Stochastic and Deterministic Byzantine Fault Detection for Randomized Gossip Algorithms

Daniel Silvestre, Paulo Rosa, João P. Hespanha, Carlos Silvestre

Abstract

This paper addresses the problem of detecting Byzantine faults in linear randomized gossip algorithms, where the selection of the dynamics matrix is stochastic. A Byzantine fault is a disturbance signal injected by an attacker to corrupt the states of the nodes. We propose the use of Set-Valued Observers (SVOs) to detect if the state observations are compatible with the system dynamics for the worst case in a deterministic setting. The concept of Stochastic Set-Valued Observers (SSVOs) is also introduced to construct a set that is guaranteed to contain all possible states with, at least, a pre-specified desired probability. The proposed algorithm is stable in the sense that it requires a finite number of vertices to represent polytopic sets and it allows for the computation of the largest magnitude of the disturbance that an attacker can inject without being detected. Results are presented to reduce the computational cost of this approach and, in particular, by considering only local information and representing the remainder of the network as a disturbance. The case of a consensus algorithm is discussed leading to the conclusion that, by using the proposed SVOs, finite-time consensus is achieved in non-faulty environments. A novel algorithm is proposed that produces less conservative set-valued state estimates by having nodes exchanging local estimates. The algorithm inherits all the previous properties and also enables finite-time consensus computation regardless of the value of the horizon.

Index Terms

Fault detection, distinguishability, finite-time consensus, randomized gossip algorithms.

I. Introduction

The problem of detecting faults in an asynchronous distributed environment relates to determining if any of the nodes enters in an incoherent state given the observed history of measurements. In particular, we are interested in randomized algorithms where the dynamics is common to all the nodes and no control messages are needed. This class of algorithms is used for iterative solutions because they offer a certain level of robustness against packet drops and node failure. Applications of randomized algorithms [1] range from computing integrals to consensus [2] and solving problems for which the solution requires a heavy computational burden [3] [4] [5]. Large scale distributed systems and the use of robot swarms highlight the importance of this problem for practical applications.

The aim of this paper is to detect the presence of an attacker who corrupts the states of the nodes or their transmissions. In this context, the small probability of an event cannot be discarded as an attacker can select the worst case signal, which motivates the use of set-valued estimation tools. Therefore, we address the problem in a distributed manner where each node models the network from its perspective as a Linear Parameter-Varying (LPV) system, where the input is the attacker signal. Since an attacker is allowed to inject any signal, we are looking at the worst case scenario and estimating the set of all possible state realizations that comply with the "fault-free" model. If the set becomes empty, we can guarantee the presence of an attacker (Byzantine fault) or any other fault.

Byzantine fault detection methods have been proposed in the literature for a number of specific applications. For instance, [6] focuses on detection in the case of a consensus problem by using unreliable fault detectors, where multiple classes of theoretic detectors are presented. The proposed method checks if the algorithm is running correctly and if all the messages are in concordance with the specifications. The research interest in Byzantine faults has motivated a number of contributions including the scenario of unreliable networks in distributed systems. In particular, [7] considers the problem of detecting and correcting the state of the system in the presence of a Byzantine fault. The case of malicious agents and faulty agents is studied and the authors provide, in both cases, bounds on the number of corrupted nodes to ensure detectability of the fault. In [7], the system dynamics are described by a linear time-invariant model that constrains the communications in each time slot to be from a fixed set of senders to a set of receivers. Here, however, a randomized gossip algorithm is considered, thus dropping the assumption that the same set of nodes is every time involved in message exchanges.

D. Silvestre is with the Dep. of Electrical and Computer Engineering, Instituto Superior Técnico, ISR, 1046-001 Lisboa, Portugal. This work was supported by the project FCT [UID/EEA/50009/2013] and with grant SFRH//BD/71206/2010, from Fundação para a Ciência e a Tecnologia. dsilvestre@isr.ist.utl.pt

P. Rosa is with Deimos Engenharia, Lisbon, Portugal. paulo.rosa@deimos.com.pt.

C. Silvestre is with the Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Macau, Macau, China, on leave from Instituto Superior Técnico/Technical University of Lisbon, 1049-001 Lisbon, Portugal. The work was supported by project MYRG117(Y1-L3)-FST12-MKM of the University of Macau. csilvestre@umac.mo

João P. Hespanha is with the Dept. of Electrical and Computer Eng., University of California, Santa Barbara, CA 93106-9560, USA. J. Hespanha was supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grants No. W911NF-09-1-0553 and W911NF-09-D-0001. hespanha@ece.ucsb.edu

The adopted strategy for fault detection has an interesting finite-time property that can be used for consensus problems. Finite-time consensus in the presence of malicious agents has been addressed in [8], where the authors show that the topology of the network categorizes its ability to deal with attacks. Both the number of corrupted nodes and vertex-disjoint paths in the network influence its resilience. In [8], it is assumed a broadcast model where, at each transmission time, the nodes send to all their neighbors the same value and the agents objective is to compute some function of the initial states. The main difference to the work described herein is the communications model, which we assume to be *gossip*, where pairs of nodes are selected randomly to exchange information, instead of having a broadcast model.

A subset of the results described herein was previously presented in the conference papers [9] and [10] by the same authors. In [9], the concept of Stochastic Set-Valued Observers (SSVOs) was introduced by resorting to the use of α-confidence sets, i.e., sets where the state of the system is known to belong with a desired pre-specified $1-\alpha$ probability; which can be viewed as a generalization of confidence intervals. The property of finite-time consensus when using (deterministic) Set-Valued Observers (SVOs) for a sufficiently large horizon in a randomized gossip consensus algorithm is shown in [10].

The main contributions of this paper are as follows:

- it is presented how to compute a threshold for the "maximum impact" of an undetected fault and the particular case of consensus and physical systems are discussed;
- the number of required communications for guaranteeing detection is reduced by analyzing the structure of randomized gossip algorithms;
- finally, we show how some of the dynamics matrices can be discarded from the model depending on the existence of failed transmissions of other nodes and links connecting neighbors, which reduces the computational complexity of the algorithm.

Besides the development of a theoretical framework to address the problem at hand, it is also needed to cover the mathematical machinery required to cope with the computation of the set where the current state can take values. From the random behavior of the gossip algorithm, a set-valued estimate requires the union over all possible transmission of the set of possible state realizations originated by that transmission and the previous state. By definition, the number of sets grows exponentially with the horizon N. We resort to the concept of SVOs for this task, firstly introduced in [11] and [12]. For the interested reader, further information can be found in [13] and [14] and the references therein.

An alternative to the use of SVOs is the use of zonotopes, described in [15] and further developed in [16] and [17]. Zonotopes represent a different trade-off between the computation complexity of unions and intersections. In particular, intersections introduce conservatism which motivated the alternative approach adopted in this article in order to attain the desired convergence guarantees, while keeping the computational requirements to a tractable level.

In the literature, there are other examples of fault detection systems that employ gossip algorithms in order to achieve scalability. In [18], the proposed protocol aims at detecting faults by using a gossip like communication. The work differs from our proposal in the sense that the protocol is limited to determining unreachable nodes and does not cope well in the presence of attackers.

The applicability of the proposed method in the detection of faults in a randomized gossip algorithms spans other purposes as several challenges in the Fault Detection and Isolation (FDI) literature - [19], [20] - share the framework described in the sequel. In [21], [22], the authors take advantage of SVOs for fault detection by resorting to a model falsification approach. This paper extends the results in [21], [22] to detect Byzantine faults in randomized gossip algorithms by rewriting the associated dynamics as an LPV model. Moreover, unlike the approach in [21], [22], the method proposed herein takes into account the information related to the probability of having a given communication, in order to reduce the conservatism of the results.

In [23], three algorithms are proposed for gossip-like fault detection in distributed consensus over large-scale networks, namely round-robin, binary round-robin and round-robin with sequence check. These improve upon the basic randomized version by constructing a better gossip list and reducing the probability for false positives. The algorithms are particularly design for the consensus problem in its version where all the nodes must select a value among the initial set of values. Our algorithm aims at detecting faults for general iterative linear distributed algorithms that can be subject to sensor noise or other effects that render the detection non-trivial.

Closely related to the concept of stochastic detection is the work presented in [24] which performs the detection by finding the change points in the correlation statistics for a sensoring network. The authors are able to provide guarantees on detection delay and false alarm probability. Such approach addresses a similar problem of detecting faults that are possible in the standard dynamics but not very "probable" to take place. Our work tackles this issue in a different way by considering the set of possible states given the more "probable" dynamics.

In the context of fault detection in distributed systems, [25] addresses the problem by looking at the whole system and constructing a batch of observers for each sub-system. By looking at the outputs of these observers it is possible to detect and isolate faults affecting one of the sub-systems. However, it is a centralized approach whereas our focus is to run each of the observers locally at each sub-system in a fully distributed way.

In [26], the authors propose an on-line fault detection and isolation algorithm for linear discrete-time uncertain systems where the detection is based on the computation of a upper and lower bound for the fault signal. The calculations are performed resorting to Linear Matrix Inequality (LMI) optimization techniques. Similar computational burden considerations to the work

presented in this article are discussed and the techniques are related to our work. However, in order to address randomized gossip algorithms we studied a more general class of systems.

Using the approach of design residual filters, [27] studies the class of linear continuous-time systems with the purpose of identifying faulty actuators. The aim is to adjust the filters parameters as to decouple them when faults affect a group of actuators. Our approach differs in the sense that we want to incorporate unknown parameters in the dynamics matrix of the system.

The organization of this paper develops towards presenting all the details of fault detection for the worst-case and in the stochastic sense for distributed linear systems. Initial focus is given to distributed gossip systems and their key elements and constraints posed on the detection, namely, the characteristics associated with the network component and how faults are modeled. The concept of SVOs is introduced and applied to the deterministic fault detection, as the worst-case is considered. Progress is made in presenting a method to extend the SVOs computation to incorporate the stochastic information of the communication process, which results in the SSVOs.

The SVO-based fault detection method motivates the introduction of a consensus algorithm that performs averages on intervals of where the state can take place, intersecting them upon neighbor communication. The algorithm is asymptotically convergent and also has the advantage that under some communication patterns allows finding the consensus value in finite-time due to the intersection phase. Therefore, this paper is proposing an SVO-based approach to fault detection with different types of SVOs which should not be confused. The deterministic worst-case detection is an SVO which a node can run to perform the fault detection using only locally available information. The stochastic detection is an extension of the previous method with the set of state estimates being a subset of the previous one corresponding to a confidence set of where the state can take values. Lastly, in the particular case of consensus, we propose an algorithm that takes advantage of the local estimates and intersects them upon communication to generate less conservative sets.

Notation : The transpose of a matrix A is denoted by $A^\intercal$. For vectors a_i, $(a_1, \ldots, a_n) := [a_1^\intercal \ \ldots a_n^\intercal]^\intercal$. The notation $[a]_i$ represents the ith component of vector a. We let $1_n := [1 \ldots 1]^\intercal$ and $0_n := [0 \ldots 0]^\intercal$ indicate n-dimensional vector of ones and zeros, respectively, and I_n denotes the identity matrix of dimension n where the vector e_i represents the canonical vector corresponding to the ith column of the matrix I_n. Dimensions are omitted when clear from context. The vector e_i denotes the canonical vector whose components are equal to zero, except for the ith element. The symbol $\otimes$ denotes the kronecker product. The notation $||.||$ refers to $\|v\| := \sup_i |v_i|$ for a vector v, and $\|A\| := \bar{\sigma}(A)$ for a matrix A. The projection operator $P_q X$ of a polytope X onto a vector q is defined as the line segment that results from the standard orthogonal projection of any point $p \in X$ to the line that passes in the origin and is defined by the vector q.

II. Problem Statement

We consider a set of n_x agents, each of which with scalar state $x_i(k)$, $1 \leq i \leq n_x$. At each transmission time, each node i chooses a random out-neighbor j, according to the communication topology modeled by a connectivity graph $G = (V, E)$, where V represents the set of n_x agents, also denoted by nodes, and $E \subseteq V \times V$ is the set of communication links. Node i can send a message to node j, if $(i, j) \in E$. If there exists at least one $i \in V$ such that $(i, i) \in E$ we say that the graph has self-loops and node i has only access to its own value at that transmission time. We associate to graph G a *weighted adjacency matrix* W with entries:

$$[W]_{ij} := \begin{cases} w_{ij}, \text{ if } (i,j) \in E \\ 0, \text{ otherwise} \end{cases}, \tag{1}$$

where the weight $w_{ij} \in [0, 1]$ is the probability of the link between node i and node j being selected for communication and, therefore, $W 1_{n_x} = 1_{n_x}$.

