On the Security of Linear Consensus Networks

Fabio Pasqualetti Antonio Bicchi Francesco Bullo

Abstract— **This work considers the problem of reaching consensus in linear networks with misbehaving agents. A solution to this problem is relevant for several tasks in multi-agent systems including motion coordination, clock synchronization, and cooperative estimation. By modelling the misbehaving nodes as unknown and unmeasurable inputs affecting the network, we recast the problem into a system theoretic framework. Only relying on their direct measurements, the agents detect and identify uncooperative behaviors using fault detection and isolation techniques. We consider both the cases of Byzantine as well as non-colluding faults, and we express the solvability conditions of the two cases in terms of the observability properties of a linear system associated with the network, and from a graph theoretic perspective. It is shown that generically any node can correctly detect and identify the misbehaving agents, provided that the connectivity of the network is sufficiently high. Precisely, for a linear consensus network to be generically resilient to k concurrent faults, the connectivity of the communication graph needs to be $2k+1$, if Byzantine agents are allowed, and $k+1$, if non-colluding agents are considered.**

I. Introduction

Distributed systems and networks have received much attention in the last years because of their flexibility and computation performance. One of the most frequent task to be accomplished by autonomous agents is to agree upon some parameters. Agreement variables represent quantities of interest such as the work load in a network of parallel computers, the clock speed for wireless sensor networks, the velocity, the rendezvous point, or the formation pattern for a team of autonomous vehicles; e.g., see [1], [2], [3].

Several algorithms achieving consensus have been proposed and studied in the computer science community [4]. In this work, we consider linear iterations, where, at each time instant, each node updates its state as a weighted combination of its own value and those received from its neighbors [1], [2]. The choice of algorithm weights is a parameter that influences the convergence speed toward the steady state value [5].

Because of the lack of a centralized entity which may monitor the activity of the nodes of the network, distributed systems are prone to attacks and components failure, and it is of increasing importance to guarantee trustworthy computation even in the presence of misbehaving parts. The misbehaving agents are here classified, depending on their abilities, as Byzantine, or malicious, and as non-colluding, or faulty. Malicious agents have complete knowledge of the network, and posses unlimited sensing, communication, and computation capabilities. Also, they collude in order to cause the biggest damage to the network. On the other hand, faulty agents do not cooperate maliciously, and their uncooperative behavior is often due to an hardware failure. When malicious agents are present, the worst case scenario for the network has to be considered, whereas, in the presence of faulty agents, atypical agents behaviors, i.e., those occurring in practice with zero probability, are not taken into account.

Reaching unanimity in an unreliable system is an important problem well known by computer scientists interested in distributed computing. A first characterization of the resilience of distributed systems to Byzantine attacks appears in [6], where the authors consider the task of agreeing upon a binary message sent by a "general," when the communication graph is complete. In [7] the resilience of a partially connected[1] network seeking consensus is analyzed, and it is shown that the well-behaving agents of a network can always agree upon a parameter if and only if the number of malicious agents

(i) is less than one-half of the network connectivity, and
(ii) it is less than one-third of the number of processors.

This result has to be regarded as a fundamental limitation of the ability of a distributed consensus system to sustain arbitrary malfunctioning: the presence of misbehaving Byzantine processors can be tolerated only if their number satisfies the above threshold, independently of whatever consensus protocol is adopted.

In this work, we consider linear consensus algorithms in which every agent, including the misbehaving ones, are assumed to send the same information to all their neighbors. This assumption appears to be realistic for most control scenarios. In a sensing network for instance, the data used in the consensus protocol consist of the measurements taken directly by the agents, and it is assumed that the measurements regarding the same quantity coincide. Also, in a broadcast network, the information is transmitted using broadcast messages, so that the content of a message is the same for all the receiving nodes. The problem of characterizing the resilience properties of linear consensus strategies has been partially addressed in recent works [8], [9], [10], where, for the malicious case, it is shown that, despite the limited abilities

This material is based upon work supported in part by the Institute for Collaborative Biotechnology and the ARO award DAAD19-03-D-0004, and in part by the Contract IST 224428 (2008) (STREP) "CHAT - Control of Heterogeneous Automation Systems: Technologies for scalability, reconfigurability and security," and the CONET, the Cooperating Objects Network of Excellence, funded by the European Commission under FP7 with contract number FP7-2007-2-224053. The authors thank Dr. Natasha Neogi for insighful conversations.

Fabio Pasqualetti is with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara fabiopas@engineering.ucsb.edu .

Antonio Bicchi is with the Centro I. R. "E. Piaggio," Università di Pisa.

Francesco Bullo is with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, bullo@engineering.ucsb.edu .

[1] The connectivity of a graph is the maximum number of disjoint paths between any two vertices of the graph.

of the misbehaving agents, the resilience to external attacks is still limited by the connectivity of the network. In [8] 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 [9], [10], 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. In this paper, we extend and improve the results along these directions, e.g., by characterizing the complete set of policies that make a set of k malicious agents undetectable or unidentifiable, and by providing the resilience bounds in the case of faulty agents. Our approach also differs from the existing computer science literature, e.g., our analysis leads to the development of algorithms that can be easily extended to work on both discrete and continuous time linear consensus networks, and also with partial knowledge of the network topology.

