Identifying cyber attacks under local model information

Fabio Pasqualetti, Ruggero Carli, Antonio Bicchi, and Francesco Bullo

Abstract— **This work considers the problem of detecting corrupted components as well as external attacks in a large scale decentralized system. The electric power system, the transportation system, and generally any computer or network system are examples of large scale systems for which cyber-attacks have become an important threat. Despite the recent advances in the theory of detection and identification of misbehaving parts, the existing procedures to enforce security are still computationally inefficient and numerically unreliable. We consider the case of linear networks, and we model an external attack as an exogenous input affecting the system. We exploit two complementary methods that relies on two different sets of assumptions to reduce the complexity of the misbehavior detection and identification. The first method takes advantage of the presence in the network of weakly interconnected subparts, it requires the agents to have only a limited knowledge of the network model, and it affords local identification of the misbehaving agents whose behavior deviates more than a threshold. The second method relies on the presence of a set of trustworthy agents (leaders) with better computation and communication capabilities. Only relying on a partial knowledge of the network model, the leaders cooperatively detect and identify the misbehaving agents. The proposed methods are shown to improve the detection time, the numerical reliability, and the computational cost of existing algorithms.**

I. Introduction

The increasing reliance on network systems to support critical operations in defense, electric power management, and telecommunication raises the issue of the reliability and the robustness of such systems against external attacks. Because of the decentralized nature of network systems, cyber attacks compromising the availability of resources, the integrity of data, or the confidentiality of information are easily launched by a malignant agent. Unfortunately, the growing dimension of network systems forbids any centralized implementation of an attack detection system, ruling out classical solutions as presented in [1], [2].

The distributed computation of an agreement on a variable of interest is among the fundamental tasks to be accomplished by the member of a distributed system. In this work, we consider linear discrete time consensus algorithms as described in [3], and we allow for the presence of misbehaving agents which interfere with the nominal behavior of the network. Following [4], we consider the extreme case of Byzantine agents, which have complete knowledge of the system structure and state, and which collude in order to cause the biggest damage to the network.

The detection and the identification of misbehaving agents in a network has been the subject of intensive study, see e.g. [5], [6], and it is now well known that the network connectivity[1] determines the ability of a network to sustain arbitrary malfunctioning. Precisely, if the attack is driven by an omniscient adversary, then the total number of misbehaving components needs to be less than one-half of the network connectivity, and less than one-third of the number of processors for the detection to take place. In the last few years, the problem of reaching consensus in the presence of misbehaving agents has been revisited from a control theoretic perspective. In this works, the network is assumed to evolve as a linear dynamical system, and the misbehaving agents are modeled as unknown and unmeasurable inputs. In [7] the problem of detecting and identifying misbehaving agents in a linear consensus network is first introduced, and a solution is proposed for the single faulty agent case. In [8], [9], the authors provide a policy that k malicious agents can follow to prevent some of the nodes of a $2k$-connected network from computing the desired function of the initial state, or, equivalently, from reaching an agreement. On the contrary, if the connectivity is $2k+1$ or more, then the authors show that generically the set of misbehaving nodes is identified independent of its behavior, so that the desired consensus is eventually reached. Finally, in [4] a complete characterization of the policies that make a set of misbehaving agents undetectable is given in terms of the zero dynamics of a linear system associated with the network, and the connection between the graph connectivity and the zero dynamics is explained. Despite the advances in the theoretical understanding of the detection and identification of misbehaving agents, efficient decentralized algorithms ensuring security against attacks are still missing. The procedures proposed so far rely indeed on an heavy combinatorial machinery to locate the attackers, they require every agent to have complete knowledge of the network structure, and they need a number of steps proportional to the cardinality of the network to converge. Therefore, although provably correct, the existing algorithms are practically applicable only when the dimension of the network is relatively small.

The main contribution of this work are as follows. We present two novel methods to reduce the computational cost of the existing detection and identification algorithms. The proposed procedures rely on two different sets of assumptions, and they can be complementarily or alternatively employed depending on the network model. The first method

This material is based in part upon work supported by ICB ARO grant DAAD19-03-D-0004 and AFOSR MURI grant FA9550-07-1-0528, and in part upon the EC Contract IST 224428 ”CHAT”.

Fabio Pasqualetti, Ruggero Carli, and Francesco Bullo are with the Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara {fabiopas, carlirug, bullo}@engineering.ucsb.edu

Antonio Bicchi is with the Centro I. R. “E. Piaggio”, Università di Pisa bicchi@ing.unipi.it

[1]The connectivity of a graph is the maximum number of disjoint paths between any two vertices of the graph. A graph is complete if it has connectivity $n-1$, where n is the number of vertices in the graph.

is designed to exploit the presence in a network of weakly interconnected subparts. We introduce the notion of network decentralization, in terms of relatively weakly connected subnetworks, and derive a sufficient condition on the consensus matrix that allows to identify a certain class of misbehaving agents under limited information on the network structure. The second method admits the presence of a subset of agents with better computation and communication capabilities (leaders), and it achieves exact detection and identification even when the entire network structure is not available to any of the leaders. Under the assumption that the leaders coincide with the vertices of a connected communication graph, two algorithms are proposed to distributively reconstruct the state of the network in the presence of an unknown input, and to detect the presence of a misbehaving agent. Both algorithms require only a limited knowledge of the network structure, and they are shown to converge in a finite number of steps. We conclude the paper by showing the effectiveness of our algorithms through a numerical study.

The rest of the material is organized as follows. Section II contains the problem setting and a brief review of the existing results. Section III describes our method to exploit the presence of weakly interconnected subnetworks, and Section IV contains an example. Sections V introduces the hierarchical structure we propose, and it contains our main results on the unknown input estimation problem and on the detection problem. Sections VI and VII contain respectively a numerical study and our conclusion.

II. Problem setup and previous results

Let G denote a directed graph with vertex set $V = \{1,\dots,n\}$ and edge set $E \subset V \times V$. The in-neighbor set of a node $i \in V$, i.e., all the nodes $j \in V$ such that the pair $(j,i) \in E$, is denoted with N_i. We let each vertex $j \in V$ denote an autonomous agent, and we associate a real number x_j with each agent j. Let the vector x contain the values x_j. A linear iteration over G is an update rule for x and it is described by the linear discrete time system

$$x(t+1) = Ax(t). \tag{1}$$

In this work, we focus on a particular class of linear iteration, in which the matrix A is row-stochastic and primitive. The matrix A is referred to as a consensus matrix, and the system (1) is called *consensus* system. Moreover, the graph G is referred to as the communication graph associated with the consensus system (1) or, equivalently, with the consensus matrix A.