The "fault-free" gossip algorithm can be defined by the dynamics discrete-time equation

$$x(k+1) = A(k)x(k), \tag{2}$$

where the matrix $A(k)$ is selected randomly from a set $\{Q_{ij}, (i, j) \in E\}$ (i.e., $A(k) = Q_{ij}$ with probability w_{ij}) modeling the process by which nodes select a random out-neighbor, as described above, and where $x(k) = [x_1(k), \cdots, x_{n_x}(k)]^\intercal$. Matrices Q_{ij} implement the update on state variables x_i and x_j caused by a transmission from node i to node j and represent a set of matrices that are equal to the identity except for rows i and j. In this paper, we assume symmetry in the communication and update rule, meaning that both rows i and j are equal (which implies matrices $A(k)$ to be symmetric), and no further structure is assumed regarding the linear iteration.

The "fault-free" algorithm in equation (2) is modified to include faults resulting in:

$$x(k+1) = A(k)x(k) + B(k)u(k), \tag{3}$$

where the input, $u(k)$, models the fact that some of the nodes may either report incorrect values regarding their state value or update their state by something other than the "fault-free" version. In particular, the case of an attacker trying to exploit the algorithm weaknesses motivates to consider any input signal $u(k)$ [7].

The objective of the detection algorithm is to use only limited information provided by local interactions between nodes in the network. A node performing the detection does not have access to all the communications between the remaining nodes. Indeed, the output of the system from the perspective of node i, $y^i(k)$, at time k, is composed of the states that were involved in the communication at time instant k with that node. In other words, if node j transmitted to node i at time k, then $y^i(k)$ will be the vector with the states x_i and x_j i.e., $y^i(k) = C_i(k)x(k)$, with $C_i = [\mathsf{e}_i, \mathsf{e}_j]^\intercal$ and will only have its own state if the node did not communicate ($C_i(k) = [\mathsf{e}_i, \mathsf{e}_i]^\intercal$)[1]. With a slight abuse of notation, we use $y^i(k)$ to refer to the output of the system at time k and $y^i_k(x(0), u_k)$ to express the same output as a function of the initial state $x(0)$ and input u_k, where u_k denotes the sequence of inputs up to time k.

The full dynamics S_i for node i, as defined above, refers to the pair of equations:

$$S_i : \begin{cases} x(k+1) = A(k)x(k) + B(k)u(k) \\ y^i(k) = C_i(k)x(k) \end{cases} \tag{4}$$

The main goal of this paper can therefore be stated as: developing algorithms for detecting nonzero inputs $u(k)$ in (3) that do not require knowledge of the matrices $B(k)$[2] and signal $u(k)$ and, instead, only use the measured variables y^i_k, which stands for all the measurements up to time k, as in (4).

We introduce the following definition:

Definition 1 (undetectable faults): Take the randomized gossip system modeled by (4) from node i's perspective. A nonzero input sequence u_k (corresponding to a fault) is said to be *undetectable in N measurements* if for some transmission sequence:

$$\forall_{k<N}, \exists_{x(0),x'(0)\in W_o} : y^i_k(x(0), u_k) = y^i_k(x'(0), 0)$$

where W_o is a set where initial state $x(0)$ is known to belong to. Otherwise, it is said to be *detectable*.

□

The intuition behind this definition is that a fault is only guaranteed to be detectable if there is no possible set of initial conditions such that the sequence $y^i(0), \cdots, y^i(N)$ of measurable states can be generated without an attacker signal. The fault being detectable as in Definition 1 relates to the observability of the system, as described in [28]. Notice that if the fault does not satisfy Definition 1, it cannot be guaranteed its detection with probability 1. The mechanism presented throughout this paper can still detect such faults depending on the sequence of transmissions and the initial state of the nodes.

In summary, the problem being tackled in this paper relates to detecting any fault which cannot be generated by a "fault-free" model only with the knowledge of local measurements of the node state itself and those to which it communicates. The fault detection mechanism is distributed and no global knowledge of which nodes are communicating is assumed and neither is known the nodes or the communication links affected by the attacker.

III. Fault Detection using Set-Valued Observers (SVOs)

In this section, we analyze the fault detection problem from a deterministic point of view, and recast the network within the LPV framework. As a consequence, the random selection of matrices $A(k)$ is disregarded and all realizations of the sequence of matrices $A(k)$ are considered regardless of their probabilities. Firstly, we start by rewriting the matrices $A(k)$ in (3) as the sum of a single central matrix A_0 with parameter-dependent terms:

$$A(k) = A_0 + \sum_{\ell=1}^{n_\Delta} \Delta_\ell(k) A_\ell \tag{5}$$

where each $\Delta_\ell(k)$, $\forall k \geq 0$ is a scalar uncertainty with $|\Delta_\ell(k)| \leq 1$, and the A_ℓ, $\ell \in \{1, 2, \ldots, n_\Delta\}$ a sufficiently rich collection of matrices so that all the $A(k)$ can be written as in (5). For the sake of simplicity, we also denote $\Delta(k) = [\Delta_1(k), \cdots, \Delta_{n_\Delta}(k)]^T$ as the vector of uncertain parameters at times k.

As an example consider a simple network with 3 nodes running a gossip consensus algorithm and let us look only at nodes 1 and 2, which we assume to have 3 different dynamics matrices

$$Q_{12} = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}, Q_{21} = \begin{bmatrix} 0.25 & 0.75 & 0 \\ 0.75 & 0.25 & 0 \\ 0 & 0 & 1 \end{bmatrix}, Q_{11} = Q_{22} = I$$

where Q_{11} and Q_{22} represent missed transmissions from node 1 and node 2 respectively. For that case, we can design the matrices A_0 and A_ℓ to be

$$A_0 = Q_{12}, A_1 = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

[1] Alternatively, one can consider simply $C_i(k) = e_i^\intercal$, although this would imply that the size of vector $y^i(k)$ depends on k.

[2] since the focus is on fault detection rather than fault isolation, we generate sets where the state of the "fault-free" system must be which does not require the knowledge of matrices $B(k)$

and matrix $Q_{11} = Q_{22} = A_0 + A_1$, $Q_{12} = A_0$ and $Q_{21} = A_0 - 0.5A_1$. Therefore, for 3 possible transmission matrices we only require 1 uncertainty (i.e., $n_\Delta = 1$) and reduce the complexity of the algorithm.

The dynamics of the system can now be cast into an LPV model with uncertainty in the time-varying matrix $A(k)$. Indeed, the dynamics in (3) can be rewritten as:

$$x(k+1) = \Big(A_0 + \sum_{\ell=1}^{n_\Delta} \Delta_\ell(k) A_\ell\Big) x(k) + B(k)u(k). \tag{6}$$

Detecting a fault in a worst-case scenario amounts to finding whether there exists an admissible initial condition $x(0)$ such that a given sequence of observations, y_k^i, can be generated by the dynamics in (6) with $u(k) = 0$ for $k \in \{0, 1, \cdots, N\}$. Therefore, the knowledge of the structure of $B(k)$ is not needed for fault detection.

A fault-free (ideal) SVO for (4) is a dynamical system that produces a sequence of sets $X(k), k \geq 0$ such that each $X(k)$ is the smallest set that contains all possible values of the state $x(k)$ of (4) that are compatible with the zero inputs $u(0) = u(1) = \cdots = u(k-1) = 0$ and the observed outputs $y^i(0), y^i(1), \cdots, y^i(k)$ of node i.

Assumption 1 (bounded state): For a "fault-free" system the following holds: $\forall k < N, \forall i : 1 \leq i \leq n_x, |x_i(k)| < c$ for a given constant c.

□

Assumption 1 is sustained by the fact that a non-faulty gossip algorithm has a bounded state. Therefore, there exists a constant c such that if the absolute value of the state is larger than c, one can trivially detect the occurrence of the fault. Assumption 1 is fundamental for enclosing the initial state in a polytope and compute the set $X(k)$ as described in the next proposition.

To prepare the proposition, we introduce some notation. A polytope at time k is defined as $\text{Set}(M, m) := \{q : Mq + m \leq 0\}$ whereas we also introduce the notation $M_{\Delta^\star}(k)$ and $m_{\Delta^\star}(k)$ to refer the polytope for a particular instantiation of the uncertainties. Similarly, $A_{\Delta^\star}$ refers to the particular instantiation of the dynamics matrix using $\Delta^\star$ value for the uncertainties.

We also recall the definition of the Fourier-Motzkin as

Definition 2 (Fourier-Motzkin [30]): Take a polytope described by $\left\{ \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^{n_x+n_y} : A \begin{bmatrix} x \\ y \end{bmatrix} \leq b \right\}$. The Fourier-Motzkin elimination method is a function

$$(A_{\text{FM}}, b_{\text{FM}}) = \text{FM}(A, b, n_x)$$

such that

$$A_{\text{FM}}\, y \leq b_{\text{FM}} \Leftrightarrow \exists_{x \in \mathbb{R}^{n_x}} : A \begin{bmatrix} x \\ y \end{bmatrix} \leq b.$$

Intuitively, we will compute the polytope containing the state for each of the vertices of the hypercube containing the vector $\Delta(k)$. For that reason, we show how to compute for a particular vertex (i.e., when $\Delta(k)$ is a constant equal to one of the hypercube vertices) and then compute the convex hull of all the sets.

Proposition 1 ($X(k+1)$ computation [31]): Consider a system described by (6), with $u(\cdot) \equiv 0$, and where $x(k)$ denotes the corresponding state at times k, for $k \geq 0$. Further assume that

- $x(0) \in X(0)$, where $X(0) = \text{Set}(M(0), m(0))$, for some matrix $M(0)$ and vector $m(0)$ with appropriate dimensions;
- $\Delta(k) \equiv \Delta^\star$, for some (constant) vector $\Delta^\star$ and all $k \geq 0$;
- $A_o + A_{\Delta^\star}$ is non-singular.

Then, the set $X(k+1) := \text{Set}(M_{\Delta^\star}(k+1), m_{\Delta^\star}(k+1))$, which contains all the possible states of the system at time $k+1$, can be described by the set of points, $\mathbf{x}$, satisfying the equation

$$\underbrace{\begin{bmatrix} M(k)(A_0 + A_{\Delta^\star})^{-1} \\ C_i(k+1) \\ -C_i(k+1) \end{bmatrix}}_{M_{\Delta^\star}(k+1)} \mathbf{x} \leq \underbrace{\begin{bmatrix} -m(k) \\ y^i(k+1) \\ -y^i(k+1) \end{bmatrix}}_{-m_{\Delta^\star}(k+1)} \tag{7}$$

where

$$A_{\Delta^\star} = \sum_{\ell=1}^{n_\Delta} \Delta_\ell^\star A_\ell$$

and $\Delta_\ell^\star$ is the realization of the uncertainty for the current transmission time. When the dynamics matrices are not invertible, the set is given by solving the inequality relating the current time $\mathbf{x}$ and the previous time with $\mathbf{x}^-$

$$\begin{bmatrix} I & -A_0 - A_{\Delta^\star} \\ -I & A_0 + A_{\Delta^\star} \\ C_i(k+1) & 0 \\ -C_i(k+1) & 0 \\ 0 & M(k) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^- \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ y^i(k+1) \\ -y^i(k+1) \\ -m(k) \end{bmatrix} \tag{8}$$

and applying the Fourier-Motzkin elimination method [32] (see Definition 2) to remove the dependence on $\mathbf{x}^-$ and obtain the set described by $M_{\Delta^\star}(k+1)\mathbf{x} \leq -m_{\Delta^\star}(k+1)$.

Inequality (8) can be extended to a generic horizon and extending the inequality to the following:

$$\begin{bmatrix} I & -\tilde{A}_0^k & \cdots & 0 \\ -I & \tilde{A}_0^k & \cdots & 0 \\ I & 0 & \cdots & 0 \\ -I & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & 0 & \cdots & -\tilde{A}_{N-1}^k \\ -I & 0 & \cdots & \tilde{A}_{N-1}^k \\ C_i(k+1) & 0 & \cdots & 0 \\ -C_i(k+1) & 0 & \cdots & 0 \\ 0 & C_i(k) & \cdots & 0 \\ 0 & -C_i(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & C_i(k+1-N) \\ 0 & \cdots & 0 & -C_i(k+1-N) \\ 0 & M(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & M(k+1-N) \end{bmatrix} \begin{bmatrix} \mathbf{x(k+1)} \\ \vdots \\ \mathbf{x(k+1-N)} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ y^i(k+1) \\ -y^i(k+1) \\ y^i(k) \\ -y^i(k) \\ \vdots \\ y^i(k+1-N) \\ -y^i(k+1-N) \\ -m(k) \\ \vdots \\ -m(k+1-N) \end{bmatrix} \tag{9}$$

where the notation $\mathbf{x(k+1-N)}$ is a variable to constrain the state at N time instants before the current time and $\tilde{A}_n^k := (A_0 + A_{\Delta(k)}) \cdots (A_0 + A_{\Delta(k-n)})$.