The main contributions of this work are as follows. By recasting the problem of consensus computation in unreliable networks into a system theoretic framework, we provide alternative and constructive proofs of existing bounds on the number of identifiable Byzantine agents in a linear network. Precisely, we show that k malicious agents can be detected and identified if the network is $(2k+1)$-connected, and they cannot be identified if the network is $(2k)$-connected or less. We exhaustively describe the strategies that the malicious nodes can follow to disrupt a linear network that is not sufficiently connected. In particular, we prove that the inputs that allow the misbehaving agents to remain undetected or unidentified coincide with the zero inputs of a linear system associated with the consensus network. Also, we show that the set of such inputs has zero Lebesgue measure in the input space, so that it can be ignored if only faulty agents are considered. For the latter case (non-colluding agents), we provide a comprehensive novel analysis on the detection and identification of misbehaving agents problem. We show that k faulty agents can be identified if the network is $(k+1)$-connected, and they cannot if the network is k-connected or less. The proposed resilience bounds are shown to be generic with respect to the network communication weights, i.e., given a consensus topology, the bounds hold for almost all choices of the communication weights. In the last part of the paper, we discuss the problem of detecting and identifying misbehaving agents when either the partial knowledge of the network, or the hardware limitation, makes it impossible to implement the exact identification procedure. We describe a heuristic, which has low complexity, and that ultimately leads to a prompt recovery of the network functionalities from non-colluding malfunctions.

The rest of the paper is organized as follows. Section II recalls some basic facts on the fault detection and isolation problem for linear systems. In Section III we describe the consensus model under consideration. Section IV contains the conditions under which the misbehaving agents are detectable and identifiable, and Section V deals with the genericity of such conditions. Section VI presents our algorithmic procedures. Sections VII and VIII contain respectively our numerical studies and our conclusions.

II. Notation and preliminary concepts

We will be using the same notation as in [11]. Throughout the paper, let the triple (A, B, C) denote the linear discrete time system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{1}$$

and let the subspaces $\mathcal{B}$ and $\mathcal{C}$ denote respectively the image space $\mathrm{Im}(B)$ and the null space $\mathrm{Ker}(C)$. A subspace $\mathcal{V} \subseteq \mathcal{X}$ is a $(A, \mathcal{B})$-controlled invariant if $A\mathcal{V} \subseteq \mathcal{V} + \mathcal{B}$, while a subspace $\mathcal{S} \subseteq \mathcal{X}$ is a $(A, \mathcal{C})$-conditioned invariant if $A(\mathcal{S} \cap \mathcal{C}) \subseteq \mathcal{S}$. A state trajectory $x(t)$ of (1) can be controlled on a subspace $\mathcal{V}$ if and only if this is a $(A, \mathcal{B})$-controlled invariant. The set of all controlled invariants contained in $\mathcal{C}$ admits a supremum, which we denote with $\mathcal{V}^*$, and which corresponds to the locus of all possible state trajectories of (1) invisible at the output. On the other hand, a conditioned invariant $\mathcal{S}$ is a subspace such that there exists an observer for the system (1) for the factor space $\mathcal{X}/\mathcal{S}$. The set of the conditioned invariants containing $\mathcal{B}$ admits an infimum, which we denote with $\mathcal{S}^*$.

In a linear system, the presence of sensors failure and actuators malfunction is usually modeled by adding some unknown and unmeasurable functions u_i to the nominal system, so that the dynamic model becomes

$$\begin{aligned} x(t+1) &= Ax(t) + \sum_{i=1}^{m} B_i u_i(t), \\ y(t) &= Cx(t). \end{aligned} \tag{2}$$

The matrices B_i and the functions u_i are referred to as failure signatures and failure modes. By definition, when the failure i is not acting, the corresponding function u_i is constantly equal to zero. Given the system (2), the fault detection and isolation problem is to design a dynamic residual generator that takes the observables $y(t)$ and generates a set of residual vectors $r_i(t)$, such that 1) every residual $r_i(t)$ decays to zero if no failure is present, and 2) the nonzero residuals, allow to uniquely identify the failures. From [11], [12] we know the following result.

Theorem II.1 (Fault detection and isolation) *Consider the system $(A, [B_1 \cdots B_m], C)$, and let $K = \{1, \dots, m\}$. The fault detection and isolation problem is solvable if and only if*

$$\mathcal{B}_i \cap (\mathcal{V}^*_{K\setminus\{i\}} + \mathcal{S}^*_{K\setminus\{i\}}) = \emptyset, \;\; \forall i \in K, \tag{3}$$

where $\mathcal{V}^*_{K\setminus\{i\}}$ and $\mathcal{S}^*_{K\setminus\{i\}}$ are the maximal controlled and minimal conditioned invariant subspaces associated with the triple $(A, [B_{j_1} \cdots B_{j_{m-1}}], C)$, $j_1, \dots, j_{m-1} \in K \setminus \{i\}$.

Theorem II.1 guarantees the existence of a filter whose output r_i, which is referred to as the i-th residual, is affected only by the dynamics generated by the i-th failure signature.

Moreover, the transfer function between the signature i and the residual r_i is left-invertible, i.e., when the initial condition of the residual generator and that of the system coincide, the mapping from u_i to r_i is one to one. Note that Theorem II.1 does not provide the solvability conditions for the detection and identification of misbehaving agents in a linear consensus network, because the misbehaving nodes, and hence the failure signatures, are unknown.

III. Linear consensus in the presence of misbehaving agents

Let G be a directed graph, V its vertex set, and E its edge set. The connectivity of G is the maximum number of disjoint paths between any two vertices of the graph, or, equivalently, the minimum number of vertices in a vertex cutset [13]. Denote with N_i the neighbors set of the node $i \in V$, i.e., all the nodes $j \in V$ such that the pair $(j, i) \in E$. Consider the discrete time linear consensus system

$$x(t+1) = Ax(t), \tag{4}$$

in which the row stochastic and primitive matrix A is such that the (i, j)-th entry equals the weight of the communication edge from j to i, and in which the vector x contains the real numbers (state) associated with the agents [1]. In the sequel, we assume that the graph associated with the consensus matrix A is not complete.