We allow for some agents to update their state differently than specified by the matrix A by adding an exogenous input to the consensus system. Let u_i, $i \in V$, be the input associated with the i-th agent, and let u be the vector of the functions u_i. The consensus system becomes

$$x(t+1) = Ax(t) + u(t).$$

Definition 1 (Misbehaving agent) *An agent j is* misbehaving *if there exists a time $t \in \mathbb{Z}_{\geq 0}$ such that $u_j(t) \neq 0$, and it is* well-behaving *otherwise.*

Throughout the paper, let $K = \{i_1, i_2, \dots\} \subseteq V$ denote the set of misbehaving agents, let e_i be the i-th vector of the canonical basis, and let $B_K = [e_{i_1} \ e_{i_2} \ \cdots]$. The consensus system with misbehaving agents K assumes the form

$$x(t+1) = Ax(t) + B_K u_K(t). \tag{2}$$

We associate an output matrix C_j to each agent j, which describes the information about the state of the network that is directly available to j. In particular, $y_j(t) = C_j x(t)$, $C_j = [e_{n_1} \ \dots \ e_{n_p}]^T$, and $\{n_1,\dots,n_p\} = N_j$. Throughout the paper, let $\mathrm{Im}(A)$ and $\mathrm{Ker}(A)$ denote the range space and the null space defined by the matrix A. We now review the existing methods to detect and identify the misbehaving agents.

A. Existing work

We focus on the agent j, and we assume that it only relies on its local observations. Let Y_j^d denote the vector containing the output y_j from time 0 up to time d, and let

$$O_j^d = \begin{bmatrix} C_j \\ C_j A \\ \vdots \\ C_j A^{d-1} \end{bmatrix},$$

where A is the iteration matrix of (1). In [10] it is pointed out that the row-space of O_j^d characterizes the set of all calculable linear functionals for the agent j in d time steps. Clearly, if $\mathrm{Ker}(O_j^d) = 0$, then any desired function of the initial state can be evaluated by j in d steps. In [8], [4] the estimation problem is extended to include the presence of unknown and unmeasurable inputs affecting the network. Precisely, consider the system (2), and assume that B_K is known. It is observed that a necessary and sufficient condition for each node to estimate the state of the whole network without knowing the input u_K is that the triple (A, B_K, C_j), is strongly observable[2]. Moreover, if the system $(A, B_{K_1 \cup K_2}, C_j)$ results to be strongly observable for every possible pair K_1, K_2 of misbehaving agents, then it is shown that the state of the network can be recovered by j without knowing the input matrix. To see this, let

$$F_j^d(K) = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ C_j B_K & 0 & \ddots & \ddots & 0 \\ C_j A B_K & C_j B_K & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ C_j A^{d-2} B_K & C_j A^{d-1} B_K & \cdots & C_j B_K & 0 \end{bmatrix}.$$

Under the strong observability assumption of the triple $(A, B_{K_1 \cup K_2}, C_j)$ for all K_1, K_2, there exists an unique set K such that $Y_j^d = O_j^d x + F_j^d(K) U^d$, where x denotes the state of the network, and U^d contains the misbehaving input up to time d. Therefore, by combinatorially testing every possible set of misbehaving agents, both the state as well as the location of the misbehaving agents are estimated. In [4] it is

[2]A linear system is called strongly observable if for every initial condition x_0 and for every input function u the following holds: $y(t) = 0$ for all $t \geq 0$ implies $x_0 = 0$, see for instance [11].

observed that the above condition is equivalent to the absence of zero dynamics for the system $(A, [B_{K_1} \;\; B_{K_2}], C_j)$, and a method based on filters providing decoupling of the inputs is proposed to identify the misbehaving agents.

Although provably correct, the procedures proposed in [8], [4] to ensure trustworthy computation in the face of misbehaving agents rely on heavy assumptions. First, each well-behaving agent is required to know entirely the network model. Second, the complexity of the identification process is proportional to $\binom{|V|}{|K|}$, since every possible set of misbehaving agents needs to be checked in order to ensure a correct identification. Third and finally, the identification time d grows with the cardinality of the network, because, in the worst case, we need $d = |V| - 1$ to guarantee an exact estimation of the state.

III. Approximate detection and identification

We first review in this section a filter approach to the detection and isolation of misbehaving agents in a linear network, and we then characterize a condition that allows to achieve detection and identification even after reducing the assumptions of the classical approach. For easy of notation, we consider now the single misbehaving agent case. Let j be a well-behaving agent, and consider the problem of deciding whether the agent i_1 or the agent i_2 is misbehaving. Let the linear discrete time filter

$$\begin{aligned} w_{i_1}(t+1) &= F_{i_1} w_{i_1}(t) + E_{i_1} y_j(t), \\ r_{i_1}(t) &= M_{i_1} w_{i_1}(t) + H_{i_1} y_j(t), \end{aligned} \tag{3}$$

be such that $r_{i_1} \neq 0$ if and only if i_1 is misbehaving. It follows that the signal r_{i_1} allows to uniquely identify the misbehaving agent i_1 against the well-behaving agent i_2. By implementing a similar filter for each possible pair[3] of misbehaving agent, the presence of the misbehaving agent i_1 is finally assessed by the agent j. A technique to design the filter (3) can be found in [4], where the knowledge of the network matrix A by the well-behaving agent j is assumed.

We consider now the case in which each well-behaving agent has a partial knowledge of the network model, and it cannot therefore compute the filter presented in (3). Let A be a consensus matrix, and observe that it can be written as $A_d + \varepsilon\Delta$, where $\|\Delta\|_\infty = 2$, $0 \leq \varepsilon \leq 1$, and A_d is block diagonal with a consensus matrix on each of the N diagonal blocks. For instance, let $A = [a_{kj}]$, and let $V_1, \dots, V_N$ be the subsets of agents associated with the blocks. Then the matrix $A_d = [\bar{a}_{kj}]$ can be defined as

(i) $\bar{a}_{kj} = a_{kj}$ if $k \neq j$, and $k, j \in V_i$, $i \in \{1, \dots, N\}$, and
(ii) $\bar{a}_{kk} = 1 + a_{kk} - \sum_{j \in V_i} a_{kj}$, and
(iii) $\bar{a}_{kj} = 0$ otherwise.

Moreover, $\Delta = 2(A - A_d)/\|(A - A_d)\|_\infty$, and $\varepsilon = \frac{1}{2}\|A - A_d\|_\infty$. Note that, if ε is "small", then the agents belonging to the same group are strongly interacting, while the agents belonging to different groups are weakly coupled, see Fig. 2(a) for an example. We assume the groups of strongly interacting agents to be given, and we leave the problem of determining such partitions as the subject of future research, for which the ideas presented in [12], [13] constitute a very relevant result.