□

The previous proposition describes the set of possible states at time $k+1$ for a particular instantiation of $\Delta(k)$, which considers no uncertainty in the system. In order to compute the set $X(k+1)$, one would need to make the union of the sets for all possible instantiations of the uncertainties. As a consequence, set $X(k+1)$ is, in general, non-convex which renders its calculation computationally heavy. For that reason, we are interested here in polytopical SVOs that produce the smallest sets of the form $\tilde{X}(k) := \text{Set}(M(k), m(k))$ that contain the sets $X(k)$ produced by the fault-free (ideal) SVO. Polytopical SVOs thus produce the smallest over approximation of the sets produced by the ideal SVO.

For a given *horizon*, N, let the coordinates of each vertex of the hypercube $H := \{\delta \in \mathbb{R}^{n_\Delta N} : |\delta| \leq 1\}$ be denoted by $\theta_i, i = 1, \cdots, 2^{n_\Delta N}$. Using (7) (or (8)), let us compute $X_{\theta_i}(k)$. Thus, the smallest set comprising all possible states of the system described by (6), with $|\Delta_\ell(k)| \leq 1$ and $u(\cdot) \equiv 0$, at time $k+1$ can be obtained by

$$\tilde{X}(k+1) = \text{co}\Big(\bigcup_{\theta_i \in H} \text{Set}(M_{\theta_i}(k+1), m_{\theta_i}(k+1)) \Big) \tag{10}$$

where co() denotes the convex hull. The vertices θ_i should not be confused with the network agents, since they represent the possible combinations of the uncertainty parameters. The convex hull in (10) can be performed using the methods described in [21], [22]. It is straightforward to conclude that $X(k+1) \subseteq \tilde{X}(k+1)$. We recall Proposition 6.2 in [33] for completeness.

Proposition 2 (Growth of $\tilde{X}(k)$): Consider a system described by (6) with $x(0) \in X(0)$ and $u(k) = 0, \forall k$, and suppose that there exists an $N^\star \geq 0$ such that

$$\gamma_N := \max_{\substack{\Delta(k), \cdots, \Delta(k+N) \\ |\Delta(m)| \leq 1, \forall m \\ k \geq 0}} \Big|\Big| \prod_{j=k}^{k+N} \mathcal{A}(j) \Big|\Big| < 1,$$

for all $N \geq N^\star$, and where

$$\mathcal{A}(j) := \Big[A_0 + \sum_{i=1}^{n_\Delta} \Delta_i(j) A_i \Big].$$

Then, it is possible to find a set $X^o(k), \forall k$ with uniformly bounded hypervolume and number of vertices, such that $\tilde{X}(k) \subseteq X^o(k)$.

□

In summary, Proposition 2 states that the volume of $\tilde{X}(k)$ is uniformly bounded for all $k \geq 0$, and that there is a hyper-parallelepiped that, at each time, contains the set $\tilde{X}(k)$, and has a uniformly bounded distance between any two vertices, for all $k \geq 0$.

Notice that the method provided before to compute $M(k)$ and $m(k)$ for the "fault-free" model, gives a set where the measurements can take values. Whenever this operation results in an empty set, the "fault-free" virtual system cannot generate the real system measurements and a fault is detected. In addition, in reference to Proposition 2, we can always derive a bounded set with a finite number of vertices to contain the set of actual possible states, $X(k)$.

The complexity of the algorithm to compute the set-valued estimates for the state is exponential in nature, since the number of vertices of the hypercube to be considered is $2^{n_\triangle N}$. The number of uncertainties, in a worst-case scenario, is equal to the number of vertices of the connectivity graph, as we can trivially associate each uncertainty with each possible communication link and define appropriate matrices A_ℓ in (6).

In order to reduce the SVO complexity, it is essential to either consider a smaller horizon or decrease the number of edges in the connectivity graph relevant to our problem. One of the main contributions of this paper is the derivation of an upper bound on the smallest horizon guaranteeing fault detection in the case of randomized gossip algorithms. Under mild assumptions, detection can be guaranteed for a sufficiently large number of observations. Let such horizon be referred to as $N^\star$. However, the combinatorial behavior of the detection problem renders the computation of the SVO intractable and, in practical applications, N may be considerably smaller than $N^\star$. In other words, the horizon used by the algorithm may be small, so as to guarantee its practical implementability and still performing the detection at the expenses of a longer detection time.

We now present a proposition to reduce the number of necessary edges by discarding irrelevant information in a worst-case perspective, when the horizon is smaller than the theoretical value of $N^\star$.

Proposition 3 (SVO with local information): Let a node i be running an SVO of a system described by (6) with $x(0) \in X(0) := \{z \in \mathbb{R}^{n_x} : \|z\|_\infty \leq c\}$ and $u(k) = 0, \forall k \leq N$ with $N < N^\star$. Further suppose that the connectivity graph G satisfies either one of the following conditions:

1) $\exists j \neq i : (j,j) \in E$
2) $\exists_{q_1,q_2 \neq i} : q_1, q_2 \neq j, j \in \{\ell : (i,\ell) \in E\}, (q_1,q_2) \in E$

Then, $\forall q \neq i, q \neq j$ we get $\forall k, P_q X(k) = \begin{bmatrix} -c, & c \end{bmatrix}$, where P_q is the projection operator on the q dimension and c is the constant in Assumption 1 such that $P_q X(0) = \begin{bmatrix} -c, & c \end{bmatrix}$ and $X(k)$ is the set generated by the ideal SVO.

□

Proof. Take the initial bounds of node i for all states, i. e., $\forall q, P_q X(0) = \begin{bmatrix} -c, & c \end{bmatrix}$, by assumption.

If condition 1) holds, then select $j \neq i$, $j : (j,j) \in E$ and consider the sequence of communications obtained from selecting the matrix Q_{jj} for k consecutive time instants. Since self-loops map failed transmissions,

$$Q_{jj} = I \implies X(k) = X(0)$$

which means that the initial set cannot be reduced due to failed transmissions. Then,

$$\forall k, q : P_q X(k) = \begin{bmatrix} -c, & c \end{bmatrix}.$$

If condition 2) holds, take q_1 and q_2 and consider successive transmissions $Q_{q_1 q_2}$. By noticing that neither node i nor node j, with $j \in \{\ell : (i,\ell) \in E\}$, are involved in any communication and that only those nodes can be observed by node i, we get

$$\forall k, q \neq i, q \neq j \in \{\ell : (i,\ell) \in E\} : P_q X(k) = \begin{bmatrix} -c, & c \end{bmatrix}$$

Since we found at least one communication pattern that makes $X(k) = X(0)$. ■

Proposition 3 provides conditions to determine which communications are necessary for the SVOs, and which ones can be disregarding without degrading performance. In particular, condition 2) means that any node apart from the first and second degree neighbors may be discarded and included in the model as a single disturbance to each second degree neighbor, since no direct measurement is ever performed. This fact comes directly from the limitation that the node running the observer cannot distinguish between second degree nodes, as it only communicates with its neighbors.

A detection mechanism is only interesting in practice if its complexity scales well with the number of nodes in the network. We showed that the set can be computed using only local information without loss of accuracy if $N < N^\star$. To produce accurate estimates, intuitively, we need all the information regarding observations that are available to build a smaller set at the expenses of propagating those observations with the system dynamics. However, we can relax this definition and discard old information that does not enhance the set-valued state estimate, according to the next theorem. We introduce the notation $M^N(k)$ to explicitly indicate the horizon N for which the set is computed. In the next theorem, for the sake of simplicity, $X_n(k) := \text{Set}(M^n(k), m^n(k))$.

Theorem 1: Take a system as defined in (6) and consider an SVO, running in node i, with local information. It is always possible to find N such that

$$\forall n > N, \forall q, \exists n^\star \leq N : P_q(X_{n^\star}(k)) \subseteq P_q(X_n(k)). \tag{11}$$

Then, $X_N(k+1)) \subseteq X_n(k+1)$.

□

Proof. From Proposition 3, we get

$$\forall N > 0, \forall q \neq i, (q,i) \notin E, P_q X^N(k+1) = [-c, c],$$

which leads to the conclusion that the set-valued state estimates of each node are not directly affected by its neighbors.

For a horizon $N = 1$, from equation (6), the set $X^1(1)$ is obtained using $\theta_i = 1, \cdots, 2^{n_\Delta}$. If a communication with node i happens then $\theta_i = \theta_i^\star$, where $\theta_i^\star$ corresponds to an instantiation of the uncertainties for that communication. For a generic N, if the node did not communicate with any of its neighbors, then $X^N(k)$ is computed using $\theta_i \times \cdots \times \theta_i$, where $\times$ represents the Cartesian product and is taken N times. From this fact, with the last observation measured at time k_q results in $\forall n > k - k_q, P_q X^{k-k_q}(k) \subseteq P_q X^n(k)$. By definition, since an observation is the equivalent of setting $\theta_i = \theta_i^\star$ for a particular instant, then $P_q X^n(k) \subseteq P_q X^{k-k_q}(k)$. Thus,

$$P_q X^n(k) = P_q X^{k-k_q}(k). \tag{12}$$

Since the condition $P_q X^{n^\star}(k) \subseteq P_q X^n(k)$ holds, this means that $\exists k_q, \forall q : (i,q) \in E : A(k_q) = Q_{iq}$. Therefore, applying (12), we reach to the conclusion

$$X^N(k+1) \subseteq X^n(k+1).$$

■

The intuition behind Theorem 1 is that we do not need to consider past time instances prior to the last communication that we established with each node. The horizon value N must be sufficiently large as to have node i (the node running the SVO) communicating with all its neighbors and the previous time instants can be neglected.

Remark 1 (Bound in the Horizon): From Theorem 1, the set $X^N(k)$, when N is selected such that there exists a transmission between all the neighbors and the node, and such that local information is selected as in Proposition 3, is the smallest possible set.

IV. Fault Detection using Stochastic Set-Valued Observers (SSVO)

SVOs are deterministic and discard the probabilistic information of each event. They consider as admissible all states that can be generated by the considered LPV dynamics, regardless of how likely they are. By taking into account the stochastic information in the definition of the SVO, one may decide to declare a fault when the observations are, in principle, possible, but have an exceedingly small probability of occurrence. This typically permits the earlier detection of attacks, at the expense of generating *false alarms*. The algorithm proposed in the sequel allows for controlling the probability of false alarms.

To better understand how probabilitic information can help detect faults, consider the 5-node complete network ($n_x = 5$) and time horizon to detect the fault $N = 20$. Each node i takes a measurement $x_i(0)$ of a quantity of interest and then starts a linear randomized gossip algorithm. Let us assume that the packet drop probability is known. In particular, let $p_{\text{drop}} = 0.01$ where a packet drop is represented as a transmission from node i to itself, using the transmission matrix $Q_{ii} = I$. Each node is chosen with probability $\frac{1}{n_x}$ and each matrix Q_{ij} representing a successful transmission from node i to j has probability $\frac{w_{ij}}{n_x}$.

If a node is not involved in a communication, it is only able to determine its own state. Suppose that the states of the agents start dissimilar from each other but that during the first N time steps, all agents are faulty and keep their states unchanged, i.e., $x(k) = x(0), \forall k \leq N$. This fault is undetectable according to Definition 1, since there is a sequence of matrices $A(k)$ that mimic the same behavior, which is a sequence of 20 failed transmissions due to the physical medium. Consequently, if the algorithm in the previous section is used, $x(k) = x(0)$ must remain in the set $\tilde{X}(k), \forall k$ and therefore a fault will not be detected. However, the probability of obtaining the sequence $x(k) = x(0), \forall k \leq N$ is extremely small:

$$\text{Prob}[\{x(k) = x(0), \forall 0 \leq k \leq 20\}] = 10^{-40}$$

and is more likely to be a fault. The inability of the SVO to incorporate the probability associated with each event is, therefore, a significant drawback. Such an example motivates the introduction of Stochastic Set-Valued Observers (SSVOs) where the polytope containing the possible state is associated with a probability. The objective of this section concerns with extending the SVO concept to cope with the probability of getting a given sequence of measurements. With that target in mind, we introduce the definition of α-confidence sets.

Definition 3 (α-confidence sets): The set $\bar{X}(k)$ is an α-confidence set at time k for a system of the form (4) with state $x(k)$ if

$$\text{Prob}[x(k) \in \bar{X}(k)] \geq 1 - \alpha.$$

Consider the algorithm described in the previous subsection to generate the sets $\tilde{X}(k)$ and recall that we rewrote each Q_{ij} as in (6), therefore associating with each hypercube vertex θ_{ij} a transmission matrix Q_{ij} with correspondent probability w_{ij}. The objective of this section is to construct the α-confident set, as in Definition 3 as to associate the probability of the events in the fault detection.

