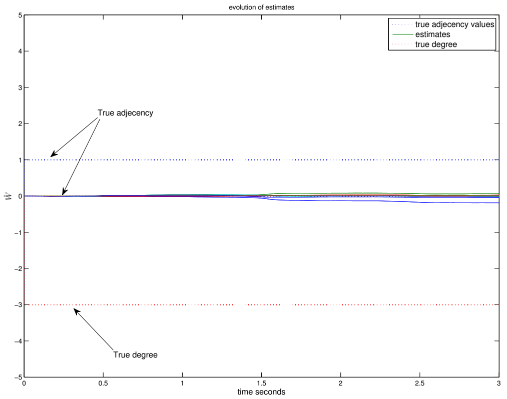

Fig. 2. Evolution of the estimate of the Laplacian vector ($\hat{W}$) for network discovery using gradient descent. Note that the estimates do not converge to the actual values depicted by dotted lines. The results go to show that the convergence of the gradient descent method is dependent not only on the presence of persistence excitation but also on its magnitude.

B. *Concurrent Gradient Descent Based Approach*

The gradient descent algorithm of Section III-A is susceptible to being stuck at local minima, and requires PE to guarantee convergence. For many networked control applications the condition on PE is infeasible to monitor online, particularly since the trajectories of individual agents are not known a-priori. On examining (7) we see that the update law uses only instantaneously available information $(x(t), \epsilon(t))$ for estimation. Chowdhary and Johnson have noted that if the update law uses specifically selected and recorded data concurrently with current data for adaptation, and if the recorded data were sufficiently rich, then intuitively it should be possible to guarantee $\hat{W} \to W$ as $t \to \infty$ without requiring persistently exciting $x(t)$. This results in a Concurrent Learning gradient descent algorithm [4], [3]. Let $j \in \{1, 2, ...p\}$ denote the index of a stored data point x_j, let $\epsilon_j = \tilde{W}^T x_j$, let $\Gamma > 0$ denote a positive definite learning rate matrix, then the concurrent learning gradient descent algorithm is given as:

$$\dot{\hat{W}}(t) = -\Gamma x(t)\epsilon(t) - \sum_{i=1}^{p} \Gamma x_j \epsilon_j. \tag{11}$$

The parameter error dynamics $\tilde{W}(t) = \hat{W}(t) - W$ for this case can be expressed as follows:

$$\begin{aligned} \dot{\tilde{W}}(t) &= -\Gamma x(t)\epsilon(t) - \Gamma \sum_{j=1}^{p} x_j \epsilon_j \\ &= -\Gamma x(t) x^T(t) \tilde{W}(t) - \Gamma \sum_{j=1}^{p} x_j x_j^T \tilde{W}(t) \\ &= -\Gamma [x(t) x^T(t) + \sum_{j=1}^{p} x_j x_j^T] \tilde{W}(t). \end{aligned} \tag{12}$$

The concurrent use of current and recorded data has interesting implications, as the exciting term $f(x_i, t)$ will not need to be persistently exciting, but only exciting over a finite period over which rich data can be recorded. In fact, Chowdhary and Johnson have also shown that the recorded data need only be linearly independent in order to guarantee weight convergence [4]. This condition on sufficient richness of the recorded data is captured in the following statement:

Condition 1 The recorded data has as many linearly independent elements as the dimension of x. That is, if $Z = [x_1, ..., x_p]$, then $rank(Z) = N$.

This condition is easier to monitor online and essentially requires that the recorded data contain sufficiently different elements to form the basis of the state space. The following theorem is proven in [4]:

Theorem 2 If the recorded data points satisfy condition 1, then the zero solution of parameter error dynamics $\tilde{W} \equiv 0$ of (12) is globally exponentially stable when using the concurrent learning gradient descent weight adaptation law of (11).

We now evaluate the performance of the concurrent learning gradient descent algorithm on the networked system simulation described in section III-A. Figure 3 shows the performance of the concurrent gradient descent algorithm for the network with $\Gamma = 10$. The simulation began with no recorded points, at each time step, the state vector $x(t)$ was scanned online, and points satisfying the condition $\|Z^T x(t)\| < 0.5$ or $y(t) - \nu(t) > 0.3$ were selected for storage. Condition 1 was found to be satisfied within 0.1 seconds into the simulation. It can be seen that the algorithm is successful in estimating the Laplacian vector for W, and thus in estimating the degree of the third agent and the identity of its neighbors. Hence, the algorithm outperforms the traditional gradient descent based method (section III-A) with the same level of enforced excitation. In general, the speed of convergence will be dependent on the minimum eigenvalue of the matrix ZZ^T and to a lesser extent, the learning rate Γ. That is, ideally we would like the stored data to not only be linearly independent, but also be sufficiently different in order to maximize the minimum singular value of Z. At the end of the simulation the minimum singular value was found to be 1.58.