We now focus on the h-th block. Let $K = v \cup l$ be the set of misbehaving agents, where $v = V_j \cap K$, and $l = K \setminus v$. Let $j \in V_h$, and consider the system (A_d, B_v, C_j). Recall from [4] that the misbehaving agents v are identifiable by agent j if the inputs u_v and u_i can be decoupled, for all $i \in V \setminus v$. Precisely, consider the system

$$\begin{aligned} \begin{bmatrix} x \\ w_v \end{bmatrix}^+ &= \begin{bmatrix} A_d & 0 \\ E_v C_j & F_v \end{bmatrix} \begin{bmatrix} x \\ w_v \end{bmatrix} + \begin{bmatrix} B_v & B_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_v \\ u_i \end{bmatrix}, \\ r_v &= \begin{bmatrix} H_v C_j & M_v \end{bmatrix} \begin{bmatrix} x \\ w_v \end{bmatrix}, \end{aligned} \tag{4}$$

and the system

$$\begin{aligned} \begin{bmatrix} x \\ w_i \end{bmatrix}^+ &= \begin{bmatrix} A_d & 0 \\ E_i C_j & F_i \end{bmatrix} \begin{bmatrix} x \\ w_i \end{bmatrix} + \begin{bmatrix} B_v & B_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_v \\ u_i \end{bmatrix}, \\ r_i &= \begin{bmatrix} H_i C_j & M_i \end{bmatrix} \begin{bmatrix} x \\ w_i \end{bmatrix}. \end{aligned} \tag{5}$$

Then the misbehaving agents v are identifiable by agent j if, for all $i \in V \setminus v$, we have $r_v \neq 0$ and $r_i = 0$ whenever $u_v \neq 0$. It should be noticed that, since A_d is block diagonal, the residual generators to identify the set v can be designed by only knowing the h-th block of A_d, and hence only a finite region of the original consensus network. Moreover, the misbehaving agents l do not affect the residuals r_i, $i \in V_h$, so that the agents v are identifiable by agent j if, for all $i \in V_h \setminus v$, we have $r_v \neq 0$ and $r_i = 0$ whenever $u_v \neq 0$. By applying the above residual generators to the consensus system $A_d + \varepsilon\Delta$ with misbehaving agents K we get

$$\begin{aligned} \begin{bmatrix} \hat{x} \\ \hat{w}_v \end{bmatrix}^+ &= \bar{A}_{\varepsilon,v} \begin{bmatrix} \hat{x} \\ \hat{w}_v \end{bmatrix} + \begin{bmatrix} B_v & B_l & B_i \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_v \\ u_l \\ u_i \end{bmatrix}, \\ \hat{r}_v &= \begin{bmatrix} H_v C_j & M_v \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{w}_v \end{bmatrix}, \end{aligned}$$

and

$$\begin{aligned} \begin{bmatrix} \hat{x} \\ \hat{w}_i \end{bmatrix}^+ &= \bar{A}_{\varepsilon,i} \begin{bmatrix} \hat{x} \\ \hat{w}_i \end{bmatrix} + \begin{bmatrix} B_v & B_l & B_i \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_v \\ u_l \\ u_i \end{bmatrix}, \\ \hat{r}_i &= \begin{bmatrix} H_i C_j & M_i \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{w}_i \end{bmatrix}, \end{aligned}$$

where

$$\bar{A}_{\varepsilon,v} = \begin{bmatrix} A_d + \varepsilon\Delta & 0 \\ E_v C_j & F_v \end{bmatrix}, \quad \bar{A}_{\varepsilon,i} = \begin{bmatrix} A_d + \varepsilon\Delta & 0 \\ E_i C_j & F_i \end{bmatrix}.$$

Because of the matrix Δ and the input u_l, the residual r_i is generally nonzero even if $u_i = 0$. However, the misbehaving agents v remain identifiable by j if for each $i \in V_j \setminus v$ it holds $\|\hat{r}_v\|_\infty > \|\hat{r}_i\|_\infty$ for all $u_v \neq 0$.

Theorem III.1 (Local identification) *Let V be the set of agents, let K be the set of misbehaving agents, and let $A_d + \varepsilon\Delta$ be a consensus matrix, where A_d is block diagonal, $\|\Delta\|_\infty = 2$, and $0 \leq \varepsilon \leq 1$. Let each block h of A_d*

[3] The design of the filter matrices depends upon the pair (i_1, i_2).

be a consensus matrix with agents $V_h \subseteq V$, and with connectivity $|K \cap V_h| + 1$. There exists $\alpha > 0$ and $u_{\max} \geq 0$, such that, if each input signal u_i, $i \in K$, takes value in $\mathcal{U} = \{u : \varepsilon\alpha u_{\max} \leq \|u\|_\infty \leq u_{\max}\}$, then each well-behaving agent $j \in V_h$ can identify in finite time the faulty agents $K \cap V_h$.[4]

Proof: We focus on the agent $j \in V_h$, and, without loss of generality, we assume that $u_K(0) \neq 0$, and that the residual generators have a finite impulse response. Let $d_j = \|V_h\|$, and note that d_j time steps are sufficient for each agent $j \in V_h$ to identify the misbehaving agents. Let u^t denote the input sequence up to time t. Let $v = K \cap V_h$, $l = K \setminus v$, and observe that

$$\hat{r}_v(d_j) = \left[\, H_v C_j \;\; M_v \,\right] \bar{A}_{\varepsilon,v}^{d_j} \bar{x}(0) + \hat{h}_v \star u_v^{d_j-1} + \hat{h}_l \star u_l^{d_j-1},$$

where $\hat{h}_v$ and $\hat{h}_l$ denote the impulse response from u_v and u_l respectively. We now determine an upper bound for each term of $\hat{r}_v(d_j)$. Let the misbehaving inputs take place in $\mathcal{U} = \{u : \varepsilon\alpha u_{\max} \leq \|u\|_\infty \leq u_{\max}\}$. By using the triangle inequality on the impulse responses of the residual generator, it can be shown that

$$\|\hat{h}_l \star u_l^{d_j-1}\|_\infty \leq \|h_l \star u_l^{d_j-1}\|_\infty + \varepsilon c_1 u_{\max} = \varepsilon c_1 u_{\max},$$

where h_l denotes the impulse response form u_l to r_v of the system (4), and c_1 is a finite positive constant independent of ε. Moreover, it can be shown that there exist two positive constant c_2 and c_3 such that

$$\|\left[\, H_v C_j \;\; M_v \,\right] \bar{A}_{\varepsilon,v}^{d_j} \bar{x}(0)\|_\infty \leq \varepsilon c_2 u_{\max},$$

and

$$\min_{u_v \in \mathcal{U}} \|\hat{h}_v \star u_v^{d_j-1}\|_\infty \geq \min_{u_v \in \mathcal{U}} \|h_v \star u_v^{d_j-1}\|_\infty - \varepsilon c_3 u_{\max}.$$

Analogously, for the residual generator associated with the well-behaving agent i, we have

$$\hat{r}_i(d_j) = \left[\, H_i C_j \;\; M_i \,\right] \bar{A}_{\varepsilon,i}^{d_j} \bar{x}(0) + \hat{h}_v \star u_v^{d_j-1} + \hat{h}_l \star u_l^{d_j-1},$$

and hence

$$\hat{r}_i(d_j) \leq \varepsilon(c_4^{(i)} + c_5^{(i)} + c_6^{(i)}) u_{\max}.$$

Let $\bar{c} = c_1 + c_2 + c_3 + \max_{i \in V_h \setminus v}(c_4^{(i)} + c_5^{(i)} + c_6^{(i)})$, and let β be such that $\min_{u_v \in \mathcal{U}} \|h_v \star u_v^{d_j-1}\|_\infty > \beta u_{\min}$. Then a correct identification of the misbehaving agents v takes place if $\beta u_{\min} > \varepsilon \bar{c} u_{\max}$. ∎

Notice that the constant α in Theorem III.1 can be computed by bounding the infinity norm of the impulse response of the residual generators. An example follows.