Take the map $\psi : \theta_i \mapsto E$ which gives the correspondence between the vertices of the hypercube H and the edges in set E and let us collect the minimum number of vertices θ_{ij} in Θ such that $\sum_{\theta_{ij}} w_{\psi(\theta_{ij})} \geq 1 - \alpha$. The set for the SSVO $\bar{X}(k)$ is then an α-confidence set defined as:

$$\bar{X}(k) := \text{co}\Big(\bigcup_{\theta_{ij} \in \Theta} \text{Set}(M_{\theta_{ij}}(k), m_{\theta_{ij}}(k)) \Big) \tag{13}$$

Computationally, it requires to sort the vertices θ_{ij} according to probabilities $w_{\psi(\theta_{ij})}$ as to construct Θ and then determining $M_{\theta_{ij}}(k)$ and $m_{\theta_{ij}}(k)$ as before. θ_{ij} depends on the selected edges and there can be multiple sets Θ generating an α-confidence set, with similar characteristics.

In the next Property, we establish that the set generated by the SSVO is an α-confidence set. In this context, the parameter α can be viewed both as the probability of false positives and also as a similar concept as the confidence interval for stochastic variables.

Property 1: Take the definition of $\bar{X}(k)$ as in (13). Then, $\forall k, \bar{X}(k)$ is a α-confidence set.

Proof. The result is straightforward from the fact

$$\begin{aligned} \text{Prob}\Big[x(k) \in \bigcup_{\theta_i \in \Theta} \text{Set}(M_{\theta_i}(k), m_{\theta_i}(k))\Big] &\geq \sum_{\theta_i \in \Theta} w_{\psi(\theta_i)} \\ &\geq 1 - \alpha \end{aligned}$$

■

Property 1 establishes the SSVOs as a generalization of the SVOs since the set $\tilde{X}(k)$ is an α-confidence set with $\alpha = 0$ and, therefore, we have $\bar{X}(k) \subseteq \tilde{X}(k)$.

Taking advantage of the definition of SSVOs, we introduce Algorithm 1 for probabilistic detection of faults. The construction of the set $\bar{X}(k)$ ensures that, with probability $1 - \alpha$, the state $x(k)$ belongs to $\bar{X}(k)$ and thus is an α-confidence set.

Algorithm 1 Detection using SSVO

Require: Set $\bar{X}(0)$, the probability matrix W and the confidence level α.
Ensure: Computation at each time instant k of $\bar{X}(k)$: Prob$[x(k) \in \bar{X}(k)] \geq \alpha$ and Fault Detection.

for each k **do**
  /* *Finding the set* Θ */
  $\Theta = \min$ *card*$(\{\theta_{ij}\})$
    s.t. $\sum w_{\phi(}\theta_{ij}) \geq 1 - \alpha$
  /* *Build the set* $\bar{X}(k+1)$ */
  SSVO_iteration($\Theta, \bar{X}(k), y(k+1)$*)*
  /* *Check if* $\bar{X}(k+1)$ *is empty* */
  if $\bar{X}(k+1) = \emptyset$ **then**
    return *System is faulty*
  end if
end for

Notice that, in Algorithm 1, the function *SSVO_iteration* is implementing the procedure to compute the set-valued estimates defined in (7) or (8), using the uncertainty values stored in Θ. In essence, the SSVO propagation is exactly the same as the standard SVO except for the fact that less uncertainties are considered in the hypercube, due to the fact that the vertices having low probability of occuring are not considered. Detection is ensured if we make the bounded assumption as in Assumption 1, and also that the transmission selection procedure operates as described in Section II. Detection guarantees will be provided later in this paper, with a further discussion on the meaning of a detection using Algorithm 1.

V. Byzantine Consensus Algorithm

In this section, we describe how the information used to construct the set of possible states can be used to introduce a novel algorithm to compute consensus of intervals in a distributed way, and detect if a fault has occurred.

In a consensus system, we are referring to the agents running a distributed iterative algorithm that guarantees convergence of the state to its initial average value, i.e.,

$$\lim_{k \to \infty} x_i(k) = x_{\text{av}} := \frac{1}{n_x} \sum_{i=1}^{n_x} x_i(0). \tag{14}$$

We refer to this problem as the *average consensus problem*. This problem can be tackled by a standard algorithm (such as [2]) and then, an SVO-based overlay to detect faults such as in [9]. In this section, an algorithm is introduced that incorporates the information used to construct the local estimate (i.e., a given node's estimate) of possible states and reduce conservatism by intersecting it with the state estimates from its neighbors. In the process, the set of possible states is reduced and the consensus solution is reached in finite time.

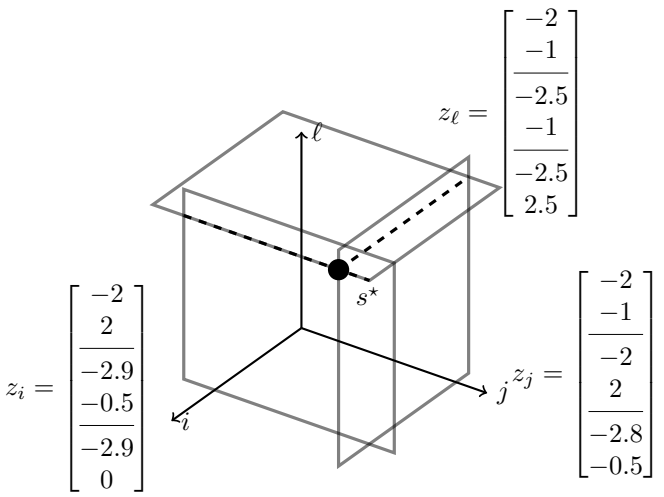

Fig. 1: Example of the set-valued estimates boundaries of node i (yellow), node j (green) and node ℓ (red), where for each node there is no uncertainty regarding its own state and where $s^\star$ represents the full state of the system that is contained in all three state boundaries.

Each node runs an SVO to determine the set of possible states of all the nodes in the network. With a slight abuse of notation, we will denote $X_i(k)$ for the set computed by node i which contains estimates for the states of all the nodes in the network using the measurements performed by node i. In general, the result of the Fourier-Motzkin elimination method produces a polytopic set with a bounded number of vertices. However, transmitting the set $X_i(k)$ would mean communicating the matrix $M_i(k)$ and vector $m_i(k)$, which define the set-valued state estimate $X_i(k)$. Since the dimension of $M_i(k)$ depends on the number of vertices, we might need to communicate a large amount of information, which may not be feasible in many applications.

For that reason, we can overbound this uncertainty set by a hyper-parallelepiped Set $\left(\hat{M}_i(k), z_i(k)\right)$, with

$$\hat{M}_i(k) = I \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

and $z_i(k) \in \mathbb{R}^{2n_x}$, where $z_i(k)$ is defined such that Set$\left(\hat{M}_i(k), z_i(k)\right) := \{q : \hat{M}_i(k)q + z_i(k) \leq 0\}$ contains $X_i(k)$. Using this approach, $z_i(k)$ will be the only vector that we need to transmit between neighbors. Thus, the $z_i(k)$'s represent state boundaries for the other agents and are obtained through the previously described algorithm to compute the SVO (7) or (8), by using the local information available when communicating with the neighbors.

An important issue here is the possible large conservativeness of the upperbounding of $X_i(k)$ by a hyper-parallelepiped set. However, in order to minimize this issue, one can increase the horizon and consider more measurements in building $X_i(k)$ and, therefore, getting better estimates [33]. Thus, there is a trade-off between speed of computation of the SVO and its conservativeness when selecting the horizon.

The algorithm (see flow chart in Fig. 2) can be briefly described as follows: in each discrete time instant, each node that does not communicate with its neighbors updates its set-valued state estimates of the corresponding SVO using (7) or (8). If node i communicates with node j, then it proceeds to an intersection of both set-valued state estimates motivated by the fact that z_i and z_j are estimates for the state boundaries of all nodes constructed using the information available to node i and j, respectively. The intersection step is described using the maximum function (z variables represent intervals and were defined to have the minimum and the maximum multiplied by -1, see Fig. 1 for a numeric example) by operating on the state of the two communicating nodes i and j

$$z_i(k) = z_j(k) = \max(z_i(k), z_j(k)) \tag{15}$$

where the $\max$ function, which operates row-wise, returns a column vector of the same length.

The result of performing the intersections can be described by $s^\star = \left[[z_1]_1^\intercal, [z_1]_2^\intercal, \cdots, [z_i]_{2i-1}^\intercal, [z_i]_{2i}^\intercal\right]^\intercal$ and represents the collaborative estimation performed by all the nodes since $s^\star \in$ Set$(\hat{M}_i, z_i)$ and $s^\star \in$ Set$(\hat{M}_j, z_j)$. The concept of $s^\star$ and the state boundaries generated by each node with the corresponding z variable is illustrated in Fig. 1. A fault is declared by node i, whenever it receives z_j from node j, with $[z_i]_{2j-1} > [z_j]_{2j-1} \vee [z_i]_{2j} > [z_j]_{2j}$. This means that their estimates do not intersect and there is no vector $s^\star$ of possible states that satisfies the observations made by the different nodes in the network.

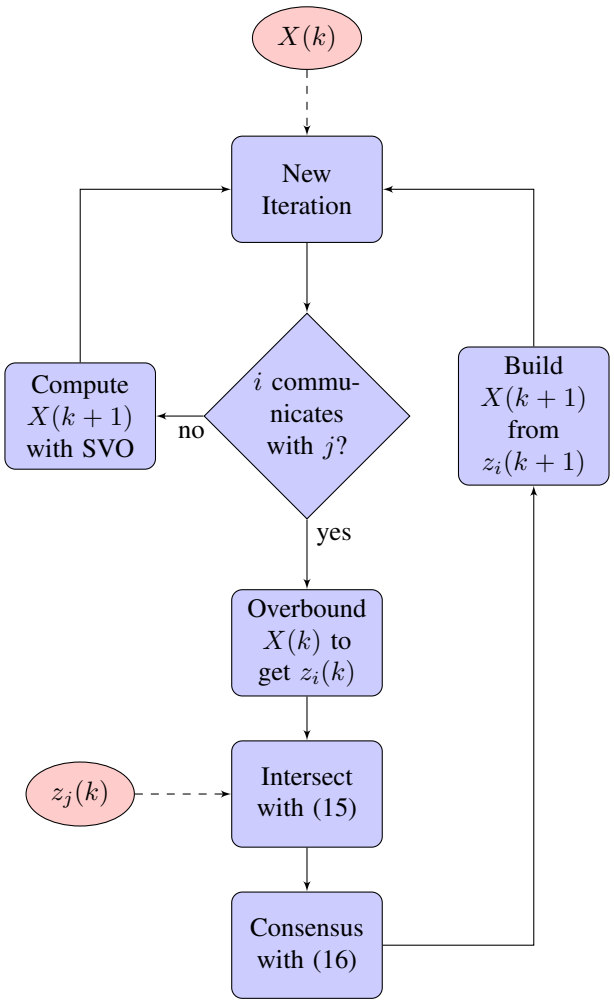

Fig. 2: Flowchart of the algorithm with the intersection phase to share observations between neighbors.

At each time k, the consensus phase runs in both communicating nodes and is defined for node i communicating with node j by the following linear iteration, similarly to what is done in [2]:

$$z_i(k+1) = \left[\left(\frac{1}{2}(e_i - e_j)(e_j - e_i)^\intercal + I_{n_x}\right) \otimes I_2\right] z_i(k) \tag{16}$$

where, as previously mentioned, the variable z_i is the vector-valued estimate of node i of all the states of the nodes of the network. It should be noticed that, for node i, we may have $[z_i]_{2i} \neq [z_i]_{2i-1}$ if there is uncertainty associated to it.

As a remark, the algorithm defined through (15) and (16), and in Fig. 2, not only computes the consensus value of its state, but also keeps estimates for all the remaining ones, using observations made by the node itself and its neighbors. This algorithm differs from the one proposed in [9] in the sense that the estimates of the SVO in each node are used to compute the state boundaries $z_i(k)$ at each time instant and then shared with the neighbors when communicating, producing an intersection of measurements that is then subjected to the standard gossip consensus step.

Definition 4: We say that a linear distributed algorithm taking the form of (2):

(i) converges almost surely to average consensus if

$$\lim_{k\to\infty} x_i(k) = x_{\text{av}} := \frac{1}{n_x}\sum_{i=1}^{n_x} x_i(0) \quad , \quad \forall_{i\in\{1,\ldots,n_x\}}$$

almost surely.