As it is shown in [8], algorithms of the form (4) have no resilience to malfunctions and external attacks, and the failure of one or more agents prevents the entire network from reaching the desired consensus. We model the presence of the misbehaving node i using an exogenous input in the i-th position, so that, if the set of the misbehaving nodes is $K = \{i_1, i_2, \dots\} \subset V$, the consensus system becomes

$$x(t+1) = Ax(t) + B_K u_K(t), \tag{5}$$

where, being e_j the j-th vector of the canonical basis, the input matrix is $B_K = [e_{i_1}\ e_{i_2}\ \cdots]$. The *misbehaving agents* are allowed to update their state in an arbitrary way by choosing the input function u_K. In particular, the misbehaving agents are said to be *malicious* if they can inject any arbitrary function u_K, while they are said to be *faulty* if, given any proper subspace $\mathcal{V}$ of the state space, they are not able to confine the evolution of the state of the network on $\mathcal{V}$. In the malicious case, the worst case scenario for the network is considered, whereas, in the faulty case, atypical network dynamics, i.e., those lying on a subset of the state space of zero Lebesgue measure, are not taken into account. Note that our choice of keeping the matrix A fixed and of letting the class of inputs u_K unspecified models also the situation in which the misbehaving agents modify some entries of the matrix A, and the case of unreliable communication edges.

Remark 1 (Complete communication graph) *If the graph associated with A is complete, then any number of misbehaving agents can be identified. Indeed, in our model, each agent receives correctly the whole state of the network after the first consensus step, so that every agent can predict the evolution of the network, and hence identify the misbehaving agents. It follows that the number of malicious agents that a linear consensus networks can tolerate needs not be less than one third of the total number of nodes.*

IV. Detection and identification of misbehaving agents

Given a k-connected linear consensus network of the form (5), we associate an output matrix C_j with each agent j, which describes the information about the state of the network that is directly available to j. In particular, $y_j(t) = C_j x(t)$, and $C_j = [e_{n_1}\ \cdots\ e_{n_p}]^T$, $\{n_1, \dots, n_p\} \in N_j$. The problem of ensuring trustworthy computation among the agents of the network can be divided into a detection phase, in which the presence of the misbehaving components is revealed, and an identification phase, in which the identity of the misbehaving agents is discovered. From a system theoretic perspective, both tasks require certain observability properties of the consensus system. Let I represent the identity matrix of appropriate dimensions, the zero dynamics of the linear system (A, B_K, C_j) are the state trajectories invisible at the output, and can be characterized by means of the $(n+p) \times (n+m)$ pencil $P(z) = \begin{bmatrix} zI-A & B_K \\ C_j & 0 \end{bmatrix}$. The complex value $\bar{z}$ is said to be an invariant zero of the system (A, B_K, C_j) if there exists a zero state direction x_0, and a zero input direction g such that $(\bar{z}I - A)x_0 + B_K g = 0$. Finally, if *rank*$(P(z)) = n + m$ for all but finitely many complex values z, then the system (A, B_K, C_j) is left-invertible, i.e., there are no two distinct inputs that give rise to the same output sequence [14]. The zero dynamics are strictly related to the connectivity of the communication graph associated with the consensus algorithm.

Theorem IV.1 (Zero dynamics and connectivity) *Let (A, B_K, C_j) be a k-connected consensus system. If $|K| \geq k$, then there exists a set K and a node j such that the triple (A, B_K, C_j) has nontrivial zero dynamics. Moreover, if $|K| > k$, then there exists a set K and a node j such that the triple (A, B_K, C_j) is not left-invertible.*

Proof: Let G be the digraph associated with A, and let k be the connectivity of G. Take a set K of $k+1$ malicious nodes, such that k of them form a vertex cut S of G. The network G is divided into two subnetworks G_1, and G_3, which communicate only through the nodes S. Assume that the misbehaving agent $K \setminus S$ belongs to G_3, while the observing node j belongs to G_1. After relabeling the nodes, the consensus matrix A is of the form $\begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix}$. Let $u_S = -A_{23}x_3$, where x_3 is the vector containing the values of the nodes of G_3, and let $u_{K \setminus S}$ be any arbitrary nonzero function. Clearly, starting from the zero state, the values of the nodes of G_1 are constantly 0, while the subnetwork G_3 is driven by the misbehaving agent $K \setminus S$. We conclude that the triple (A, B_K, C_j) is not left-invertible. Suppose now that $K \equiv S$ as previously defined, and let $u_K = -A_{23}x_3$. Let the initial condition of the nodes of G_1 and of S be zero. Since every state trajectory generated by $x_3 \neq 0$ does not appear in the output of the agent j, the triple (A, B_K, C_j) has nontrivial zero dynamics. ■

A set of misbehaving agents may remain undetected from the observations of the node j if and only if there exists an initial condition of the network such that their behavior does not appear in the output sequence y_j.

Theorem IV.2 (Detectable malicious agents) *Let A be a consensus matrix, let V be the nodes of the network, and let $K \subset V$ be the set of malicious agents. The set K is detectable by the node $j \in V$ if and only if the system (A, B_K, C_j) has no nontrivial zero dynamics.*

Proof: It follows directly from the properties of a strongly observable system [14]. ■

Following Theorem IV.2, we can state a first upper bound on the number of malicious agents that can be detected.

Theorem IV.3 (Detection of malicious agents) *Given a k-connected linear consensus network, at most $k-1$ concurrent malicious agents can be detected by every node in the network.*

Proof: Suppose there are k malicious nodes, and that the communication graph associated with the consensus system is k-connected. Let the misbehaving agents form a vertex cut. Because of Theorem IV.1, for some output matrix C_j, the consensus system has nontrivial zero dynamics, so that the malicious nodes may remain undetected. ■

Let K_1 and K_2 be any two disjoint sets of possible misbehaving agents, and let x_1 and x_2 be the state trajectories generated by K_1 and K_2 with the inputs u_1 and u_2. Clearly, if the difference $x_1 - x_2$ belongs to the null space of the output matrix C_j, then it is not possible to determine from the observations of the agent j whether the set K_1 or the set K_2 is misbehaving.