IV. An example of local identification

We show in this section the advantages of the clustered setup described in Section III. Consider the consensus network in Fig. 1, where $A = A_d + \varepsilon\Delta$, $\varepsilon \leq 1$, and

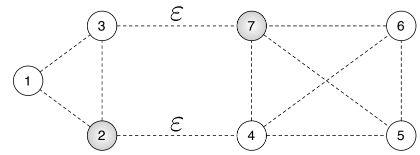


Fig. 1. A consensus network with two weakly interconnected subnetworks.

$$A_d = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix},$$

$$\Delta = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

Let $K = \{2, 7\}$ be the set of misbehaving agents, and let $\|x(0)\|_\infty \leq 1$. Consider the agent 1, and let (F_2, E_2, M_2, H_2) and (F_3, E_3, M_3, H_3) be the residual generators as in (4) and (5) respectively, where

$$F_2 = \begin{bmatrix} -1/3 & -1/3 \\ 1/3 & 1/3 \end{bmatrix}, \quad E_2 = \begin{bmatrix} -2/3 & 0 & -1/3 \\ 2/3 & 0 & 1/3 \end{bmatrix},$$
$$M_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

and

$$F_3 = \begin{bmatrix} -1/3 & 1/3 \\ -1/3 & 1/3 \end{bmatrix}, \quad E_3 = \begin{bmatrix} -2/3 & -1/3 & 0 \\ -2/3 & -1/3 & 0 \end{bmatrix},$$
$$M_3 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_3 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Let $\hat{h}_2^3$ (resp. $\hat{h}_7^3$) be the impulse response from the input u_2 (resp. u_7) to $\hat{r}_3$, and let u_2^1 (resp. u_7^1) denote the input signal u_2 (resp. u_7) up to time 1. Because the filters (F_2, E_2, M_2, H_2) and (F_3, E_3, M_3, H_3) converge in two steps, the misbehaving agent can be identified after 2 time steps. The residual associated with the agent 3 is

$$\hat{r}_3(1) = \left[\, H_3 C_1 \;\; M_3 \,\right] \begin{bmatrix} A_d + \varepsilon\Delta & 0 \\ E_3 C_1 & F_3 \end{bmatrix}^2 \begin{bmatrix} x(0) \\ 0 \end{bmatrix} + \hat{h}_2^3 \star u_2^1 + \hat{h}_7^3 \star u_7^1$$

or, equivalently,

$$\hat{r}_3(1) = \varepsilon \left[\, H_3 C_1 \;\; M_3 \,\right] \begin{bmatrix} A_d\Delta + \Delta A_d + \varepsilon\Delta^2 & \Delta B_2 & \Delta B_7 \\ E_3 C_1 \Delta & 0 & 0 \end{bmatrix} \begin{bmatrix} x(0) \\ u_2(0) \\ u_7(0) \end{bmatrix}.$$

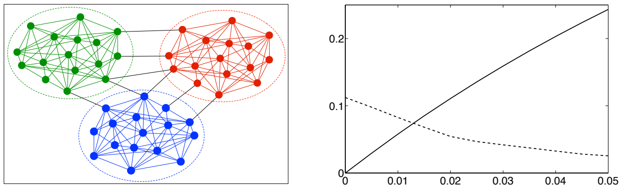
Analogously, we have

$$\hat{r}_2(1) = \varepsilon \left[\, H_2 C_1 \;\; M_2 \,\right] \begin{bmatrix} A_d\Delta + \Delta A_d + \varepsilon\Delta^2 & \Delta B_2 & \Delta B_7 \\ E_2 C_1 \Delta & 0 & 0 \end{bmatrix} \begin{bmatrix} x(0) \\ u_2(0) \\ u_7(0) \end{bmatrix}$$
$$+ \left[\, H_2 C_1 \;\; M_2 \,\right] \begin{bmatrix} A_d B_2 & B_2 \\ E_2 C_1 B_2 & 0 \end{bmatrix} \begin{bmatrix} u_2(0) \\ u_2(1) \end{bmatrix}.$$

The agent 1 is able to identify the misbehaving agent 2 if it holds $\|\hat{r}_2(1)\|_\infty > \|\hat{r}_3(1)\|_\infty$ independently of u_2^1 and u_7^1. Let the inputs u_2 and u_7 take value in $\mathcal{U} = \{u : u_{\min} \leq \|u\|_\infty \leq u_{\max}\}$. It can be verified that $\|\hat{r}_2(1)\|_\infty > \|\hat{r}_3(1)\|_\infty$ if

$$\min_{u_2 \in \mathcal{U}} \left\| \left[\, H_2 C_1 \;\; M_2 \,\right] \begin{bmatrix} A_d B_2 & B_2 \\ E_2 C_1 B_2 & 0 \end{bmatrix} \begin{bmatrix} u_2(0) \\ u_2(1) \end{bmatrix} \right\|_\infty > 11 \varepsilon u_{\max},$$

[4] An identification procedure based on this method is in [14].



(a) (b)

Fig. 2. In Fig. 2(a) a consensus network partitioned into 3 areas. Each agent identifies the neighboring misbehaving agents by only knowing the topology of the subnetwork it belongs to. In Fig. 2(b) the smallest magnitude of the residual associated with a misbehaving agent (dashed line) and the largest magnitude of the residual associated with a well-behaving agent (solid line) are plotted as a function of the weak connections ε.

and, after some computation, if $u_{\min} > 47\varepsilon u_{\max}$, in which case we conclude that the agent 1 correctly identifies the misbehaving agent 2 when $K = \{2, 7\}$. The analysis of the other pairr of misbehaving agents is done analogously, and it is not reported here.