(ii) converges in expectation to average consensus if

$$\lim_{k\to\infty} \mathbb{E}[x_i(k)] = x_{\text{av}} \quad , \quad \forall_{i\in\{1,\ldots,n_x\}}.$$

(iii) converges in second moment to average consensus if

$$\lim_{k\to\infty} \mathbb{E}[(x_i(k) - x_{\text{av}})^2] \to 0 \quad , \quad \forall_{i\in\{1,\dots,n_x\}}.$$

Where $\mathbb{E}$ is the expected value operator. The next theorem proves asymptotic convergence as in Definition 4 and we delay the presentation of its finite-time property as a main result of this paper.

Theorem 2: Take the SVO-based consensus algorithm defined in this section. If the support graph of the matrix of probabilities W is strongly connected, then the algorithm converges in:

- *expectation*
- *mean square sense*
- *almost surely*.

□

Proof. The proof follows a similar reasoning as in [2]. We start by stacking each node own estimates $[z_i]_{2i-1}$ and $[z_i]_{2i}$ and prove the convergence of the whole system. Let us introduce variable $\mathbf{z}$:

$$\mathbf{z} = \begin{bmatrix} [z_1]_1 \\ [z_1]_2 \\ [z_2]_3 \\ [z_2]_4 \\ \vdots \\ [z_{n_x}]_{2n_x-1} \\ [z_{n_x}]_{2n_x} \end{bmatrix} \tag{17}$$

with $\mathbf{z} \in \mathbb{R}^{2n_x}$, where n_x is the number of nodes. Then, one can write

$$\mathbf{z}(k+1) = U_k \mathbf{z}(k) \tag{18}$$

where U_k is a matrix randomly selected from $\{Q_{ij}\}$, where the matrices Q_{ij} respect the given structure if we consider that each node has two states, given by

$$Q_{ij} = \left(\frac{1}{2}(e_i - e_j)(e_j - e_i)^\intercal + I_{n_x}\right) \otimes I_2 \tag{19}$$

for each pair of nodes i and j communicating with each other with probability w_{ij} gathered in the probability matrix W.

We start by proving convergence in expectation since convergence in mean square will be derived from this result. Let us define

$$R = \mathbb{E}[U_k].$$

Then

$$\mathbb{E}[\mathbf{z}(k+1)] = R\mathbb{E}[\mathbf{z}(k)]$$

due to the probability distributions w_{ij} being independent. By applying iteratively we get

$$\mathbb{E}[\mathbf{z}(k+1)] = R^k\mathbb{E}[\mathbf{z}(0)]$$

Rearranging our variables using the transformation $T^\intercal Q_{ij} T$ with

$$[T]_{ij} = \begin{cases} 1, & \text{if } j = 2i-1 \wedge i \leq n_x \\ 1, & \text{if } j = 2(i-n_x) \wedge i > n_x \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

we get

$$T^\intercal R T = I_2 \otimes \left((1 - \frac{1}{n_x})I_{n_x} + \frac{1}{n_x}W\right)$$

The eigenvalues of R are the eigenvalues of $(1 - \frac{1}{n_x})I_{n_x} + \frac{1}{n_x}W$ counted twice. We can use the fact that

$$\lambda((1 - \frac{1}{n_x})I_{n_x} + \frac{1}{n_x}W) = (1 - \frac{1}{n_x})I_{n_x} + \frac{1}{n_x}\lambda(W)$$

and since W is a doubly stochastic matrix with a strongly connected support graph with all but one eigenvalues less than 1. The $\lambda(W) = 1$ is associated to the eigenvector $\mathbf{1}_{n_x}$. Thus, $\lim_{k\to\infty} R^k = I_2 \otimes \mathbf{1}_{n_x}/n_x$ which proves the convergence in expectation with rate equal to $(1 - \frac{1}{n_x})I_{n_x} + \frac{1}{n_x}\lambda_2(W)$, where λ_2 is the second largest eigenvalue.

In order to prove convergence in the mean square sense, let us compute

$$\mathbb{E}[z(k+1)^\intercal z(k+1)] = R_2\mathbb{E}[z(k)^\intercal z(k)]$$

where $R_2 = R$ due to the fact that $Q_{ij}^{\intercal} Q_{ij} = Q_{ij}$. Therefore, using the same argument as for the convergence in expectation, the algorithm converges in the mean square sense with the same rate as the convergence in expectation. Almost surely convergence is given by using the fact that $\mathbb{E}[z(k+1)] = R^k \mathbb{E}[z(0)]$, which means that convergence is achieved at an exponential rate. Using the Borel-Cantelli first lemma [34], [35], the sequence converges almost surely. ■

The previous theorem shows the asymptotic convergence of the algorithm and found a close form for its convergence rates. The result is useful when characterizing its behavior in the presence of approximations, since we over-bounded the set $X_i(k)$ with a hyper-parallelepiped to reduce the amount of information that is communicated at each time instant. However, such a result only considers each node current state which is known and does not need to be estimated since it is measured every time instant. We defer to the next section a result of finite-time convergence that provides a faster convergence by exploring the SVO estimates and intersection during each communication.

VI. Theoretical overbound on the fault signal

An important issue regarding any fault detection method is the "maximum impact" of a fault in the system. The meaning of "maximum impact" depends on the specific application. Whereas in a physical system it makes sense to measure the energy of the fault signal being injected, in the case of consensus the maximum impact is given by the sum of the fault signal at each time instant. More generally, we can consider any function $f(u_k)$ where u_k stacks all the values of signal u until time instant k.

For the case of a physical system, function f takes the form

$$\frac{1}{N}\sum_{k=0}^{N} ||u(k)||^2 \tag{21}$$

whereas for the consensus case, f is a linear combinations of fault signal u of the form

$$\frac{1}{N}\sum_{k=0}^{N} u(k). \tag{22}$$

As an example, consider a 3-node network where all the nodes can communicate among them. Now take two fault signals for two time slots $u_1 = \mathbf{1}_2$ and $u_2 = 10^6 \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Signal u_1 has an energy equal to 1 while u_2 has 10^{12}. Using the energy of the signal as a metric, u_2 should have a higher impact on the system, even though its real impact on the final consensus value is zero, while for the signal u_1 shifts the true steady state in $2/3$.

A theoretical bound can be computed *a priori* using the SVO framework for the maximum impact of a fault. We start by looking at the worst possible attack that is not guaranteed to be detected. Let us borrow the definitions from [33]:

$$(A_N, b_N) = FM\left(\begin{bmatrix} M_N \\ -M_N \\ \tilde{M}_0 \\ \tilde{M}_W \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \tilde{m}_0 \\ \tilde{m}_W \end{bmatrix}, 2n_x \right) \tag{23}$$

where the FM stands for the Fourier-Motzkin elimination method [32] and:

$$\tilde{M}_0 = \begin{bmatrix} diag(M_0, M_0) & 0 \end{bmatrix}, \tilde{m}_0 = \begin{bmatrix} m_0 \\ m_0 \end{bmatrix},$$

$$\tilde{M}_W = \begin{bmatrix} 0 & diag(M_d, \cdots, M_d) \end{bmatrix},$$

$$\tilde{m}_W = \begin{bmatrix} m_d^{\intercal} & \cdots & m_d^{\intercal} \end{bmatrix}^{\intercal},$$

$$M_N = \left[\begin{array}{cc|c} C_A & -C_B & \\ C_A A_A & -C_B A_B & \bar{R} \\ \vdots & \vdots & \\ C_A A_A^N & -C_B A_B^N & \end{array}\right],$$

$$\bar{R} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ R_1^1 & 0 & \cdots & 0 \\ R_1^2 & R_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ R_1^N & R_2^N & \cdots & R_N^N \end{bmatrix},$$

$$R_i^k = \begin{bmatrix} C_A A_A^{k-i} B_A & -C_B A_B^{k-i} B_B \end{bmatrix}.$$

where M_d and m_d define the set of allowable realizations of u, i.e., M_d and m_d are defined such that $u(k) \in \text{Set}(M_d, m_d)$; and A_i, B_i, and C_i, with $i \in \{A, B\}$ are the matrices of the dynamics of two linear time-varying systems, as defined in (4) and further described in the sequel. With a slight abuse of notation, we write the product of N matrices $A(k)$ as $A^N = A(k)A(k-1)\cdots A(k-N+1)$ for shorter notation.

The aforementioned definitions characterize the set of admissable inputs that make both models have the same outputs. In the following proposition, a theoretical threshold γ_{min} for any function f of the input fault u is given. The value of γ_{min} defines the maximum impact of a fault that is not guaranteed to be detected.

Proposition 4 (Attacker signal bound): Let us consider a "fault-free" system:

$$S_A = \begin{cases} x_A(k+1) = A(k)x_A(k) \\ y_A(k) = C(k)x_A(k) \end{cases}$$

and a faulty system:

$$S_B = \begin{cases} x_B(k+1) = A(k)x_B(k) + B(k)u(k) \\ y_B(k) = C(k)x_B(k) \end{cases}$$

where $u \in \mathbb{R}^{n_u}$, $x_i \in \mathbb{R}^{n_x}$, $y_i \in \mathbb{R}^2$, initialized with the same initial conditions. Compute the pair (A_N, b_N), which is the set for all possible values of u of the last N time instants, defined as in (23).

Consider γ_{min} to be the theoretical threshold for the fault given as the result of the convex optimiazation

$$\gamma_{min} := \max_{A_N\xi \leq b_N} f(\xi),$$

where the vector ξ is a variable stacking all possible attacker signals u in the last N time instants and f be a generic function depending only on ξ. The fault is guaranteed to be detected if

$$f(u_k) > \gamma_{min}. \tag{24}$$

□

The result presented in Proposition 4 is a direct consequence of the definition of the set $\{\xi : A_N\xi \leq b_N\}$. The advantage of the representation in (24) is that the distinguishability problem is cast as an optimization or feasibility problem subject to linear constraints. Definition 1 has a clear connection to Proposition 4. The value of γ_{min} identifies detectable faults since any fault signal that is detectable will have an evaluation of function f higher than the theoretical γ_{min}. Proposition 4 was discussed in [9] for the quadratic norm function.

In the context of the two particular cases that we were describing, the threshold for the energy of the fault signal can be computed using

$$\gamma_{min} := \max_{A_N\xi \leq b_N} \xi^\intercal P \xi$$

with

$$P = \frac{1}{N} diag(0_{n_u}, I_{n_u}, \cdots, 0_{n_u}, I_{n_u})$$

and the maximum impact for the consensus case is given by

$$\gamma_{min} := \max_{A_N\xi \leq b_N} P_c\xi. \tag{25}$$

with

$$P_c = \frac{1}{N}[0_{n_u}^\intercal, \mathbf{1}_{n_u}^\intercal, \cdots, 0_{n_u}^\intercal, \mathbf{1}_{n_u}^\intercal].$$

In the case of consensus, if we define the true consensus value as x_{true}, using Proposition 4 with function (25) we get:

$$1_{n_x}x(k+N) - x_{\text{true}} = \frac{\gamma_{min}}{n_x} \tag{26}$$

The value of γ_{min} decreases as N increases, as more information is considered and, therefore, the longer the sequence of observations, the smaller impact an attacker can have on the final consensus value while avoiding detection. Thus, increasing the observation horizon decreases the impact of undetectable faults on the final consensus value. Since the algorithm introduced in Section V produces better estimates than the distributed individual detection using an SVO per node, it ensures a smaller effect of undetectable faults.

Proposition 4 defines a possible categorization of the undetectable faults using their impact on the final value of consensus. Nevertheless, calculating γ_{min} *a priori* to determine what value of N we should choose in order to meet a certain level of quality in the final consensus value, requires a combinatorial calculation. We recall that computing the set $\text{Set}(A_N, b_N)$ is

combinatorial both in the number of uncertainties and also in the horizon N. As an alternative, one can simply compute the set-valued estimates and at each time compute an overbound for γ_{min} as the summation of all the edges of the polytopic set. If no fault was detected, the maximum change in the states is given by the difference between the maximum of the estimate interval and its minimum.

Parameter γ_{min} is the smallest input before systems S_1 and S_2 are distinguishable in the sense that the measured output of the faulty system cannot be generated by the dynamics of the non-faulty one. As a consequence, we can use the same line-of-thought to derive the following result.

Corollary 1 (Attacker signal bound for SSVO): Consider a non-faulty system S_1 and a faulty system S_2 as in Proposition 4. Then, a fault is detectable in N measurements with a false alarm probability lower than or equal to α, if

$$f(u_k) > \gamma_{min}. \tag{27}$$

□

VII. Asymptotic correctness

In this section, a set of relevant results regarding the correctness of the SVOs, i.e., the SVOs ability to estimate without error the state of the system, are presented. These results allows us to have finite-time consensus even for the case where a node estimates are built using its own local measurements and without receiving estimates from its neighbors. In the next theorem, we show an important feature of the proposed algorithm, when applied to fault detection in networks, although its verification may be costly in terms of required computational power.