Theorem IV.4 (Identifiable malicious agents) *Let A be a consensus matrix, let V be the nodes of the network, and let $K_1 \subset V$ be the set of malicious agents. The set K_1 is identifiable by the node $j \in V$ if and only if the system $(A, [B_{K_1} \ B_{K_2}], C_j)$ has no nontrivial zero dynamics, for every set $K_2 \subset V$ of possible misbehaving agents.*

Proof: (Only if) By contradiction, let x_0 and $[u_1 \ -u_2]^T$ be a zero state direction, and a zero input sequence for the system $(A, [B_{K_1} \ B_{K_2}], C_j)$. We have

$$y_j(t) = 0 = C_j \left(A^t x_0 + \sum_{\tau=0}^{t-1} A^{t-\tau-1} B_1 u_1(\tau) - \sum_{\tau=0}^{t-1} A^{t-\tau-1} B_2 u_2(\tau) \right)$$

where B_1 and B_2 are the input matrices associated with the sets K_1 and K_2. Therefore,

$$C_j \left(A^t x_0^1 + \sum_{\tau=0}^{t-1} A^{t-\tau-1} B_1 u_1(\tau) \right) = C_j \left(A^t x_0^2 + \sum_{\tau=0}^{t-1} A^{t-\tau-1} B_2 u_2(\tau) \right),$$

where $x_0^1 - x_0^2 = x_0$. Clearly, since the output sequence generated by K_1 coincide with the output sequence generated by K_2, the two sets of misbehaving nodes can not be distinguished.

(If) Recall that a system with no zero dynamics is strongly observable [14], i.e., there exists a unique pair of initial condition and input sequence that generates the output sequence. Let $\mathcal{K}$ be the set containing all the possible sets of misbehaving nodes, and let $K \in \mathcal{K}$ be the set of malicious nodes. Let Y be the vector containing the output sequence of the node j. Consider the systems $\Sigma_{i,l} = (A, [B_{K_i} \ B_{K_l}])$, with $K_i, K_l \in \mathcal{K}$, and $K_i \cap K_l = \emptyset$, and compute the input sequence, if any, that produces Y for every system $\Sigma_{i,l}$. Since each system Σ_{il} has no zero dynamics, there is a unique input sequence producing Y. In particular, whenever $K_i = K$, the input corresponding to the set K_l is zero, so that all the sets K_l, such that $K_l \cap K = \emptyset$, are recognized as well-behaving, and, by exclusion, the set K is identified. ■

As a consequence of Theorem IV.4, if up to k malicious agents are allowed to act in the network, then a necessary and sufficient condition to correctly identify the set of malicious nodes is that the consensus system subject to any set of $2k$ inputs has no nontrivial zero dynamics.

Theorem IV.5 (Identification of malicious agents) *Given a k-connected linear consensus network, at most $\lfloor \frac{k-1}{2} \rfloor$ malicious agents can be identified by every node in the network.*

Proof: Let K_1 and K_2 be two sets of $\lfloor \frac{k-1}{2} \rfloor + 1$ agents, and let K_1 be malicious. Since $2(\lfloor \frac{k-1}{2} \rfloor + 1) \geq k$, by Theorem IV.1 there exist K_1, K_2, and j such that the system $(A, [B_{K_1} \ B_{K_2}], C_j)$ has nontrivial zero dynamics. By Theorem IV.4, the set K_1 is not identifiable. ■

A complete characterization of the undetectable or unidentifiable malicious behaviors is derived from Theorem IV.4.

Theorem IV.6 (Undetectable and unidentifiable inputs) *Let A be a consensus matrix, and let $K_1 \subset V$ be the set of malicious agents. The set of inputs that make the agents K_1 undetectable coincide with the zero inputs of the system (A, B_{K_1}, C_j). Moreover, the functions u_{K_1} that make the set K_1 undistinguishable from the set $K_2 \subset V$ are such that there exists an input $[u_{K_1} \ u_{K_2}]^T$ that generates a zero dynamic for the system $(A, [B_{K_1} \ B_{K_2}], C_j)$.*

Proof: It follows directly from Theorem IV.4. ■

For a linear consensus network, Theorem IV.5 provides an alternative proof of the resilience bound first presented in [7] and later rediscovered in [9], and Theorem IV.6 fully characterizes the behaviors for which a group of malicious agents remains unidentified from the output observations of a certain node.

In most of the practical applications, it is too restrictive to assume that the misbehaving nodes are able to generate zero dynamics, since they need to be able to steer the state along particular directions, which have zero Lebesgue measure in the state space.[2] This motivates the study of the resilience of linear consensus networks to faulty (non-colluding) attacks, which, by definition, are not allowed to generate zero dynamics. The following theorem states an upper bound on the number of faulty agents that can be detected and identified.

[2] In a zero dynamic, the state is confined on the subspace $\mathcal{V}^*$, which is a proper space of the state space, and hence has zero Lebesgue measure.