As a final remark, note that the larger the consensus network, the more convenient the proposed approximation procedure becomes. For instance, consider the network presented in [15], and here reported in Fig. 2(a). Such a clustered interconnection structure, in which the edges connecting different clusters have a small weight, may be preferable in many applications because much simpler and efficient protocols can be implemented within each cluster. Assume that there is a misbehaving agent in each cluster, and consider the residuals computed after 5 steps of the consensus algorithm. Let ε be the weight of the edges connecting different clusters, and let the misbehaving inputs take value in $\mathcal{U} = \{u : 0.1 \leq u \leq 3\}$. Fig. 2(b) shows, as a function of ε, the smallest magnitude of the residual associated with a misbehaving agent (dashed line) versus the largest magnitude of the residual associated with a well-behaving agent (solid line). If ε is sufficiently small, then our local identification method allows each well-behaving agent to promptly detect and identify the misbehaving agents belonging to the same group, and hence to restore the functionality of the network.

V. Hierarchical estimation and detection

The previous section shows how to detect a misbehaving agent under limited knowledge of the overall system. The proposed algorithm relies on the key assumption that the magnitude of the misbehaving signal is within an interval whose size strictly depends on the parameters of the system. In this section we present a complementary and alternative method to remove this constraint while maintaining the assumption of local knowledge of the network.

We introduce a hierarchical structure that reduces the decentralization of the network by allowing for the presence of a subset of nodes with better communication and computation capabilities. We refer to these nodes as the *leaders* of the network. In this Section, before considering the detection problem, we exploit the presence of this hierarchical structure for solving the state estimation problem in a linear system with unknown inputs. In Subsection V-A we propose an algorithm that allows each leader to recover the state $x(0)$ in a finite number of steps. In Subsection V-B we modify the above algorithm to include the detection of misbehaving agents. While illustrating our algorithms we characterize also the local knowledge of the network required by each leader to accomplish the state estimation and the detection goals.

A. Hierarchical unknown input state estimation

Consider the linear network[5] $x(t+1) = Ax(t) + Bu(t)$ and let $G = (V, E)$ be the graph associated with the matrix A. Let $V^{(\ell)} = \{\ell_1, \dots, \ell_m\} \subseteq V$ denote the subset of the leaders. We assume the presence of a directed graph $G^{(\ell)} = (V^{(\ell)}, E^{(\ell)})$, where $E^{(\ell)} \subseteq V^{(\ell)} \times V^{(\ell)}$ describes the feasible communications among the leaders. We assume that $G^{(\ell)}$ is strongly connected, and we refer to it as to the *leader graph*. Let $N_i^{(\ell)}$ denote the set of the neighbors of the leader ℓ_i in $G^{(\ell)}$. As in Section II, the information of the state $x(t)$ directly available to the leader ℓ_i is given by $y_i(t) = C_{\ell_i} x(t)$, where C_{ℓ_i} is defined according to the neighbors set N_{ℓ_i} in G. The composite information available to the set of leaders can be conveniently described by the output matrix $C^{(\ell)} = [C_{\ell_1}^T \ \cdots \ C_{\ell_m}^T]^T$. We now show how our hierarchical setup can be conveniently used to solve the unknown input state estimation problem, in which the input matrix B is known by the leaders, while the input signal $u(t)$ is unknown and unmeasurable. For $s \in \mathbb{Z}_{>0}$, let

$$O_i^s = \begin{bmatrix} C_{\ell_i} \\ C_{\ell_i} A \\ C_{\ell_i} A^2 \\ \vdots \\ C_{\ell_i} A^{s-1} \end{bmatrix}, \quad Y_i^s = \begin{bmatrix} y_i(0) \\ y_i(1) \\ y_i(2) \\ \vdots \\ y_i(s-1) \end{bmatrix},$$

and

$$F_i^s = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ C_{\ell_i} B & 0 & \ddots & \ddots & 0 \\ C_{\ell_i} AB & C_{\ell_i} B & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ C_{\ell_i} A^{s-2} B & C_{\ell_i} A^{n_i - 1} B & \cdots & C_{\ell_i} B & 0 \end{bmatrix}.$$

Finally, let

$$O^s = \begin{bmatrix} O_1^s \\ O_2^s \\ \vdots \\ O_m^s \end{bmatrix}, \quad Y^s = \begin{bmatrix} Y_1^s \\ Y_2^s \\ \vdots \\ Y_m^s \end{bmatrix}, \quad F^s = \begin{bmatrix} F_1^s \\ F_2^s \\ \vdots \\ F_m^s \end{bmatrix}.$$

Note that

$$\begin{bmatrix} Y_1^s \\ Y_2^s \\ \vdots \\ Y_m^s \end{bmatrix} = \begin{bmatrix} O_1^s \\ O_2^s \\ \vdots \\ O_m^s \end{bmatrix} x(0) + \begin{bmatrix} F_1^s \\ F_2^s \\ \vdots \\ F_m^s \end{bmatrix} U^s, \tag{6}$$

where U^s contains the input sequence from time 0 up to time $s-1$. From [11] we know that a system is finite-time

[5]The results presented in this section hold for general linear networks, i.e., they are not restricted to consensus dynamics.

unknown input observable (UIO), i.e., the initial state $x(0)$ can be recovered without knowing the input signal U^d, if and only if there exists an integer $d < |V|$ such that

$$\mathrm{Ker}(O^d) = 0 \qquad \text{and} \qquad \mathrm{Im}(O^d) \cap \mathrm{Im}(F^d) = 0. \qquad (7)$$

In particular conditions (7) imply that $x(0)$ can be computed as the solution of the system $Y^d = \begin{bmatrix} O^d & F^d \end{bmatrix} \begin{bmatrix} x^T & U^T \end{bmatrix}^T$. To see this, let I_n denote the n-dimensional identity matrix, and observe that

$$\mathrm{Ker} \begin{bmatrix} O^d & F^d \end{bmatrix} \perp \mathrm{Im} \left(\begin{bmatrix} I_n \\ 0 \end{bmatrix} \right) = 0, \qquad (8)$$

where, given two subspaces $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \perp \mathcal{B}$ denotes the orthogonal projection of $\mathcal{A}$ onto $\mathcal{B}$. Let

$$\begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix} := \begin{bmatrix} O^d & F^d \end{bmatrix}^\dagger Y^d, \qquad (9)$$

where $\dagger$ denotes the pseudo-inverse operation, then it follows from (8) that $\hat{x} = x(0)$. Consider the basic algebraic equality

$$\mathrm{Ker}([O^d \;\; F^d]) = \cap_{i=1}^m \mathrm{Ker}([O_i^d \;\; F_i^d]), \qquad (10)$$

