Triangular Networks for Resilient Formations

David Saldaña, Amanda Prorok, Mario F. M. Campos, Vijay Kumar

Abstract Consensus algorithms allow multiple robots to achieve agreement on estimates of variables in a distributed manner, hereby coordinating the robots as a team, and enabling applications such as formation control and cooperative area coverage. These algorithms achieve agreement by relying only on local, nearest-neighbor communication. The problem with distributed consensus, however, is that a single malicious or faulty robot can control and manipulate the whole network. The objective of this paper is to propose a formation topology that is resilient to one malicious node, and that satisfies two important properties for distributed systems: *(i)* it can be constructed incrementally by adding one node at a time in such a way that the conditions for attachment can be computed locally, and *(ii)* its robustness can be verified through a distributed method by using only neighborhood-based information. Our topology is characterized by triangular robust graphs, consists of a modular structure, is fully scalable, and is well suited for applications of large-scale networks. We describe how our proposed topology can be used to deploy networks of robots. Results show how triangular robust networks guarantee asymptotic consensus in the face of a malicious agent.

1 Introduction

Coordination of distributed autonomous systems with nearest neighbor communication has been widely used in swarms and multi-robot systems [1, 2, 3]. A majority of these applications build on information diffusion (or consensus) algorithms to synchronize the network with respect to a specific variable. However, such systems rely on the fact that all robots in the network are reliable, and contribute only legit-

David Saldaña, Amanda Prorok, Vijay Kumar
University of Pennsylvania, e-mail: {dsaldana, prorok, kumar}@seas.penn.edu

Mario F. M. Campos
Universidade Federal de Minas Gerais, e-mail: mario@dcc.ufmg.br

imate information. As a consequence, the systems are susceptible to failure when one or several robots are non-cooperative and share wrong information. This situation can be due to malicious attacks (e.g., a malicious outsider trying to manipulate the whole network) or due to platform-level faults (e.g., a robot sharing an incorrect location due to a defective GPS sensor). Hence, the question of network resilience is of utmost importance. In this work, we focus on *topological properties* that guarantee resilience to faults and attacks on individual nodes. We build on a distributed consensus algorithm, termed Weighted-Mean Subsequence-Reduced (W-MSR) algorithm [4], which guarantees resilience if certain topological requirements are met. Throughout this paper, we use the terms node and agent to refer to a robot in a network.

1.1 Related work

The topic of robustness has received considerable attention, in particular in the domain of complex networks [5, 6]. A main result of this body of work states that robustness can be achieved through sufficiently high connectivity: if the connectivity of the network is $2F$ or less, then F malicious nodes can prevent some of the nodes from receiving legitimate information from other nodes in the network. Conversely, when the network connectivity is $2F + 1$ or higher, there are various algorithms that enable a reliable diffusion of information [7, 8]. However, these algorithms not only depend on high connectivity, but also require non-local information in order to compute updates. As a consequence, Zhang et al., and later LeBlanc et al. [9, 10] introduced an alternative definition of network robustness that can deal with purely local update rules, and that provides resilient asymptotic consensus. The strength of the proposed approach is that it can deal with an arbitrary number F of malicious agents, the identities of which are unknown to the normal nodes in the network. However, in order to build the required topologies, the method assumes that any node can be connected to any other node in the network (i.e., in absence of sophisticated node placement algorithms, the networks must provide full connectivity). Hence, it remains unclear how the approach is practically implemented in networks where nodes have constrained communication radii. Also, it is noteworthy that the prior approach mainly deals with the problem of distributed estimation, and it remains to be explored how it applies to problems that require robust control of shapes and distributions, such as for cooperative exploration and coordination tasks [11, 12]. Finally, the problem of robustness has also been considered in mobile robot systems [13, 14, 15], with a particular focus on rendez-vous (i.e., the application of consensus algorithms to induce a gathering of robots in n-dimensional space). Most similarly to our work, the work in [14] considers the presence of non-compliant robots, and develops a solution that controls a team of robots so that rendez-vous can be achieved robustly. However, since their systems are mobile, they assume that communication radii can be adjusted over time (to ensure connectivity). As a consequence, their method requires a relatively large local neighborhood size:

for a single malicious robot, the neighborhood size must be at least 5 (cf. assumption 5.1 in their paper).