Theorem IV.7 (Identification of faulty agents) *Given a k-connected linear consensus network, at most $k-1$ concurrent faulty agents can be detected and correctly identified by every node in the network.*

Proof: Since, by definition, faulty nodes do not generate zero dynamics, we only need to show that at most $k-1$ faulty agents can be correctly identified by every node. Suppose there are k faulty agents, and suppose that they form a vertex cut. The network is divided into two subnetworks G_1 and G_2 by the faulty nodes K. Let i be a node of G_2, and consider the problem of understanding, from the observations of the agent j of G_1, whether the set K or the agent i is faulty. As in the proof of Theorem IV.1, the system $(A, [B_S\ \ B_i], C_j])$ is not left-invertible, and, since every signal starting from i reaches j through the agents K, we have $\mathcal{B}_S \cap \mathcal{S}_i^* \neq 0$. From Theorem II.1, the dynamics generated by the two sets K and i can not be decoupled, and, in particular, the set K can reproduce the output sequence generated by any u_i. We conclude that j, and in fact any node in G_1, is not able to distinguish whether i, and in fact any set of nodes in G_2, or the set K is faulty. ■

Theorems IV.5 and IV.7 only give an upper bound on the maximum number of concurrent misbehaving agents that can be detected and identified. In the next section it will be shown that, generically, in order to detect and identify k malicious agents, the connectivity of the communication graph needs to be $2k+1$, while, for faulty agents, a $(k+1)$-connected network is sufficient. In other words, if there exists a set of k misbehaving nodes that can not be identified by the agent j, then a random and arbitrarily small change of the consensus matrix makes the misbehaving nodes detectable and identifiable with probability one, provided that the connectivity of the communication graph is sufficiently high.

V. Structural properties and generic solvability

We will be using some known results in the field of linear structured systems, and we refer the interested reader to [15], [16] for a detailed treatment of the subject. Given a linear structured system of the form

$$\begin{aligned} x(t+1) &= [A]x(t) + [B]u(t) \\ y(t) &= [C]x(t) + [D]u(t), \end{aligned} \tag{6}$$

we associate a directed graph $G = (V, E)$ with it. The vertex set V is given by $U \cup X \cup Y$, with $U = \{u_1, \dots, u_m\}$ the set of input vertices, $X = \{x_1, \dots, x_n\}$ the set of state vertices, and $Y = \{y_1, \dots, y_p\}$ the set of output vertices. The indices n, m, and p denote respectively the dimension of the state space, the input space, and the output space. Denoting (i, j) for a directed edge from the vertex i to the vertex j, the edge set E of G is $E_{[A]} \cup E_{[B]} \cup E_{[C]} \cup E_{[D]}$, with $E_{[A]} = \{(x_j, x_i) | [A]_{ij} \neq 0\}$, $E_{[B]} = \{(u_j, x_i) | [B]_{ij} \neq 0\}$, $E_{[C]} = \{(x_j, y_i) | [C]_{ij} \neq 0\}$, $E_{[D]} = \{(u_j, y_i) | [D]_{ij} \neq 0\}$. In the latter, for instance $[A]_{ij} \neq 0$ means that the entry (i, j) of the matrix $[A]$ is a nonzero parameter. A path, i.e., a sequence of vertices where each node is connected to the following one in the path, is simple if every vertex in the path occurs only once, and two paths are disjoint if they consist of disjoint sets of vertices. A set of l mutually disjoint and simple paths between two sets of vertices S_1 and S_2 is called a linking of size l from S_1 to S_2. A simple path in which the initial and the last vertex coincide is called cycle, and a cycle family of size l is a set of l mutually disjoint cycles. Finally, a path is called Y-topped if its end vertex is in the set Y. From [15] we know the following results.

Theorem V.1 (Generic normal rank of a matrix pencil) *Let $P(z)$ be the system pencil of the structured system (6). The normal rank of $P(z)$ is generically equal to n plus the size of a maximum linking from U to Y.*

In other words, for almost any numerical realization Σ of the structure matrices $([A], [B], [C], [D])$, the normal rank of the pencil of Σ equals n plus the size of a maximum linking from the input to the output vertices. Recall that the union of a linking, a Y-topped path family and a cycle family is disjoint if they mutually have no vertices in common.

Theorem V.2 (Generic number of invariant zeros) *Let the pencil $P(z)$ of the structured system (6) have full column rank $n+m$, even after the deletion of an arbitrary row. The generic number of invariant zeros of the system (6) is equal to n minus the maximal number of vertices in X contained in the disjoint union of the following sets:*

(i) *a linking of size m from U to Y,*
(ii) *a set of cycles in X, and*
(iii) *a set of Y-topped paths.*

For our purposes, assume $[D] = 0$, and note that the connectivity of the graph associated with a structured system $([A], [B], [C])$ can be used to characterize the zero dynamics of almost all numerical realization of $([A], [B], [C])$.

Theorem V.3 (Generic zero dynamics and connectivity) *Let $([A], [B], [C])$ be a k-connected structured system. If the number of independent columns of $[B]$ is less than k, then almost any numerical realization of $([A], [B], [C])$ has no zero dynamics.*

Proof: Consider the digraph G associated with the structured system $([A], [B], [C])$, and let $P(z)$ be its matrix pencil. Because of Theorem V.1, $P(z)$ has full normal rank $n + |U|$, so that almost any realization of $([A], [B], [C])$ is left-invertible. Deleting the row v from $P(z)$ corresponds to deleting all the incoming edges to the node v. Let $G^{'}$ be the digraph associated with $P(z)$ after deleting one of its rows. Since G is k-connected, $G^{'}$ is at least $k-1$ connected. The maximum size of a linking from U to Y is still $|U|$, and hence $P(z)$ has full normal rank even after the deletion of an arbitrary row. By considering a set of n self loops in G, which are always present in our consensus model, we have that all the n vertices in X are covered, and therefore, by Theorem V.2, almost any realization of $([A], [B], [C])$ has no invariant zeros. ■

Given a structured triple $([A], [B], [C])$ with δ nonzero elements, the set of parameters that make $([A], [B], [C])$ a consensus system is a subset S of $\mathbb{R}^\delta$, because the matrix A needs to be nonnegative and row stochastic. A certain property that holds generically in $\mathbb{R}^\delta$ needs not be valid generically with respect to the feasible set S. However, a consensus system with no zero dynamics can generically be found.