which leads to the useful geometric interpretation of (9) that is next described. For $i \in \{1, \dots, m\}$, let $\mathcal{S}_i = \left[\begin{smallmatrix} \hat{x}_i \\ \hat{u}_i \end{smallmatrix}\right] + \mathcal{V}_i$, where $\left[\begin{smallmatrix} \hat{x}_i \\ \hat{u}_i \end{smallmatrix}\right] = [O_i^d \;\; F_i^d]^\dagger Y_i^d$ and $\mathcal{V}_i = \mathrm{Ker}([O_i^d \;\; F_i^d])$. Then x coincides with the projection onto the subspace $\mathrm{Im}\left(\left[\begin{smallmatrix} I_n \\ 0 \end{smallmatrix}\right]\right)$ of the intersection of the affine subspaces $\{\mathcal{S}_1, \dots, \mathcal{S}_m\}$. It follows indeed from (8) and (10) that $\cap_{j=1}^m \mathcal{S}_i \perp \left[\begin{smallmatrix} I_n \\ 0 \end{smallmatrix}\right]$ results in a vector whose first n components coincide with $x(0)$. Based on the above discussion, in Algorithm 1 we propose a distributed procedure that allows each leader to estimate x and that only requires a local knowledge of the network. The *Decentralized state estimation* algorithm is briefly described as follows. For $i \in \{1, \dots, m\}$, the i-th leader ℓ_i keeps in memory an estimate $z_i = \left[\begin{smallmatrix} \hat{x}_i \\ \hat{u}_i \end{smallmatrix}\right]$ and an uncertainty subspace $\mathcal{V}_i$. These variables are initialized as $z_i(0) = ([O_i^d \;\; F_i^d])^\dagger Y_i^d$ and $\mathcal{V}_i = \mathrm{Ker}([O_i^d \;\; F_i^d])$. At the t-th iteration, each leader ℓ_i performs the following three actions in order:

(i) receive $\mathcal{S}_j(t-1) = z_j(t-1) + \mathcal{V}_j$ from all $\ell_j \in N_i^{(\ell)}$,
(ii) set $z_i(t)$ to the orthogonal projection of $z_i(t-1)$ onto the intersection of the affine subspaces $\{\mathcal{S}_j(t-1) : \ell_j \in N_i^{(\ell)} \cup \{\ell_i\}\}$, and $\mathcal{V}_i(t)$ to the intersection of the uncertainty subspaces $\{\mathcal{V}_j(t-1) : \ell_j \in N_i^{(\ell)} \cup \{\ell_i\}\}$,
(iii) transmit $z_i(t)$ and $\mathcal{V}_i(t)$ to all $\ell_j \in N_i^{(\ell)}$.

Let $\mathrm{diam}(G^{(\ell)})$ denote the diameter of $G^{(\ell)}$. The convergence of Algorithm 1 is next stated.

Theorem V.1 (Decentralized UIO) *Let $(A, B, C^{(\ell)})$ be the unknown input linear system associated with the graph G and the leader graph $G^{(\ell)}$. Assume that*

(i) *$G^{(\ell)}$ is strongly connected, and*
(ii) *there exists an integer d such that $\mathrm{Ker}(O^d) = 0$ and $\mathrm{Im}(O^d) \cap \mathrm{Im}(F^d) = 0$, and*
(iii) *each leader i knows the matrices O_i^d and F_i^d.*

The Decentralized state estimation algorithm provides each leader with the system initial state in $\mathrm{diam}(G^{(\ell)})$ steps.

Algorithm 1: *Decentralized state estimation (leader i)*

Input : O_i^d, Y_i^d, F_i^d;
Require : $\mathrm{Ker}(O^d) = 0, \mathrm{Im}(O^d) \cap \mathrm{Im}(F^d) = 0$;
set $\begin{bmatrix} \hat{x}_i \\ \hat{u}_i \end{bmatrix} = ([O_i^d \;\; F_i^d])^\dagger Y_i^d$, $\mathcal{V}_i = \mathrm{Ker}([O_i^d \;\; F_i^d])$;
transmit $\mathcal{S}_i = \begin{bmatrix} \hat{x}_i \\ \hat{u}_i \end{bmatrix} + \mathcal{V}_i$;
while $\mathcal{V}_i \perp \mathrm{Im}\left(\begin{bmatrix} I_n \\ 0 \end{bmatrix}\right) \neq 0$ **do**
  for $\ell_j \in N_i^{(\ell)}$ **do**
    receive $\begin{bmatrix} \hat{x}_j \\ \hat{u}_j \end{bmatrix}$ and $\mathcal{V}_j$;
    set $\begin{bmatrix} \hat{x}_i \\ \hat{u}_i \end{bmatrix} = \begin{bmatrix} \hat{x}_i \\ \hat{u}_i \end{bmatrix} \perp (\mathcal{S}_i \cap \mathcal{S}_j)$, $\mathcal{V}_i = \mathcal{V}_i \cap \mathcal{V}_j$;
  transmit $\mathcal{S}_i = \begin{bmatrix} \hat{x}_i \\ \hat{u}_i \end{bmatrix} + \mathcal{V}_i$;
return $\hat{x}_i$;

Proof: According to the initialization of Algorithm 1, for $i \in \{1, \dots, m\}$, we have that

$$\mathcal{S}_i(0) = \begin{bmatrix} \hat{x}_i(0) \\ \hat{u}_i(0) \end{bmatrix} + \mathcal{V}_i(0),$$

where $\begin{bmatrix} \hat{x}_i^T(0) & \hat{u}_i^T(0) \end{bmatrix}^T = \begin{bmatrix} O_i^d & F_i^d \end{bmatrix}^\dagger Y_i$ and $\mathcal{V}_i(0) = \mathrm{Ker}(\begin{bmatrix} O_i^d & F_i^d \end{bmatrix})$. For $\ell_i, \ell_j \in V^{(\ell)}$ let d_{ℓ_j,ℓ_i} denote the distance of the shortest path in $G^{(\ell)}$ connecting ℓ_j to ℓ_i. Then, for $i \in \{1, \dots, m\}$, let $N_{i,k}^{(\ell)} = \left\{\ell_j \in V^\ell | d_{\ell_j,\ell_i} \leq k\right\}$. We show by induction that, for $t \in \mathbb{Z}_{\geq 0}$ and $i \in \{1, \dots, m\}$, it holds

$$\mathcal{S}_i(t) = \bigcap_{j \in N_{i,t}^{(\ell)}} \mathcal{S}_j(0). \qquad (11)$$

Notice that, for $t = 0$, equation (11) trivially follows from the fact that $N_{i,0}^{(\ell)} = \{\ell_i\}$. Let $t \in \mathbb{Z}_{>0}$, and assume that equation (11) holds true up to $t-1$. Observe that

$$\begin{aligned} \mathcal{S}_i(t) &= S_i(t-1) \cap \left(\bigcap_{j \in N_i^{(\ell)}} \mathcal{S}_j(t-1) \right) \\ &= \left(\bigcap_{h \in N_{i,t-1}^{(\ell)}} \mathcal{S}_h(0) \right) \cap \left(\bigcap_{j \in N_i^{(\ell)}} \bigcap_{h \in N_{j,t-1}^{(\ell)}} \mathcal{S}_h(0) \right) \end{aligned}$$

where the last equality follows from the inductive hypothesis. Since $N_{i,t}^{(\ell)} = \cup_{j \in N_i^\ell} N_{j,t-1}^{(\ell)}$, equation (11) follows from the above equality. Because $G^{(\ell)}$ is strongly connected, we have that $\mathcal{S}_i(\mathrm{diam}(G^{(\ell)})) = \cap_{j \in \{1,\dots,m\}} \mathcal{S}_j(0)$ for all $i \in \{1, \dots, m\}$. Because of Assumption (ii), the first n components of the vector $\cap_{j=1}^m \mathcal{S}_i(0) \perp \left[\begin{smallmatrix} I_n \\ 0 \end{smallmatrix}\right]$ coincide with $x(0)$. We conclude that $\hat{x}_i(\mathrm{diam}(G^{(\ell)})) = x(0)$. ■