Before stating the theorems, take a 5-node network as an example, $n_x = 5$, where node 1 is running the SVO and has as neighbors nodes 2 and 3. Nodes 4 and 5 are neighbors of nodes 2 and 3. After some time, node 1 will determine exactly nodes 2 and 3 due to direct communication. However, since nodes 4 and 5 are both neighbors of nodes 2 and 3, even though the numeric value for the state of node 4 and 5 can be computed, node 1 cannot associate which numeric value corresponds to which node. The same reasoning allowed to discard edges of the communication graph in Proposition 3. Thus if the true state after some time is $x(k) = \begin{bmatrix}1 & 2 & 3 & 4 & 5\end{bmatrix}^\intercal$, then $X(k) = \{\begin{bmatrix}1 & 2 & 3 & 4 & 5\end{bmatrix}^\intercal, \begin{bmatrix}1 & 2 & 3 & 5 & 4\end{bmatrix}^\intercal\}$. However, the ordering of the nodes is irrelevant to consensus and the final value can be computed by averaging any of the points in $X(k)$.

Following the example let us define the set $X^\star := \{x : Px_{\text{true}}\}$, where P is a permutation matrix, i.e., P has exactly an entry equal to 1 in each row and column and all the remaining are equal to zero. We further restrict P as to have the rows associated with the node running the SVO and its neighbors equal to the respective row in the identity matrix (i.e., there are no permutations of the node running the SVO and its neighbors). Using $X^\star$, we can state the following theorem.

Theorem 3: Consider the fault detection described in Section III where an SVO estimates the state without sharing node measurements and a horizon $N^\star$. Take $X(N^\star)$ constructed using (8). Then,

$$\text{Prob}\left[X(N^\star) \to X^\star\right] \to 1 \text{ as } N^\star \to \infty$$

□

Proof. Let us rewrite the matrix in (8) recursively:

$$\underbrace{\begin{bmatrix} \mathcal{R}_1 & 0 & 0 & \cdots & 0 \\ 0 & \mathcal{R}_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \mathcal{R}_N & 0 \\ 0 & 0 & \cdots & 0 & M_0 \end{bmatrix}}_{\mathbf{M_{\Delta^\star}}} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \vdots \\ \mathbf{x_k} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ y(k+1) \\ -y(k+1) \\ 0 \\ 0 \\ \vdots \\ y(1) \\ -y(1) \\ -m_0 \end{bmatrix} \tag{28}$$

where

$$\mathcal{R}_n = \begin{bmatrix} I & -\mathcal{A}_n \\ -I & \mathcal{A}_n \\ C(k+1-(n-1)) & 0 \\ -C(k+1-(n-1)) & 0 \end{bmatrix}$$

and $\mathcal{A}_n$ represents the matrix $A_0 + A_{\Delta^\star}$ with a $\Delta^\star$ that accumulates the uncertainties for n periods of time, i.e., the parameter $\Delta^\star$ is the uncertainty instantiation for the respective horizon (see [33]).

Construct a sequence of time instants $\{c_k : 0 \leq c_k \leq N^\star\}$ as follows, with respect to node i running the SVO:

- There exists a communication between i and all of its first degree neighbors where only the state is transmitted and not the estimates, i.e., $\forall j : (i,j) \in E$ we have $A(k) = Q_{ij} \vee A(k) = Q_{ji}$;

- with each neighbor of i there exists a communication at time even and at time odd a communication between that neighbor and a second-degree neighbor and this pattern is repeated for the number of second-degree neighbors of each of our node neighbor, i.e., $\forall j : (i,j) \in E, \forall \ell : (j,\ell) \in E$ such that $A(2k) = Q_{ij} \vee A(2k) = Q_{ji}$ and $A(2k+1) = Q_{j\ell} \vee A(2k+1) = Q_{\ell j}$;
- repeat the same as before for the third-degree neighbors with communication between the nodes happening at each multiple of three communication instants. The number of communications must be equal to the number of possible paths with length 2.
- we continue with the same reasoning until all the nodes are included in the sequence.

Since when a node is involved in a communication there is no uncertainty, the sequence was constructed such that with the first condition all the neighbor states can be determined. With the second condition all the second-degree neighbor states can be determined. The same applies for any degree neighbors. This implies that for a specific instantiation of $\Delta^\star$, the system in (28) either:

- has only one solution;
- is infeasible.

Thus, the estimate set $X(k)$ is a union of at most card(Δ) points. $\forall \epsilon > 0, \exists N^\star$ such that the sequence exists with probability $1-\epsilon$ and the conclusion follows. ■

The previous result shows that SVOs have an intrinsic correctness property that can be used to compute the average consensus. Theorem 3 assumes that estimates are not shared between neighbors at the expenses of considering a large horizon $N^\star$. Nevertheless, in practice its applicability is questionable, as $N^\star$ can be arbitrarily large and represent a prohibitive computational burden. Since the SVO complexity grows exponentially with the horizon, one cannot use Theorem 3 to determine the states of each node in the network, in the general case. However, the result is interesting in the scenario where the node running the SVO is controlling the network and is allowed to impose a given communication pattern. In such cases, it can calculate a pattern ensuring the conditions of the theorem are fulfilled, guaranteeing finite-time consensus and detection of (detectable) faults in the sense of Definition 1. Progress is made in the next theorem to drop the horizon condition by taking advantage of state sharing between nodes.

Theorem 4: Consider the algorithm described in Section V and illustrated in Fig. 2 and $X(\tilde{N})$ constructed using (8). Then,

$$\text{Prob}\left[X(\tilde{N}) \to \{x_{\text{true}}\}\right] \to 1 \text{ as } \tilde{N} \to \infty$$

□

Proof. Construct the sequence of time instants $\{c_k : 0 \leq c_k \leq \tilde{N}\}$ that fulfills the following conditions

- every transmission shares one of the nodes involved in the previous transmission, i.e.,

$$\begin{aligned}\forall k \in &\{c_k\} : \\ &A(k) = Q_{ij}, A(k+1) = Q_{i\ell} \vee A(k+1) = Q_{\ell i}\end{aligned}$$

  for any node ℓ;
- there exists a time instant such that before and after that time, all the nodes in the network were involved in the communication, i.e.,

$$\begin{gathered}\exists k_c \forall i \exists k_i \leq k_c : (A(k_i) = Q_{i\ell} \vee A(k_i) = Q_{\ell i}) \\ \wedge \\ \exists k_i' \geq k_c : (A(k_i') = Q_{i\ell} \vee A(k_i') = Q_{\ell i})\end{gathered}$$

  for any node ℓ.

$\forall \epsilon > 0, \exists N^\star$ such that this sequence exists with probability $1-\epsilon$.

Define a function

$$V(k) = \text{card}(\tilde{z}_i(k))$$

where the function card(x) counts the number of non-zero entries of vector x, and i is a node involved in communication at time k. Function $V(k)$ counts, therefore, the number of uncertain states of the last node i involved in a communication at time k, and

$$\tilde{z}_i(k) = [z_i(k)]_{2i-1} - [z_i(k)]_{2i}.$$

Recall that, from equation (15), both nodes i and j involved in the communication have the same estimates of the states for all the nodes in the network.

Moreover, notice that

$$V(k+1) - V(k) \leq 0$$

for all time instants $k \leq k_c$, since every transmission is assumed to include one node involved in the previous communication and it is a strict inequality whenever it is the first time the node appear in a communication. In addition, the equilibrium

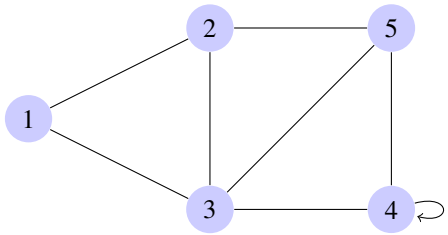

Fig. 3: Communication graph used for simulation.

points satisfy card$(\tilde{z}_i(k)) = 0, \forall i$ by construction, since they are the only points that, when computing the new set-valued state estimates, will return a set with only one point. Thus, for some time $k_c \geq 0$, $V(k_c) = 0$ using the two conditions of the sequence, which means that the two nodes communicating at time k_c have access to the full state of the network, regardless of the horizon of the SVOs. By the discrete version of the La Salle Principle, the conclusion follows.

Since, for every node ℓ, $\exists k_i' \geq k_c : A(k_i') = Q_{i\ell}$, the full state is passed to all the remaining nodes. We conclude that all nodes have $X(k)$ equal to a singleton. ■

Remark 2: Notice that, in practice, by implementing a token-passing scheme, the algorithm can be forced to converge in finite-time regardless of the chosen horizon, if no fault is detected.

The main point of the construction was that any two consecutive time instants share one of the nodes that communicated. However, caution is necessary to avoid reducing the algorithm to a deterministic setting. One possible solution is to consider that the token is passed randomly when communicating (i.e., with a probability p, the node sends the token when it communicates, and with probability $1-p$ the node retains the token). In addition, instead of nodes having equal probability $\frac{1}{n_x}$ of initiating a communication, the probability distribution is concentrated in the node that possesses the token. This means that there is a non-zero probability of a node starting a communication even though it does not possess the token.

The advantage of having a non-zero probability for any node to initiate a communication is to prevent an attacker from stopping the whole network by controlling the node that possesses the token. Mechanisms for fault robustness in a token-based gossip algorithm are outside the scope of this paper and also further work is needed to evaluate its effects on the convergence rate.

VIII. Simulation Results

In this section, we show simulation results for some meaningful scenarios which are used to illustrate specific features of the proposed fault detection schemes: deterministic, stochastic and consensus algorithm with fault detection. Two different types of faults are tested against the standard deterministic SVO when running in a single node. Comparison is also made to the case where each node runs a local SVO as to determine the first time of detection. A third type is detected by the SSVO, to motivate the use of the stochastic information, when a worst-case detection is not suitable. Lastly, the properties of the consensus algorithm are demonstrated, in particular its finite-time convergence.

The network used in the simulations has a small number of connections between the node running the estimates and the remaining node as to make the detection more challenging. The intuition is that the node running the estimates will not directly observe all the nodes making the detection harder. Without loss of generality, we illustrate the results from the perspective of node one with a faulty neighbor, i..e, the output $y(k)$ corresponds to the observations of one of the neightbors of the faulty node.

We consider a 5-node network with nodes labelled $i, i \in \{1,2,3,4,5\}$ and initial state $x_i(0) = i-1$ and a nominal bound for the state magnitude of $|x_i| \leq 5$. In order to reduce complexity and to study the properties of the algorithms in a disadvantageous setting, we considered $N = 1$, meaning that we only use the information from the previous iteration for the estimates. This is a worst-case scenario, as the algorithm only takes into account the dynamics of the system with one time step from the last estimate and discards prior observations and their propagation using multiple steps with the system dynamics. A missed detection is considered if the algorithm is not able to detect the fault within 300 observations. Each result presented corresponds to 1000 Monte-Carlo runs. For convenience, node 1 is the node that performs the detection and node 2 is the failing node, and no faults occur in the first 10 transmissions. Note that if a node sends a different value than its initial state from the start of the simulation, it can trivially do so without being detected since the network has no information about the initial state of that node. The following probability matrix is used:

$$W = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.25 & 0 & 0.25 \\ 0.5 & 0.25 & 0 & 0.125 & 0.125 \\ 0 & 0 & 0.125 & 0.25 & 0.625 \\ 0 & 0.25 & 0.125 & 0.625 & 0 \end{bmatrix}$$

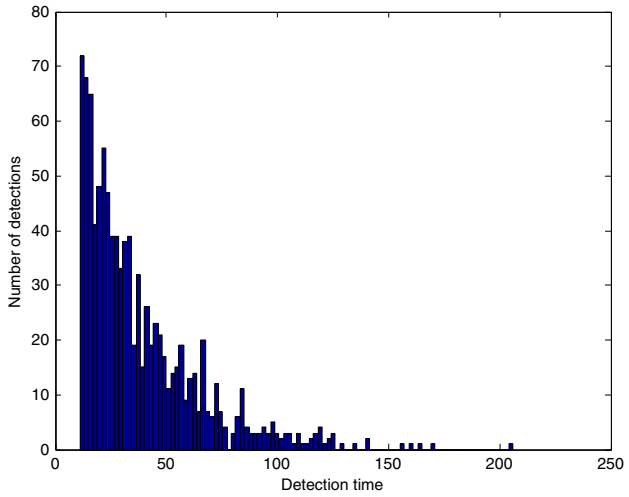

Fig. 4: Detection times for the stochastic fault.