Theorem V.4 (Genericity of consensus systems) *Let* $([A],[B],[C])$ *be a* k*-connected structured system. If the number of independent columns of* $[B]$ *is less than* k*, then, for almost every nonnegative numerical realization of* $([A],[B],[C])$*, there exists a consensus system with no nontrivial zero dynamics.*

Proof: Let (A,B,C) be a nonnegative numerical instance of $([A],[B],[C])$. The set of parameters for which a generic property fails to hold coincides by definition with an algebraic hypersurface of the parameter space [16], so that a property remains generic when the parameter set is restricted to the nonnegative orthant. Because of Theorem V.3, the triple (A,B,C) has generically no zero dynamics. Moreover, the Perron-Frobenius Theorem for nonnegative matrices ensures the existence of a positive eigenvector x for the matrix A associated with the eigenvalue of largest magnitude r [17]. Let D be the diagonal matrix whose main diagonal equals x, then the matrix $r^{-1}D^{-1}AD$ is a consensus matrix [18]. A similarity transformation using D yields the system $(D^{-1}AD, D^{-1}B, CD)$, which generically has also no zero dynamics. Finally, the system $(r^{-1}D^{-1}AD, D^{-1}B, CD)$ is a k-connected consensus system with, generically, no zero dynamics. Indeed, if there exists a value $\bar{z}$, a zero direction x_0, and a zero input direction g for the system $(r^{-1}D^{-1}AD, D^{-1}B, CD)$, then the value $\bar{z}r$, with state direction x_0/r and input direction u, is an invariant zero of $(D^{-1}AD, D^{-1}B, CD)$, which contradicts the hypothesis. ■

Following Theorem V.4, we are able to state our results concerning the resilience of a linear consensus network to external attacks.

Theorem V.5 (Generic identification of malicious agents) *Given a* k*-connected consensus network, up to* $\lfloor \frac{k-1}{2} \rfloor$ *malicious agents can generically be detected and correctly identified by any agent.*

Proof: Since $2\lfloor \frac{k-1}{2} \rfloor < k$, by Theorem V.3 the consensus system with any set of $2\lfloor \frac{k-1}{2} \rfloor$ has generically no zero dynamics. By Theorem IV.4, any set of $\lfloor \frac{k-1}{2} \rfloor$ malicious agents is detectable and identifiable by any node in the network. ■

We conclude this Section with the resilience bound for the faulty agents model.

Theorem V.6 (Generic identification of faulty agents) *Given a* k*-connected consensus network, up to* $k-1$ *faulty agents can generically be detected and correctly identified by any agent.*

Proof: Let V be the set of nodes, and $K \subset V$ the set of faulty agents. Let k be the connectivity of the graph associated with a structure matrix $[A]$, and let $|K| = k-1$ be the rank of the input matrix B_K. By virtue of Theorem V.3, almost any numerical instance of $([A],[B_K],[C_j])$ has no zero dynamics, regardless of the choice of j, and therefore $\mathcal{V}_K^* = \emptyset$. Let $i \in V \setminus K$, and consider the system $\Sigma_i = (A, [B_K \;\; B_i], C_j)$, $j \in V$. Since the number of inputs in Σ_i equals the connectivity of G, the system Σ_i is generically left-invertible because of Theorem V.1. Therefore, $\mathcal{B}_i \cap (\mathcal{V}_K^* + \mathcal{S}_K^*) = \mathcal{B}_i \cap \mathcal{S}_K^* = 0$, for any set of k intruders $K \cup \{i\}$. The dynamics of the K intruders can be fully decoupled from the output trajectory generated by any other node i, and therefore up to $k-1$ faulty nodes are successfully detected and identified. Indeed, for each $i \in V \setminus K$, the residual associated with i in the system $(A, [B_K \;\; B_i], C_j)$ converges to zero, so that the agent i is regarded as well-behaving, and, by exclusion, the set K is identified. Note that, since the faulty agents are not allowed to inject the inputs described in Lemma IV.6, there is no other set of agents able to generate the output observations. ■

VI. Detection and identification algorithms

A distributed procedure to detect and identify the misbehaving agents in a linear consensus network is in Algorithm 1. Here is an informal description.

> *(Exact identification)* We focus on the agent j. Let k be the number of misbehaving nodes to be identified, and let $\mathcal{K}$ be the set containing all the $\binom{n-1}{k+1}$ combinations of $k+1$ elements of $V \setminus \{j\}$. For each set $\tilde{K} \in \mathcal{K}$, consider the system $\Sigma_{\tilde{K}} = (A, B_{\tilde{K}}, C_j)$, and compute[3] a set of residual generator filters for $\Sigma_{\tilde{K}}$. If the connectivity of the communication graph is sufficiently high, then, as described in the previous sections, each residual function is nonzero if and only if the corresponding failure mode is active. Let K be the set of misbehaving nodes, then, whenever $K \subset \tilde{K}$, the residual function associated with the failure mode $\tilde{K} \setminus K$ becomes zero after an initial transient, so that the agent $\tilde{K} \setminus K$ is recognized as well-behaving. By exclusion, because the residuals associated with the misbehaving agents are always nonzero, the set K is identified.

Notice that, since the residual generators are dead beat filters, the detection and the identification of the misbehaving agents take place in finite time, and that, because each agent only relies on its local observations, no communication overhead is introduced in the consensus protocol.

Algorithm 1 requires every agent to know the entire topology of the network, and to compute a number of residuals which grows exponentially with the number of nodes of the network. In a more realistic scenario each agent is only aware of the communication structure of some neighborhood, and they can perform only a certain number of operation in a reasonable amount of time. It follows that, in practice, the proposed procedure is implementable only on a small consensus network. In the sequel, we briefly present a heuristic to address this issue.

Consider the set $V_j^d \subset V$ of the d-neighbors of the agent j, i.e., the set of nodes within distance d from the agent j,

[3] A procedure to design a residual generator filter can be found in [12].