Remark 1 *The computation of the matrices O_i^d, $i \in V^{(\ell)}$, does not require the knowledge of the entire network model. Given a graph, let a path be a sequence of vertices, such that any two consecutive vertices in the sequence are connected through an edge. Let the length of a path equal the number of its edges. Let A be the network matrix, and observe that*

Algorithm 2: *Decentralized detection (leader i)*

Input : O_i^d, Y_i^d;
Require : $\mathrm{Im}(O^d) \cap \mathrm{Im}(F^d) = 0$;

set $\mathcal{S}_i = (O_i^d)^\dagger Y_i^d + \mathrm{Ker}(O_i^d)$;
transmit $\mathcal{S}_i$;
for *diam*$(G^{(\ell)})$ *iterations* **do**
  for $j \in N_i^{(\ell)}$ **do**
    receive $\mathcal{S}_j$;
    set $\mathcal{S}_i = \mathcal{S}_i \cap \mathcal{S}_j$;
  transmit $\mathcal{S}_i$;
if $\mathcal{S}_i = \emptyset$ **then return** *1*
else return *0*

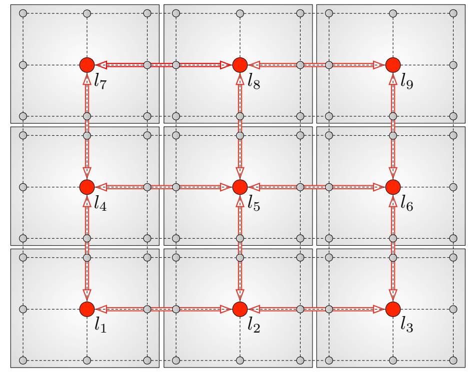


Fig. 3. A grid partitioned into 9 cblocks. Each block is identical and it contains 9 vertices. The central vertex of a block corresponds to the leader.

the (i, j)-th entry of A^k, with $k \in \mathbb{Z}$, is nonzero if and only if there exists a path of length k connecting the agent j to i. Let $N_{\ell_i}^d \subseteq V$ denote the set of neighbors within distance d from the leader ℓ_i, i.e., the set of agents connected to ℓ_i through a path of length at most d. It can be shown that the matrix O_i^d can be computed by only knowing the sub-matrix of A with rows and columns in $N_{\ell_i}^d$.

It should be noticed that Algorithm 1 may converge with less than $\mathrm{diam}(G^{(\ell)})$ iterations. In particular, by increasing the number of measurements d, we have that the number of iterations for the convergence of the proposed Algorithms decreases. To see this, note that if the pair (A, C_{ℓ_1}) is observable, then, with a sufficiently large number of observations d, the leader ℓ_1 is able to reconstruct the state without communicating with the other leaders. Note however that a larger d requires the leaders to know a larger subnetworks, and, as it is shown in Section VI, it introduces numerical difficulties in the execution of our algorithms.

B. Hierarchical detection

We consider now the problem of detecting the presence of the misbehaving agents. Because the misbehaving set is a priory unknown, the input matrix B and hence the matrix F^d are to be considered unknown as well.

Let $U^d = [u(0)^T \; \dots \; u(d-1)^T]^T$, and assume that each leader ℓ_i has collected the observations $y_{\ell_i}(0), \dots, y_{\ell_i}(d-1)$. In Algorithm 2 we propose a procedure that allows the leaders to detect if $F^d U^d \neq 0$ without using the matrix F^d.

Theorem V.2 (Decentralized detection) *Let $(A, B, C^{(\ell)})$ be the unknown input linear system associated with the graph G and the leader graph $G^{(\ell)}$. Let u be the misbehaving input, and let $U^d = [u(0)^T \; \cdots \; u(d-1)^T]^T$, with $d \in \mathbb{Z}$. Assume that*

(i) *$G^{(\ell)}$ is strongly connected, and*
(ii) *$\mathrm{Im}(O^d) \cap \mathrm{Im}(F^d) = 0$, and*
(iii) *each leader i knows the matrices O_i^d.*

Then the Decentralized detection algorithm allows each leader to detect if $F^d U^d \neq 0$ in at most diam$(G^{(\ell)})$ steps.

Proof: Because $\mathrm{Im}(O^d) \cap \mathrm{Im}(F^d) = 0$, we have that the system $Y^d = O^d x$ is inconsistent if $F^d U^d \neq 0$, so that $\bigcap_{i=1}^m (\hat{x}_i + \mathcal{V}_i) = \emptyset$, where $\hat{x}_i = (O_i^d)^\dagger Y_i^d$ and $\mathcal{V}_i = \mathrm{Ker}(O_i^d)$. Because $G^{(\ell)}$ is strongly connected, after at most $\mathrm{diam}(G^{(\ell)})$ iterations each leader detects if $F^d U^d \neq 0$. ■

The following aspects should be noticed. First, the condition $\mathrm{Ker}(O^d) = 0$ is not required by the *Decentralized detection* because only the presence of an unknown input has to be assessed. Second, in order to detect a misbehaving input that becomes nonzero at an arbitrary instant of time, the detection algorithm needs to be executed iteratively. Precisely, at each time $t \geq d-1$, the consistency of the system $Y_t^d = O^d x(t-d+1)$ is checked with the detection algorithm, where $Y_t^d = [y(t-d+1)^T \; \cdots \; y(t)^T]^T$, $y(t) = C^{(\ell)} x(t)$, and $U_t^d = [u(t-d+1)^T \; \cdots \; u(t)^T]^T$. Third and finally, for the detection to be possible, there must exist $d \in \mathbb{Z}$ and $t \geq d-1$ such that $F^d U_t^d \neq 0$. Such condition coincides with the left-invertibility of the linear network, which has to be assumed by any detection method [11].

VI. An example of hierarchical estimation and detection

We show in this section the advantages of the hierarchical structure presented in Section V. Let the network G be a two dimensional lattice with $(ab)^2$ agents, and let the network be partitioned into b^2 identical blocks containing a^2 vertices each. An example with $b = 3$ and $a = 3$ is in Fig. 3. Let A describe the linear algorithm running on G, and assume that the entries of A have been chosen independently from each other and uniformly in the interval $(0, 1)$, and then normalized so that the row sums of A are all 1. Let V_i, with $i \in \{1, \dots, b^2\}$, denote the set of agents belonging to the i-th block, and let the central vertex $l_i \in V_i$ represent the i-th leader. We assume that the leaders l_i and l_j are connected through an undirected edge if there exists $h_1 \in V_i$ and $h_2 \in V_j$ that are connected in G. We focus on leader l_1 and we analyze the performance of our procedures as a function of the parameters a and b.