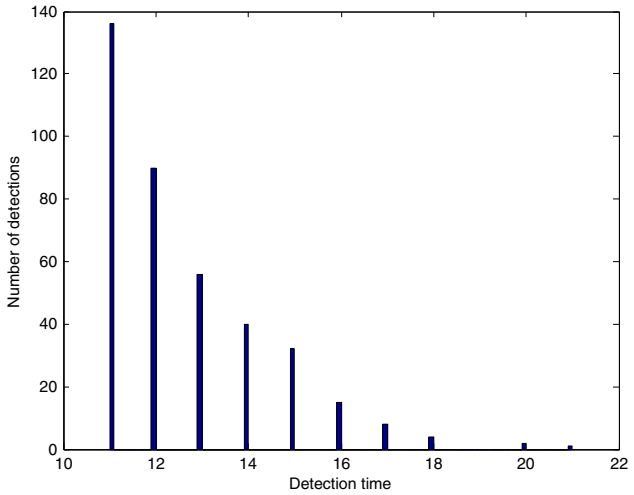

Fig. 5: Detection times for the deterministic fault.

The first scenario corresponds to an erratic node failure in which the node will respond with a random value. Specifically, after 10 iterations the node always replies as if its state was drawn uniformly from the interval of admissible states $[-5, 5]$.

Fig. 4 depicts the histogram of the detection times for the aforementioned fault. In this simulation, the detection rate was 100%, which is not surprising from the erratic behaviour of the node. Analysing the distribution, one key observation that is recurrent in other simulations is that, as times passes, the detection is more likely to occur. At the moment of detection, we have $\gamma_{min} = 56.25$ and the correspondent magnitude of the injected signal $\|u\|^2 = 4.405$. We concluded that the value of γ_{min} as a worst-case scenario is conservative in the sense that signals with a smaller energy are also detected.

We also considered a less erratic scenario where a node becomes unresponsive due to CPU load or software crash, does not perform the consensus update and, therefore, replies always with the same value.

Fig. 5 depicts the detection time for the deterministic fault where the node replies with the same value. In this case, the detection rate is 38.4%. In some sense, the lower detection rate is motivated by the fact that this fault does not change the state as much as the previous one. Since node 2 has other neighbors not in common with node 1, the fault is undetectable in more transmission sequences than in the previous simulation. Nonetheless, we still observe the behaviour that the fault is more likely to be detected as time progresses. Once again, we calculate $\gamma_{min} = 76.56$ and $\|u\|^2 = 2.997$ and observe that the injected signal is still detected even though its energy is less than the theoretical bound.

To illustrate the benefits of the SSVO when detecting faults, we consider a scenario where a node takes advantage of the network and initiates communication with a neighbour regardless of the probability matrix W, but does not change any of

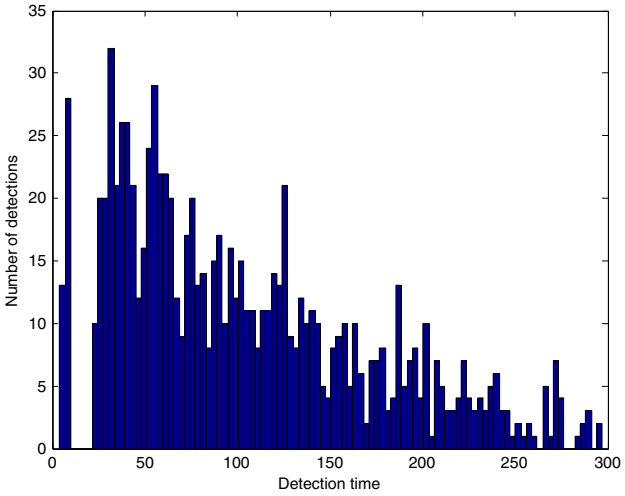

Fig. 6: Detection times for the SSVO.

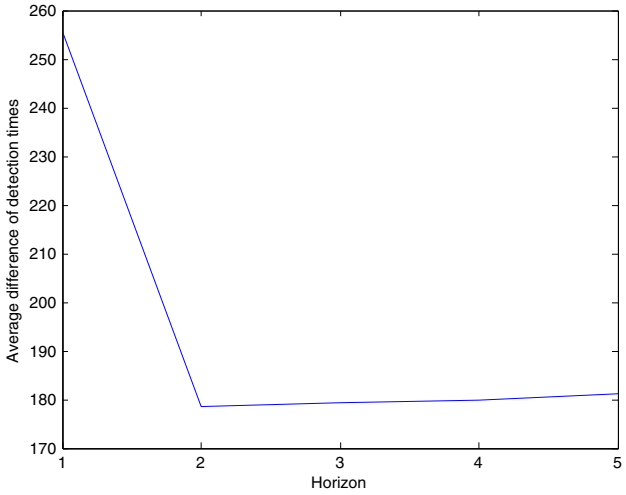

Fig. 7: Average difference between detecting with a SVO in one node or in all the nodes.

the nodes state. Notice that using an SVO, such faults would not be detected as any communication pattern that is possible is considered regardless of its probability. Between transmission time $10 < k < 20$, it is assumed that the communication takes place between node 3 and 4. Moreover, define $\alpha = 0.1$.

Fig. 6 depicts the detection times for the SSVO case with a detection rate of 92.8%. Even though the behaviour is still the same, we can no longer guarantee that the detection is caused by the fault and not by a communication pattern which we consider to be a fault, but that has non-zero probability of occurring in a healthy scenario.

In the previous simulation results, we depicted the detection time for a single node point of view in the network. However, when running the detection scheme presented in this paper, each node will run an SVO of their own to estimate the possible set of states and it is therefore important to assess the first time any node detects the fault. The simulation setup is the same as before and we assume that a node is trying to drive the consensus value by repeating the same value. Without a fault detection scheme, all the nodes would asymptotically reach a final consensus equal to the repeated constant. To make the results comparable, the data presented was generated using a thousand different seeds for the random number generator used to select the communication pairs, according to the probability matrix W.

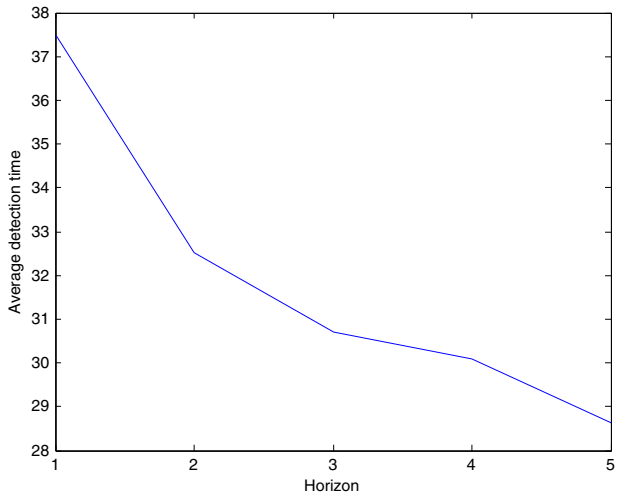

Fig. 8: Detection time for different horizon values for a fault constant equal to 3.

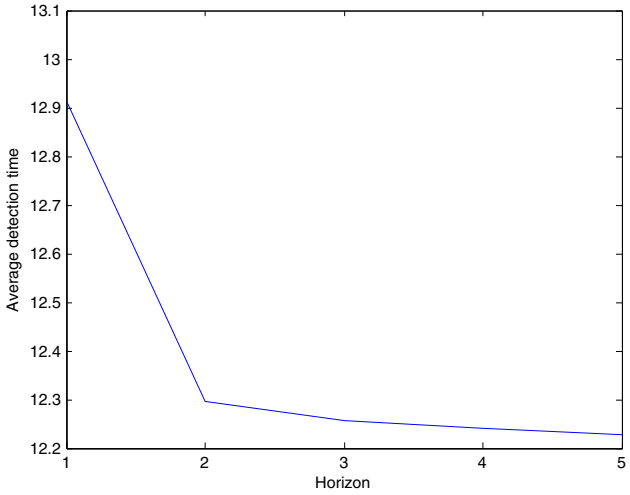

Fig. 9: Detection time for different horizon values for a fault constant equal to 4.9.

In Fig. 7, the average difference between the time that any node detects a fault and that node 1 detects the fault is presented. For a horizon equal to 1, we have a huge difference motivated by the fact that when considering just the detection from node 1, and using this faulty scenario, there is a remarkable number of undetected faults leading to considering the detection time as 300 time steps, which is the maximum length of the simulation. For the remaining values of the horizon, we have an increase in the detection time, which illustrates the importance of considering the different observations available to the nodes.

Another interesting issue is to determine the impact of changing the horizon in the detection time. By construction, incrementing the horizon leads to a smaller or equal time of detection. The rate at which the detection time varies is of particular interest when assessing the trade-off between fast detection and computational complexity.

In order to show the decreasing trend in the detection time as the horizon increases, we selected two fault constant values, namely 3 and 4.9. The intuition behind this choice is that a fault characterized by using a constant 4.9 is "easier" to detect, since the magnitude of the difference between the constant and the true state is larger than when considering a fault constant of 3. Fig. 8 and Fig. 9 show the mean detection time for different horizon values of having a fault constant equal to 3 and 4.9, respectively. When considering the case of constant 3, there is a faster decrease in the detection time which goes from over 37 time steps when the horizon is equal to 1, to under 29 when the horizon is equal to 5. For the case of constant 4.9, the difference is between using a horizon equal to 1 and higher horizons.

Emphasizing on the observed behavior, we present in Fig. 10 the mean detection time for different constant values. From Proposition 4, this phenomenon can be seen as the magnitude of the fault approaching γ_{min}, which is the worst case for the magnitude of the injected signal before being detectable in the worst case scenario. Depending on the specific application, the horizon can be selected so as to meet the specific requirements. In the example of consensus, the horizon can be selected in

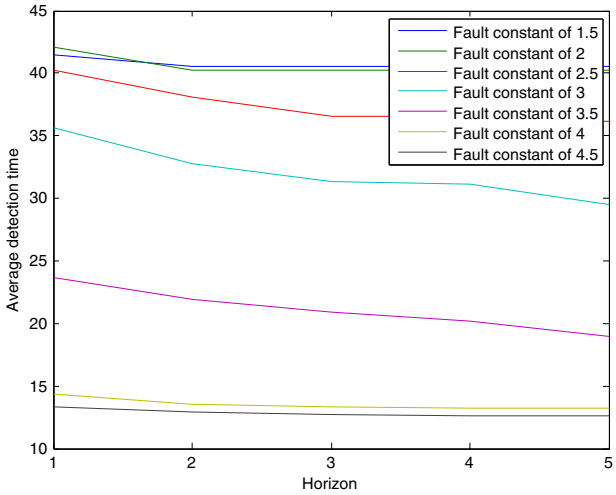

Fig. 10: Detection time for different fault constants.

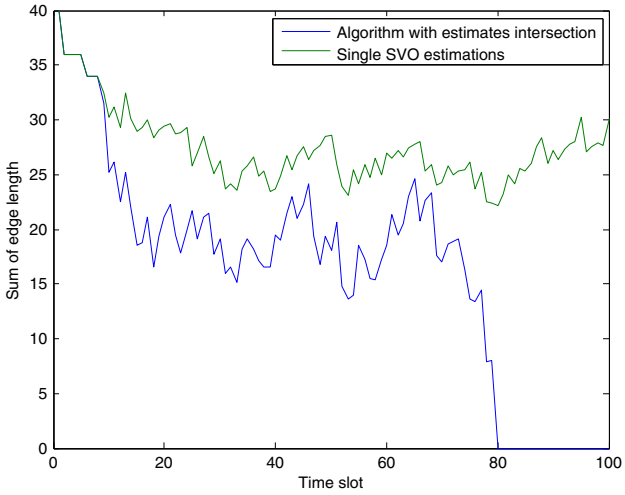

Fig. 11: Typical behavior of the size of the SVO.

order to decrease the expected deviation in the final consensus value, since by increasing the horizon, the maximum magnitude of the input signal decreases. In applications where the computation cost is not a problem, but there is a demanding criteria for the detection time, the horizon should be set as close to $N^\star$ as possible. However, for real-time applications, where the running time of the detection is crucial, a small horizon should be selected and the detection scheme becomes a best-effort approach.

We now present simulations that illustrate the finite-time consensus property derived in the previous section. Focus is given on how a measure of the set dimension evolves with the algorithm as opposed to a setting where nodes just run SVOs without sharing their estimates. The simulations also indicate how likely it is to find a sequence of transmissions that produce finite-time consensus when using randomized gossip algorithms.

Our experiment setting for the following tests does not include any fault and, at each time instant, we compute a measure of the size of the SVO. Computing the volume would be meaningless since at least the dimension corresponding to the node value has size zero, as the node has access to its own value at all time. Since the representation of the set of estimates is converted into a hyper-parallelepiped before being sent to a neighbor upon communication, we sum the length of uncertainty for each state and regard that measure as the size of the set. Each node has its own set-valued estimate, which we represented as a vector after bounding with a hyper-parallelepiped, as described in the previous section. For that reason, to measure the size of the SVOs across the network, we take the mean values (computed element-wise) of those vectors. By definition, if such measure reaches zero, then all nodes have reached consensus.