Algorithm 1: Detection and identification of misbehaving agents in a linear consensus network.

```
Input   : Consensus matrix; number of misbehaving
          nodes $k$;
Require: The communication graph has
          connectivity $k+1$, if only faulty agents
          are allowed, and $2k+1$ otherwise;
Each agent computes the residual generators for
every possible set of $k+1$ misbehaving agents;
while the misbehaving agents are unidentified do
    Exchange data with the neighbors;
    Update the state;
    Evaluate the residual functions;
    if every $i_{th}$ residual is nonzero then
        Agent $i$ is recognized as misbehaving.
```

and let A_j^d be the matrix describing the interaction among the nodes V_j^d. Let $V_j^d = \{\bar{v}_1, \dots, \bar{v}_l\}$, then, for all $i, k \in \{1, \dots, l\}$, the (i,k)-th entry of A_j^d equals

- the $(\bar{v}_i, \bar{v}_k)$-th entry of A if $N_{\bar{v}_i} \subseteq V_j^d$;
- $1/|N_{\bar{v}_i} \cap V_j^d|$ if $N_{\bar{v}_i} \not\subseteq V_j^d$, and if the $(\bar{v}_i, \bar{v}_k)$-th entry of A is positive; and
- 0 otherwise.

For a set of possible misbehaving agents K, let $\Sigma_{K,j}^d = (A_j^d, B_{K,j}^d, C_j^d)$ denote the reduced system computed by the agent j, where $B_j^d = HB_K$, $C_j^d = CH^T$, $H = [e_{\bar{v}_1} \ \dots \ e_{\bar{v}_l}]^T$. In the absence of misbehaving nodes, the residual functions associated with the reduced system $\Sigma_{K,j}^d$ asymptotically decay to zero.

Theorem VI.1 (Convergence of residuals) *Let $\Sigma_{K,j}^d$ be the reduced consensus system computed by the agent j. In the absence of misbehaving agents, and for every set K of possible misbehaving agents, the residual functions computed by j decay to zero.*

Proof: Let $\Sigma_{K,j}^d$ be the reduced system of the agent j. Note that the residual functions for a consensus system are not affected by the state trajectories lying on the subspace **1**, because the state of a consensus system converges to **1**, and the residuals are designed to decay to zero in the absence of misbehaving nodes. Finally, since in the absence of misbehaving agents both the consensus system and $\Sigma_{K,j}^d$ converge to the subspace **1**, the residuals computed by j decay to zero. ■

Because the evolution of the reduced system $\Sigma_{K,j}^d$ differs from the dynamic of the consensus system Σ_K, the residual generators designed using the system $\Sigma_{K,j}^d$ do not provide exact decoupling for the trajectories of the system Σ_K. In other words, every residual function is in general nonzero, so that the identification of the misbehaving set needs to rely on a threshold mechanism. As in [19], we design the residual generators so that a good compromise between sensitivity to faults and robustness to noise is achieved. As in Fig. 1, the magnitude of the residual functions associated with the faulty agents turns out to be larger than the magnitude of the residual functions associated with the well-behaving nodes, so that a correct identification of the misbehaving set is generally still possible. Here is a description of our heuristic, a detailed version of which is forthcoming.

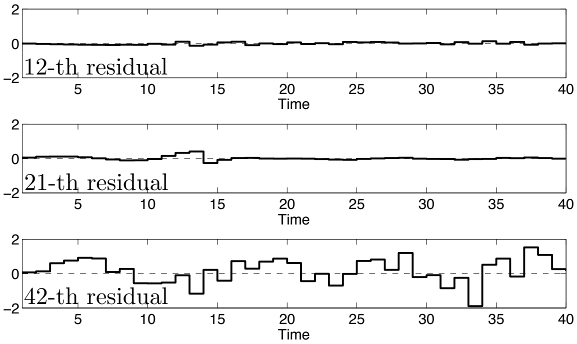


Fig. 1. The residual generators are designed to achieve the best compromise between sensitivity to faults and robustness to noise. As an example, for the network in Fig. 3(a), the magnitude of the 42-th residual, computed by the agent 32, is sensibly larger than the 12-th and the 21-th residual.

> *(Low-complexity identification)* We focus on the agent j. Let k be the number of misbehaving nodes to be identified, and let $\mathcal{K}$ be the set containing all the $\binom{|V_j^d|-1}{k+1}$ combination of $k+1$ elements of $V_j^d \setminus \{j\}$. For each set $\tilde{K} \in \mathcal{K}$, consider the system $\Sigma_{\tilde{K},j}^d$, and compute a set of residual generator filters for $\Sigma_{\tilde{K},j}^d$. Compare the residuals with a predetermined threshold and identify the misbehaving agents.

Note that, the required memory and the computational burden are a function of d, and not of the dimension of the network.

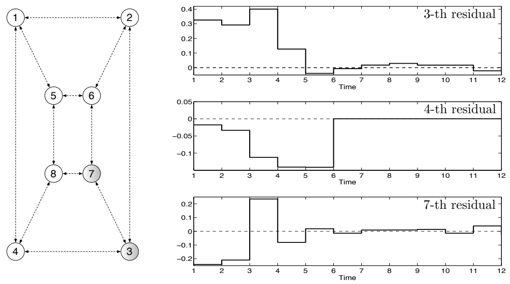
VII. Examples

A. Exact detection and identification

Consider the network of Fig. 2(a), and let A be a randomly chosen consensus matrix. In particular,

$$A = \begin{bmatrix} 0.2795 & 0.1628 & 0 & 0.1512 & 0.4066 & 0 & 0 & 0 \\ 0.0143 & 0.3363 & 0.3469 & 0 & 0 & 0.3025 & 0 & 0 \\ 0 & 0.0718 & 0.1904 & 0.2438 & 0 & 0 & 0.4941 & 0 \\ 0.0844 & 0 & 0.4457 & 0.0660 & 0 & 0 & 0 & 0.4040 \\ 0.1709 & 0 & 0 & 0 & 0.2694 & 0.2472 & 0 & 0.3125 \\ 0 & 0.4199 & 0 & 0 & 0.1575 & 0.3293 & 0.0932 & 0 \\ 0 & 0 & 0.0174 & 0 & 0 & 0.4241 & 0.2850 & 0.2735 \\ 0 & 0 & 0 & 0.3024 & 0.2039 & 0 & 0.2065 & 0.2873 \end{bmatrix}.$$