1.2 Contribution

The main contribution of our work is the definition of a network topology that can be constructed in a distributed and fully scalable manner, and that guarantees resilient asymptotic consensus in the face of malicious agents. In addition, our method is also able to deal with networked robots that have fixed communication radii. We consider the special case when a network is susceptible to a single fault or malicious node, and we identify a particular class of networks (which we term *triangular robust graphs*) for this case. We derive proofs of all our claims.

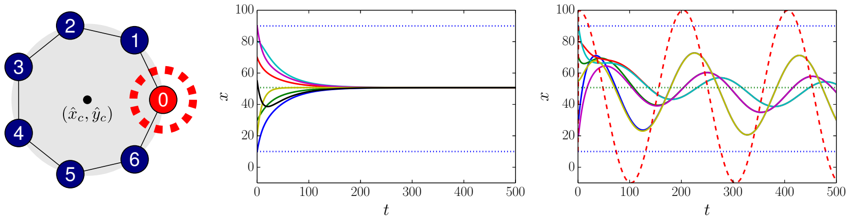
2 Preliminaries

Consider a network composed of a set of nodes $\mathcal{V} = \{1, 2, ..., n\}$ that are represented by points in a planar space $v_i \in \mathbb{R}^2$, for all $i \in \mathcal{V}$. Every node is equipped with a communication module that allows it to communicate with other nodes. The ability to communicate with adjacent nodes defines the set of connections $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Therefore, we model the network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The neighbors of node i are $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$. For a node subset $S \subset \mathcal{V}$, we denote its complement by $\bar{S} = \{i | \, i \notin S, i \in \mathcal{V}\}$.

2.1 Consensus

In distributed networked systems, each node is an autonomous entity that can adapt to changing conditions based on incoming data streams originating from neighboring nodes. As there is no central master, the nodes need to find an agreement with respect to the shared information in order to make unified decisions. The question of how to do this is solved by *consensus algorithms* [2, 1, 16, 17]. When performing a consensus algorithm, each node i has a variable of interest x_i, e.g., that describes the locations of the nodes, or that measures local temperatures. Subsequently, the whole network may want to estimate a global variable, such as the centroid of the network, or the average temperature of the environment, respectively, based on the distributed information available to the network as a whole. This goal can be achieved by local interaction, where each node i updates its own value at time-step t based on an update function:

$$x_i[t+1] = f(x_i[t], \{x_j[t] | j \in \mathcal{N}_i\}). \qquad (1)$$



(a) (b) (c)

Fig. 1 Consensus in a circular formation with seven robots. Panel (a) shows the network topology. Panel (b) shows a successful consensus for the variable x_c, where all nodes perform the same averaging update rule. Panel (c) shows an unsuccessful consensus that is manipulated by node 0, who shares an oscillatory value to avoid convergence for the non-malicious robots.

In [1], the authors show that, given a connected and balanced graph $\mathcal{G}$, every node $i \in \mathcal{V}$ reaches consensus on the average of the initial values, $x_i[t] \to \bar{x}[0] = \frac{1}{n}\sum_{i\in\mathcal{V}} x_i[0]$, when $t \to \infty$ by exchanging messages with the local neighborhood and applying an averaging function $x_i[t+1] = \frac{1}{|\mathcal{N}_i|+1}(x_i[t] + \sum_{j\in\mathcal{N}_i} x_j[t])$. We illustrate an example of consensus in Figure 1 for a circular formation in a *biconnected* ring topology (Figure 1a). In this context, each robot i moves to the location (x_i, y_i), given by $x_i = \hat{x}_c + \rho\cos(i\ 2\pi/n)$ and $y_i = \hat{y}_c + \rho\sin(i\ 2\pi/n)$, where $(\hat{x}_c, \hat{y}_c)$ is the center of the circle, estimated by consensus, and $\rho > 0$ is the known radius of the circle. Figure 1b shows how the robots achieve consensus on the average of the variable $\hat{x}_c$ (consensus for the variable $\hat{y}_c$ is analogous). This kind of robotic system works scales well when every node is functional and trustworthy. However, when a malicious node [1] stops adhering to the consensus update rule due to a hardware failure or a malicious attack, that robot can affect the behavior of all other robots in the network. As we observe in Figure 1c, when robot 0 starts to share an arbitrary oscillatory value, the values of all robots in the network are affected and consensus is not achieved. We see that a single malicious agent can hinder convergence and manipulate the whole network. For this reason, it is desirable to devise a resilient strategy. The problem of consensus in the presence of malicious nodes (such as robot 0 seen above) can be solved by deploying a *resilient consensus*s algorithm on all nodes. The following section introduces this method.