Fig. 11 depicts a typical run where finite-time consensus is achieved. All the simulations share the same behavior and what distinguishes them is the time where consensus is achieved for the algorithm. In comparison, the same measure is calculated

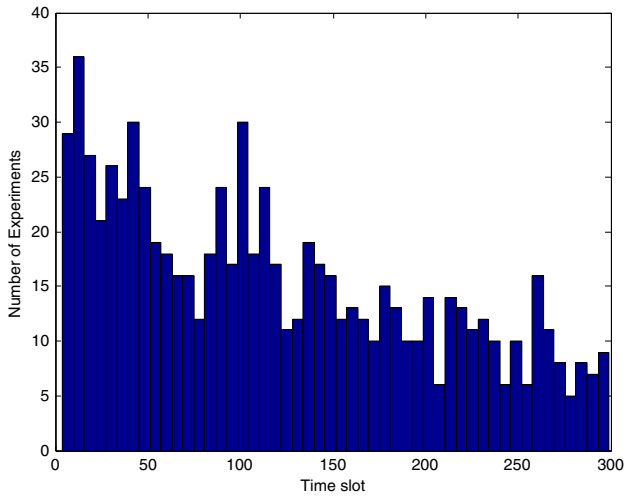

Fig. 12: Histogram for the stopping time with the proposed algorithm.

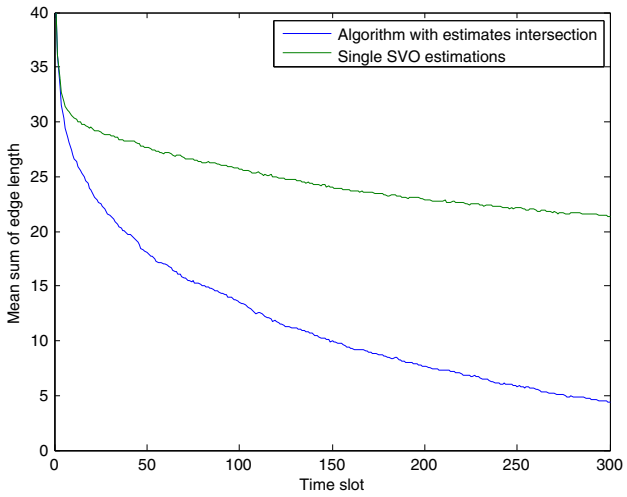

Fig. 13: Evolution of the mean sum of edges of all node set-valued state estimations.

for the case where each node runs its own independent SVO computed using only its own measurements. As expected, the estimates using the algorithm are less conservative as they incorporate the measurements performed by the node itself and the estimation set transmitted by its neighbors. In this particular run, consensus was achieved by all the nodes at iteration 80.

Using a 1000 Monte-Carlo run, in Fig. 12 is shown the histogram for the stopping time of the algorithm when using a horizon of 1. The experiments where consensus was not achieved in less than 300 communications are not represented in the histogram and corresponded to 21.9% of the cases. We then repeated the simulations for the same sequence of communications using a horizon of 5. The percentage of experiments that did not end in a finite-time consensus within the 300 time instants were 13.4%. The decrease is justified by the smaller sets that each node generates. In essence, to get 100% of the experiments to end in finite-time, we either have to increase the time of the simulation, increase the horizon, or both.

An important issue is the influence of the intersection step on the size of the set-valued state estimates. Fig. 13 depicts the mean of the sum of edges length for the 1000 Monte-Carlo runs for both the case of an SVO with and without estimate sharing using the intersection algorithm. Since the gossip random consensus algorithm is stable [2], the size of the generated set converges to a point (the consensus value) and the sum of edge lengths goes to zero asymptotically when in a non-faulty scenario and subject to a horizon smaller than $N^\star$. The measure of the sum of edges captures the size of the set-valued estimates, and correspondingly, how conservative they are. Fig. 13 shows that, in average, estimates are less conservative by exchanging set-valued estimates. Also, the set-valued state estimates, provided by the proposed algorithm, converge much faster to zero, since the conditions of the Theorem 4 are less restrictive.

IX. Conclusions

In this paper, the problem of fault detection in randomized gossip algorithms is addressed using the concept of SVOs. The introduction of the stochastic information to build the set is one of the main contributions of this paper that allows to detect faults based on the probability of that event. Two functions to measure the maximum attacker input signal for an undetectable fault are presented. The quadratic function is suitable for systems where the energy plays an important role whereas the linear function is characteristic of problems such as the consensus, where inputting a positive signal cancels the effect of a fault injecting a negative signal. We also showed the necessary number of past observations for the case of local information and when keeping the best value for the horizon is computational intractable.

Building on the results of having an SVO for fault detection, without sharing state estimates, SVOs in the absence of faults are capable of determining average consensus in finite-time using only measurements available to the node, but may require a large computational burden. The result is suitable to situations where one node is able to control/command the sequence of communications in the network.

In order to drop the requirement of a large horizon, an algorithm is presented where each node computes its own set-valued state estimates and performs an intersection with state estimates received by the neighbors. Besides reducing the computational burden, this method also achieves finite-time average consensus for any horizon value, provided that the algorithm runs for sufficiently large number of observations, and each node computes less conservative set-valued estimates. The result is relevant in practice to determine a stopping time in a faulty environment, which is not a straightforward issue due to the iterative nature and uncertainty generated by the random choice of communicating nodes. If conditions for finite-time convergence are not met within the time that the algorithm is running, asymptotic convergence of the state of the nodes is also provided.

We envisage as directions of future work, the study of additional properties of specific classes of algorithms. In particular, structural premises that allow to eliminate certain sequences of matrices $A(k)$ which are irrelevant for the computation of the SVO. In essence, associated with the results presented in this paper, such a mechanism would decrease the complexity even further and broaden the spectrum of application of the proposed fault detection method. Another line of possible research would be to integrate the SVO in a fault isolation scheme as to progress towards a fault correction scheme where the nodes would, after detecting a fault, isolate the faulty node and correct the state of the algorithm to a value closer to the true state if there was no fault. Such a goal poses very interesting research problems.

References

[1] R. Motwani and P. Raghavan, *Algorithms and Theory of Computation Handbook*, M. J. Atallah and M. Blanton, Eds. Chapman & Hall/CRC, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1882757.1882769

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508 – 2530, Jun. 2006.

[3] Y. E. Ioannidis and Y. Kang, "Randomized algorithms for optimizing large join queries," *SIGMOD Rec.*, vol. 19, no. 2, pp. 312–321, May 1990. [Online]. Available: http://doi.acm.org/10.1145/93605.98740

[4] M. Dietzfelbinger, T. Hagerup, J. Katajainen, and M. Penttonen, "A reliable randomized algorithm for the closest-pair problem," *Journal of Algorithms*, vol. 25, no. 1, pp. 19 – 51, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0196677497908737

[5] K. Mulmuley, *Computational Geometry: An Introduction through Randomized Algorithms*. NJ: Prentice-Hall, 1994.

[6] K. P. Kihlstrom, L. E. Moser, and P. M. Melliar-Smith, "Solving consensus in a byzantine environment using an unreliable fault detector," in *Proceedings of the International Conference on Principles of Distributed Systems (OPODIS)*, 1997, pp. 61–75.

[7] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 90 –104, jan. 2012.

[8] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *Automatic Control, IEEE Transactions on*, vol. 56, no. 7, pp. 1495–1508, July 2011.

[9] D. Silvestre, P. Rosa, R. Cunha, J. P. Hespanha, and C. Silvestre, "Gossip average consensus in a byzantine environment using stochastic set-valued observers," in *Decision and Control, 2013. CDC 2013. 52nd IEEE Conference on*, 2013.

[10] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre, *Finite-time Average Consensus in a Byzantine Environment Using Set-Valued Observers*. American Control Conference, 2014.

[11] H. Witsenhausen, "Sets of possible states of linear systems given perturbed observations," *Automatic Control, IEEE Transactions on*, vol. 13, no. 5, pp. 556 – 558, oct 1968.

[12] F. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *Automatic Control, IEEE Transactions on*, vol. 13, no. 1, pp. 22 – 28, feb 1968.

[13] F. Schweppe., *Uncertain Dynamic Systems*. Prentice-Hall, 1973.

[14] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: An overview," *Automatica*, vol. 27, no. 6, pp. 997 – 1009, 1991.

[15] D. Bertsekas and I. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *Automatic Control, IEEE Transactions on*, vol. 16, no. 2, pp. 117 – 128, apr 1971.

[16] C. Combastel, "A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, dec. 2005, pp. 7228 – 7234.

[17] T. Alamo, J. Bravo, and E. Camacho, "Guaranteed state estimation by zonotopes," *Automatica*, vol. 41, no. 6, pp. 1035 – 1043, 2005.

[18] R. Renesse, Y. Minsky, and M. Hayden, "A gossip-style failure detection service," in *Middleware'98*, N. Davies, S. Jochen, and K. Raymond, Eds. Springer London, 1998, pp. 55–70. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-1283-9_4

[19] R. J. Patton, "Fault-tolerant control systems: The 1997 situation," in *IFAC symposium on fault detection supervision and safety for technical processes*, vol. 3, 1997.

[20] J. Bokor and Z. Szabó, "Fault detection and isolation in nonlinear systems," in *Annual Reviews in Control 33.2*, 2009, pp. 113–123.

[21] P. Rosa, C. Silvestre, J. Shamma, and M. Athans, "Fault detection and isolation of LTV systems using set-valued observers," in *Proceedings of the 49th IEEE Conference on Decision and Control*, December 2010, pp. 768–773.

[22] P. Rosa and C. Silvestre, "Fault detection and isolation of {LPV} systems using set-valued observers: An application to a fixed-wing aircraft," *Control Engineering Practice*, vol. 21, no. 3, pp. 242 – 252, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066112002262
[23] S. Ranganathan, A. George, R. Todd, and M. Chidester, "Gossip-style failure detection and distributed consensus for scalable heterogeneous clusters," *Cluster Computing*, vol. 4, no. 3, pp. 197–209, 2001. [Online]. Available: http://dx.doi.org/10.1023/A%3A1011494323443
[24] R. Rajagopal, X. Nguyen, S. Ergen, and P. Varaiya, "Distributed online simultaneous fault detection for multiple sensors," in *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, April 2008, pp. 133–144.
[25] P. Menon and C. Edwards, "Robust fault estimation using relative information in linear multi-agent networks," *Automatic Control, IEEE Transactions on*, vol. 59, no. 2, pp. 477–482, Feb 2014.
[26] Z. Zhang and I. M. Jaimoukha, "On-line fault detection and isolation for linear discrete-time uncertain systems," *Automatica*, vol. 50, no. 2, pp. 513 – 518, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109813005207
[27] A. Cristofaro and T. A. Johansen, "Fault tolerant control allocation using unknown input observers," *Automatica*, vol. 50, no. 7, pp. 1891 – 1897, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109814001824
[28] M. Grewal and K. Glover, "Identifiability of linear and nonlinear dynamical systems," *IEEE Trans. on Automatic Control*, vol. 21, no. 6, pp. 833–837, 1976.
[29] P. Rosa and C. Silvestre, "On the distinguishability of discrete linear time-invariant dynamic systems," in *Proceedings of the 50th IEEE Conference on Decision and Control*, December 2011.
[30] J. Telgen, "Minimal representation of convex polyhedral sets," *Journal of Optimization Theory and Applications*, vol. 38, no. 1, pp. 1–24, 1982. [Online]. Available: http://dx.doi.org/10.1007/BF00934319
[31] J. Shamma and K.-Y. Tu, "Set-valued observers and optimal disturbance rejection," *Automatic Control, IEEE Transactions on*, vol. 44, no. 2, pp. 253 –264, feb 1999.
[32] S. Keerthi and E. Gilbert, "Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints," *Automatic Control, IEEE Transactions on*, vol. 32, no. 5, pp. 432 – 435, may 1987.
[33] P. Rosa, "Multiple-model adaptive control Multiple-Model Adaptive Control of Uncertain LPV Systems," Ph.D. dissertation, Technical University of Lisbon, 2011.
[34] E. Borel, "Les probabilités dénombrables et leurs applications arithmetiques," *Rend. Circ. Mat. Palermo (2)*, vol. 27, pp. pp. 247–271, 1909.
[35] F. P. Cantelli, "Sulla probabilità come limite della frequenza," *Atti Accad. Naz. Lincei*, vol. 26:1, pp. pp. 39–45, 1917'.