The network is 3-connected, and it can be verified that for any set K of 3 misbehaving agents, and for any observer node j, the triple (A, B_K, C_j) is left-invertible. Also, for any set K of cardinality 2 the triple (A, B_K, C_j) has no invariant zeros. As previously discussed, any well-behaving node can detect and identify up to 2 faulty agents, or up to 1 malicious agent. Consider the observations of the agent 1, and suppose that the agents $\{3, 7\}$ inject a random signal into the network. As described in Algorithm 1, the agent 1 computes the residual functions for each of the $\binom{7}{3}$ possible sets of misbehaving nodes, and identify the well-behaving agents. For example, independent of the initial condition of the network, for the system $x(t+1) = Ax(t) + B_3u_3(t) + B_4u_4(t) + B_7u_7(t)$, after 7 time steps, the residual function associated with the input 4 is zero, as in 2(b), so that the agent 4 is regarded



(a) (b)

Fig. 2. In Fig. 2(b) a consensus network where the nodes 3 and 7 are faulty. In Fig. 2(a) the residual functions computed by the agent 1, under the hypothesis that the misbehaving set is $\{3, 4, 7\}$.

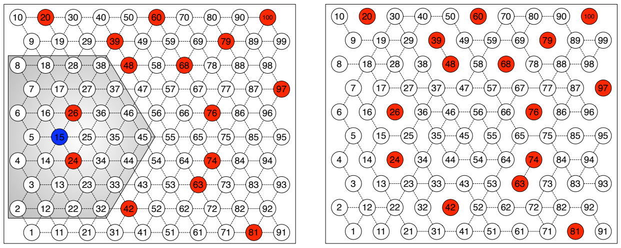
as well-behaving. The agents 3 and 7 instead, since they always have nonzero residual functions, are recognized as misbehaving. If the misbehaving nodes are malicious, then no more than 1 misbehaving node can be tolerated. Indeed, the consensus system with 4 inputs exhibits nontrivial zero dynamics, so that a set of 2 malicious nodes may remain unidentified. For example, the system $(A, B_{\{2,4,6,8\}}, C_1)$ has nontrivial zero dynamics, since the nodes $\{2, 4, 6, 8\}$ form a vertex cut. It follows that there exists an initial condition and an input function such that the nodes $\{2, 4\}$ and $\{6, 8\}$ generate the same output observations, and, therefore, can not be distinguished.

B. Approximate detection and isolation

The goal of the following example is to show that an effective detection and identification mechanism can be designed using the heuristic presented in Section VI. Suppose that the network topology is as in Fig. 3(a), and that the agents only know the structure of a 3-neighborhood, as the shaded region in Fig. 3(a) for the agent 15. We will be considering only faulty agents, because the partial knowledge of the network makes it impossible to identify malicious nodes. Suppose that the 15 red agents in Fig. 3(a) are faulty, and suppose that they add a random signal to the consensus algorithm. The agents design the residual generators for the portion of consensus network they know, and they execute 40 steps of the consensus algorithm. By comparing the residuals with a predetermined threshold, all the misbehaving agents are identified and isolated from the network, as in Fig. 3(b). Clearly, because the identification algorithm is not exact, some communication edges are cut erroneously.

VIII. Conclusions

The problem of distributed reliable computation in networks with misbehaving nodes is considered, and its relationship with the fault detection and isolation problem for linear systems is discussed. The resilience of linear consensus networks to external attacks is characterized through some properties of the underlying communication graph, as well as from a system-theoretic perspective. In almost every linear consensus network, the misbehaving components can be correctly detected and identified, as long as the connectivity of the communication graph is sufficiently high. Precisely, for a linear distributed consensus network to be resilient to k concurrent faults, the connectivity of the communication graph needs to be $2k + 1$, if Byzantine failures are allowed, and $k+1$, otherwise. Finally, for the faulty agents case, good performance can be obtained even when the agents do not know the entire topology of the consensus network, or when they are subject to memory or computation constraints.



(a) (b)

Fig. 3. In Fig. 3(a) a consensus network with 100 nodes, 15 of which are faulty. In Fig. 3(b) the result of the detection and identification heuristic. All the faulty agents are isolated from the network of the well-behaving agents.

References

[1] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control: Collective group behavior through local interaction,” *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

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

[4] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1997.

[5] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems & Control Letters*, vol. 53, pp. 65–78, 2004.

[6] 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.

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

[8] 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.

[9] 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, Jun. 2008, pp. 1350–1355.

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

[11] G. Basile and G. Marro, *Controlled and Conditioned Invariants in Linear System Theory*. Prentice Hall, 1991.

[12] M.-A. Massoumnia, G. C. Verghese, and A. S. Willsky, “Failure detection and identification,” *IEEE Transactions on Automatic Control*, vol. 34, no. 3, pp. 316–321, 1989.

[13] C. D. Godsil and G. F. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2001, vol. 207.

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

[15] J. M. Dion, C. Commault, and J. van der Woude, “Generic properties and control of linear structured systems: a survey,” *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[16] W. M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 3rd ed. Springer, 1985.

[17] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.

[18] H. Minc, *Nonnegative Matrices*. John Wiley, 1988.

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