2.2 *Resilient Consensus*

A robust network is defined as a network that can reach consensus, even in the presence of F malicious nodes. Neither the identity nor the strategy of the malicious nodes is known. A known method that achieves consensus by converging

[1] We consider an attack or a failure as the same case, where a node shares a value that does not adhere to the consensus update rule. We call this kind of node a *malicious node*.

to a weighted average is the *Weighted Mean-Subsequence-Reduced (W-MSR)* algorithm [10, 4, 18]. Yet, for this method to work in the presence of malicious nodes, the network must satisfy certain topological conditions, which we detail below.

The W-MSR algorithm consists of three steps, executed at time t. First, node i creates a sorted list, from smallest to largest, with the received values from its neighbors $\mathcal{N}_i$. Second, the list is compared to $x_i[t]$, and if there are more than F values that are larger than $x_i[t]$, the F largest values are removed. The same removal process is applied to the smaller values. The remaining values in the list are denoted by $\mathcal{R}_i[t]$. Third, node i updates its value with the following rule:

$$x_i[t+1] = w_{ii}[t]x_i[t] + \sum_{j \in \mathcal{R}_i[t]} w_{ij}[t]x_j[t], \tag{2}$$

where $w_{ij} > 0$, and $\sum_j w_{ij}[t] = 1$. In the remainder of this paper, we consider all weights $w_{ij} = 1/(|\mathcal{R}_i[t]| + 1)$. An extended explanation of this algorithm is given in [10]. Using this algorithm, it is possible to achieve asymptotic consensus in a network with at most F malicious nodes, if the communication graph $\mathcal{G}$ is $(F+1, F+1)$*-robust*.

Definition 1. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *(r,s)-robust*, with constants $r \in \mathbb{Z}_{\geq 0}$, and $0 \leq s \leq |\mathcal{V}|$, if for any pair of disjoint subsets $S_1, S_2 \subset \mathcal{V}$, at least one of the following conditions is satisfied:

1. $|\mathcal{X}^r_{S_1}| = |S_1|$;
2. $|\mathcal{X}^r_{S_2}| = |S_2|$;
3. $|\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \geq s$,

where the r-reachable set $\mathcal{X}^r_{S_k} = \{i \in S_k \,|\, \mathcal{N}_i \backslash S_k \geq r\}$, $k \in \{1, 2\}$ is composed of the nodes in S_k with at least r neighbors outside S_k.

Based on this definition of the communication topology, LeBlanc et al. [10] stated the following theorem, which specifies the conditions for asymptotic consensus in presence of F malicious agents.

Theorem 1 (Th. 1, [10]). *Consider a network modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each normal node updates its value based on the W-MSR algorithm with an upper bound of F malicious agents. Then, resilient asymptotic consensus is achieved if the $\mathcal{G}$ is $(F+1, F+1)$-robust.*

Although these recent works provide a rigorous study of the topological characteristics that are necessary to provide resilience against a number of malicious agents [14, 10], they do not consider the physical constraints that real-world systems often have, such as limited or non-adjustable communication radii. Hence, it is not clear how the methods are applicable in real settings, and it is still an open question if their implementations are suitable to distributed actuator/sensor networks.

2.3 Biconnected Graphs

In the following, we describe a concept that is relevant to the derivations in the subsequent sections.

Definition 2. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *biconnected* if it stays connected after removing any vertex $i \in \mathcal{V}$.

An example of a biconected graph in shown in Figure 1a. Biconnected graphs have two special properties. First, they have two disjoint paths (no common vertices) between every pair of vertices (Theorem 2), and second, they can be extended/grown by adding nodes iteratively, as described by the expansion lemma below (Lemma 1).