A. State estimation

We compare here the performance of Algorithm 1 with the method proposed in [10], where the state is recovered by relying on the observability property of the pair (A, C_{l_1}). We show that, although theoretically correct, the latter method suffers from numerical instability when the dimension of A grows. Let $a = 3$ and $b = 3$, and compute the condition number[6] of the observability matrix of the pair $(A, C_k^{(\ell)})$, where $C_k^{(\ell)}$ is the composite output matrix associated with the leaders set $V_k^{(\ell)} = \{l_1, \dots, l_k\}$, $k = 1, \dots, 9$. As we see from Table I, the condition number rapidly decreases by increasing the number of leaders. Precisely, in the case of $V_1^{(\ell)}$ the condition number of the observability matrix is $\sim 10^{14}$ so that the problem of estimating the state only relying on the measurements of l_1 is very ill-conditioned. When the

TABLE I

Leader	Condition number	Size (a)	Size (b)	Measurement (d)
$V_1^{(\ell)}$	$\sim 10^{14}$	3	1	2
$V_2^{(\ell)}$	$\sim 10^{7}$	3	3	3
$V_3^{(\ell)}$	$\sim 10^{5}$	3	5	3
$V_4^{(\ell)}$	$\sim 10^{4}$	3	7	3
$V_5^{(\ell)}$	$\sim 10^{4}$	5	1	6
$V_6^{(\ell)}$	$\sim 10^{3}$	5	3	7
$V_7^{(\ell)}$	$\sim 10^{3}$	5	5	7
$V_8^{(\ell)}$	$\sim 10^{2}$	5	7	7
$V_9^{(\ell)}$	$\sim 10^{2}$	5	9	7

leaders set is $V_9^{(\ell)}$, the condition number becomes $\sim 10^2$, so that each leader can estimate the correct state reliably and with limited model information by means of Algorithm 1.

We now investigate numerically a scalability property of Algorithm V.1. Let $|V^{(\ell)}| = b^2$. Table I contains the minimum number of measurements d such that $\mathrm{Ker}(O^d) = 0$, or, in other words, such that each leader is assured to estimate the network state. Observe that, when a is fixed and b grows, the number d remains constant. It follows that the knowledge about the network model that a leader needs to possess does not depend upon the cardinality of the network.

B. Detection

We analyze here the performance of Algorithm 2. Let the agent $i \in V \setminus V^{(\ell)}$ be misbehaving, and let the input sequence $\{u_i(t),\ t \in \mathbb{Z}_{\geq 0}\}$ be an i.i.d. sequence taking value in the interval $(0, 1)$. For each $a \in \{3, 5\}$ and each $b \in \{2, \dots, 12\}$ we consider 20 randomly chosen consensus networks, we locate b^2 leaders (cfr. Fig. 3), and we choose the misbehaving agent i. The first instant of time in which a leader detects the presence of i by means of Algorithm 2 is reported in Fig. 4. Note that the detection time remains constant when the dimension of the network grows beyond a threshold. It follows that Algorithm 2 converges before $\mathrm{diam}(G^{(\ell)})$ iterations, exhibiting therefore desirable scalability properties.

VII. Conclusion

The problem of estimating the state of a linear network in the presence of an unknown input, as well as the problem of detecting misbehaving parts in a linear network have been considered. Whereas classical approaches require a complete knowledge of network model, our methods only assumes

[6]The condition number equals the ratio of the largest singular value to the smallest. Large condition numbers indicate a nearly singular matrix.

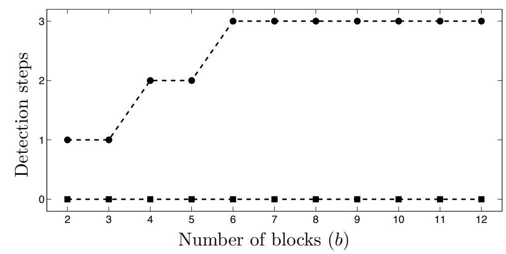


Fig. 4. The figure shows the number of iterations required for the detection of a misbehaving agent by means of Algorithm 2. Both the cases of $a = 3$ (squares) and $a = 5$ (circles) are plotted as a function of b.

partial knowledge of the system structure by the observer agents. For the unknown input state estimation problem, we assume the presence of some interconnected leader agents, and we present a novel iterative algorithm which produces an exact state estimation. For the detection of misbehaving agents, we present two different approaches. The first method exploits the presence of weakly interconnected subnetworks, and it affords detection and identification when the misbehaving inputs overcome a certain threshold. The second method relies on our hierarchical leader structure, and it achieves exact detection independent of the misbehaving inputs.

References

[1] R. Patton, P. Frank, and R. Clark, *Fault Diagnosis in Dynamic Systems: Theory and Applications*. Prentice Hall, 1989.

[2] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.

[3] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[4] F. Pasqualetti, A. Bicchi, and F. Bullo, "On the security of linear consensus networks," in *IEEE Conf. on Decision and Control*, Shanghai, China, Dec. 2009, pp. 4894–4901.

[5] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[6] D. Dolev, "The Byzantine generals strike again," *Journal of Algorithms*, vol. 3, pp. 14–30, 1982.

[7] F. Pasqualetti, A. Bicchi, and F. Bullo, "Distributed intrusion detection for secure consensus computations," in *IEEE Conf. on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 5594–5599.

[8] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents - Part I: Attacking the network," in *American Control Conference*, Seattle, WA, June 2008, pp. 1350–1355.

[9] ——, "Distributed function calculation via linear iterations in the presence of malicious - Part II: Overcoming malicious behavior," in *American Control Conference*, Seattle, WA, June 2008, pp. 1356–1361.

[10] ——, "Distributed function calculation and consensus using linear iterative strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 650–660, 2008.

[11] H. L. Trentelman, A. Stoorvogel, and M. Hautus, *Control Theory for Linear Systems*. Springer, 2001.

[12] R. G. Phillips and P. Kokotović, "A singular perturbation approach to modeling and control of Markov chains," *IEEE Transactions on Automatic Control*, vol. 26, no. 5, pp. 1087–1094, 1981.

[13] J. H. Chow, J. Cullum, and R. A. Willoughby, "A sparsity-based technique for identifying slow-coherent areas in large power systems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 3, pp. 463–473, 1984.

[14] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, Feb. 2010, submitted.

[15] E. Bıyık and M. Arcak, "Area aggregation and time-scale modeling for sparse nonlinear networks," *Systems & Control Letters*, vol. 57, no. 2, pp. 142–149, 2007.