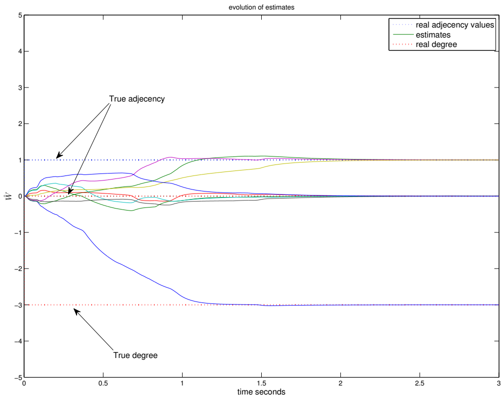

Fig. 3. Evolution of the estimate of the Laplacian vector ($\hat{W}$) for network discovery using concurrent gradient descent. Note that the estimates converge to the actual values (shown using dotted lines) within 2 seconds of the simulation. The converged estimates of the Laplacian vector directly yield the degree and the neighbors the target agent.

C. Least Squares based Approaches

Recursive least squares, or equivalently a Kalman filter based implementation, can be used to solve the estimation problem. In this approach a recursive law is developed such that a quadratic cost of the integral of the estimation error is minimized [8], [1], [10]. To achieve this using assumption 1 an update model for the estimate of the Laplacian vector $\hat{W}$ as $\dot{\hat{W}} = 0$, and a Kalman filter is designed to estimate $\hat{W}$ using a measurement model $\nu = \hat{W}^T x$ and the estimation error $y - \hat{W}^T x$. The benefit of this approach is that the solution can be shown to be optimal in the least squares sense, and noise in measurements can be handled. The downside is that the method is computationally expensive as the covariance matrix must also be propagated. Furthermore, PE is required to guarantee convergence [1], [16].

IV. Conclusion

In this paper we considered the problem of network discovery, in which, an agent uses locally available information to estimate the global topology of a networked system attempting to reach consensus. We showed that subject to two key assumptions, the network discovery problem can be cast as a parameter estimation problem and the elements of the graph Laplacian can be estimated in real-time. The graph Laplacian contains the adjacency and degree information for a given agent, and is sufficient to form an estimate of the network topology. The first assumption requires that the network is slowly varying, that is, it requires the network topology to remain static over a predefined time interval. The second assumption requires that the estimating agent can measure (or otherwise know) the states of all of target agent's neighbors. In fact, we showed that if not satisfied, this assumption forms a major obstruction to solving the network discovery problem using only one estimating agent. We discussed three methods for solving the network discovery problem in the parameter estimation framework, and compared the performance of two in simulation. We noted that the concurrent gradient descent method requires far less excitation than the traditional gradient descent method, and has improved convergence. In conclusion, we note that regardless of what parameter estimation method is used to solve the network discovery problem, some amount of excitation must be inserted into the networked system for converging to a solution.

V. Acknowledgments

References

[1] Karl Johan Aström and Björn Wittenmark. *Adaptive Control*. Addison-Weseley, Readings, 2 edition, 1995.

[2] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[3] Girish Chowdhary. *Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010.

[4] Girish Chowdhary and Eric N. Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *49th IEEE Conference on Decision and Control*, 2010.

[5] Magnus Egerstedt and Mehran Mesbahi. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[6] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu. Decentralized laplacian eigenvalues estimation of the network topology of a multi-agent system. In *IEEE Conference on Decision and Control*, 2009.

[7] M. Franceschelli, Egerstedt M., and A. Giua. Motion probes for fault detection and recovery in networked control systems. In *American Control Conference*, 2008.

[8] Arthur Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, 1974.

[9] Vijay Gupta. *Distributed Estimation and Control in Networked Systems*. PhD thesis, California Institute of Technology, 2006.

[10] Ravindra V. Jategaonkar. *Flight Vehicle System Identification A Time Domain Approach*, volume 216 of *Progress in Astronautics and Aeronautics*. American Institute of Aeronautics and Astronautics, Reston, 2006.

[11] A. Muhammad and A. Jadbabaie. Decentralized computation of homology groups in networks by gossip. In *American Control Conference*, 2007.

[12] Marzieh Nabi Abdolyousefi and Mehran Mesbahi. Network identification via node knockout. In *IEEE Conference on Decision and Control*, 2010.

[13] Kumpati S. Narendra and Anuradha M. Annaswamy. *Stable Adaptive Systems*. Prentice-Hall, Englewood Cliffs, 1989.

[14] Office of the Secretary of Defense. Unmanned aircraft systems roadmap 2005-2030. Technical report, Department of Defense, August 2005.

[15] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215 –233, jan. 2007.

[16] Gang Tao. *Adaptive Control Design and Analysis*. Wiley, New York, 2003.