Theorem 2 (Th. 4.2.1, [19]). *If a graph $\mathcal{G}$=($\mathcal{V}$, $\mathcal{E}$), with at least three nodes, is biconnected, then there exist at least two disjoint paths between any pair of nodes $i, j \in \mathcal{V}$.*

Lemma 1 (L. 4.2.2, [19]). *If $\mathcal{G}$ is a biconnected graph, and $\mathcal{G}'$ is obtained from $\mathcal{G}$ by adding a new vertex k adjacent to at least two vertices of $\mathcal{G}$, then $\mathcal{G}'$ is biconnected.*

3 Triangular Robust Graphs

The creation of robust and (r, s)-robust networks is challenged by three main items: *(i)* high connectivity is required; *(ii)* verifying if a graph is (r, s)-robust is an NP-hard problem [20]; *(iii)* default algorithms for creating robust networks are not designed for physically embedded systems, which potentially have hard constraints on edge lengths. In this section, we propose a particular network topology, termed *triangular robust graph*, that is resilient to one malicious node (i.e., $F = 1$) whose identity is unknown to the rest of the nodes. Our proposed network topology has the following convenient properties: it is $(2, 2)$-robust; it is incrementally expandable; and it can be verified in polynomial time and in a distributed manner. Furthermore, due to the inherent geometric properties of triangles, the topology is well suited to networked robotic systems with agents that have homogeneous, constrained communication radii. As a consequence, triangular robust graphs are well-suited to distributed robotic systems.

Definition 3. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *triangular robust* if it has at least three nodes and satisfies the following conditions:

1. The graph $\mathcal{G}$ is *biconnected*.
2. The neighbors of node i form a connected sub-graph, $\mathcal{G}_i = (\mathcal{N}_i, \mathcal{E})$, for all $i \in \mathcal{V}$.

Property 1. Based on Condition 1, the minimum degree of any node is two, $\text{Deg}(i) \geq 2$ for all $i \in \mathcal{V}$.

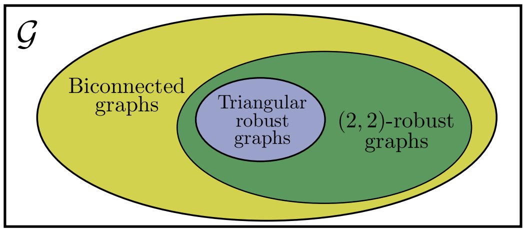


Fig. 2 In the whole set of connected graphs, this Venn diagram represents the set of the triangular robust graphs and its relationship with biconnected graphs and $(2,2)$-robust graphs.

In Figure 2, we show a Venn diagram that represents in the whole set of connected graphs, our proposed topology is a subset of the Biconnected graphs and also a subset of the $(2,2)$-robust graphs. We highlight that not all the $(2,2)$-robust graphs have the properties that we describe along this section.

Theorem 3. *Consider a network modeled by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. If $\mathcal{G}$ is triangular robust, then it is (2,2)-robust.*

Proof. We show that the triangular robust graph $\mathcal{G}$ is *(2,2)-robust*, as it satisfies the conditions of Definition 1. Given any pair of disjoint subsets $S_1, S_2 \subset \mathcal{V}$, by Definition 3, and Theorem 2, the graph $\mathcal{G}$ is *biconnected* and there exist two disjoint paths between any source node $s \in S_1$ and any target node $g \in S_2$. It implies that there are also two edges, $e_1 = (i, j)$ and $e_2 = (o, p)$, for each path respectively, such that $i, o \in S_1$ and $j, p \in \bar{S}_1$. Figure 3 illustrates both paths and their intermediate edges e_1 and e_2. The conditions of robustness are satisfied by checking the neighbors of j. There are only the following two cases:

1. If the node j does not have any neighbors outside S_1, *i.e.* $|\mathcal{N}_j \cap \bar{S}_1| = 0$. It implies that the node j is the target node, $j = p = g$, and $S_2 = \{j\}$. Then, the condition of robustness $|\mathcal{X}^r_{S_2}| = |S_2| = 1$ is satisfied for any S_1 (Definition 1, Condition 2). Figure 4a illustrates this case.
2. Otherwise, the node j has at least one neighbor outside S_1, i.e., $|\mathcal{N}_j \cap \bar{S}_1| \geq 1$. By Definition 3, Condition 2, since the neighbors of j are connected, there exists an edge $(k, l) \in \mathcal{E}$ such that $k \in \mathcal{N}_j \cap S_1$, and $l \in \mathcal{N}_j \cap \bar{S}_1$. Then the node k has two neighbors (j and l) outside S_1, and the 2-reachable set of S_1 contains at least one element $|\mathcal{X}^2_{S_1}| \geq 1$. Figure 4b illustrates this case.
   Applying the same sequence of statements for S_2, there exist at least one node with two neighbors outside S_2, then $|\mathcal{X}^2_{S_2}| \geq 1$. In this way, we show that the condition of robustness $|\mathcal{X}^2_{S_1}| + |\mathcal{X}^2_{S_2}| \geq 2$ is satisfied.

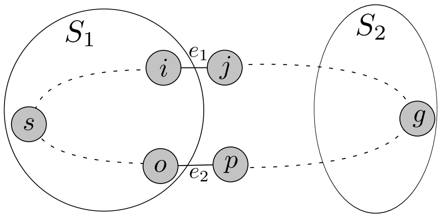
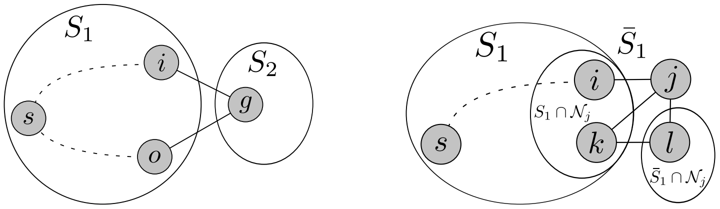


Fig. 3 A graph with two partitions S_1 and S_2 and two disjoint paths between the nodes $s \in S_1$ and $g \in S_2$.



(a) $|\mathcal{N}_j \cap S_1| = 0$. Node j is g.

(b) $|\mathcal{N}_j \cap \bar{S}_1| \geq 1$.

Fig. 4 Possible cases based on the neighbors of node j.

With the two cases above, we show that the conditions of robustness are satisfied for any pair of sets in a triangular robust network. □

Remark 1. The converse of the Theorem 3 does not hold, i.e., not all $(2, 2)$-robust graphs are triangular robust (see Figure 6 in [10]).

3.1 Determining Triangular Robustness

Checking if a graph is *(r,s)-robust* is an NP-hard problem [20], but we can check if a graph is triangular robust in polynomial time, based on the two conditions of Definition 3.

Condition 1: A simple centralized algorithm to check if $\mathcal{G}$ is *biconnected* can also be done in polynomial time. For each node $i \in \mathcal{N}$, we check if the graph maintains connectivity after removing i. It is possible to check connectivity based on *breadth-first-search* (BFS) with time complexity $O(|\mathcal{V}|+|\mathcal{E}|)$. Then, the result of checking connectivity $|\mathcal{V}|$ times is $O(|\mathcal{V}|^2 + |\mathcal{V}||\mathcal{E}|)$. It is also possible to check it in distributed way by using the algorithm presented in [21].

Condition 2: Checking if the neighbors are connected can be compute in linear time and in a distributed way. First, every node $i \in \mathcal{V}$ only needs to know the neighbors of its neighbors to create the sub-graph $\mathcal{G}_i = (\mathcal{N}_i, \mathcal{E})$. The worse case is the complete graph, for which the time complexity to check connectivity for the neighbors of node i is $O(|\mathcal{V}| + |\mathcal{E}|)$. For the complete network it is $O(|\mathcal{V}|^2 + |\mathcal{V}||\mathcal{E}|)$.

Since Conditions 1 and 2 are checked independently, we conclude that it is possible to check triangular robustness in polinomial time $O(|\mathcal{V}|^2 + |\mathcal{V}||\mathcal{E}|)$.

3.2 Inductive Construction

In the following, we show a simple method that expands a $(2,2)$-robust network, starting from an initial network configuration. Our starting point is a strongly connected graph with three nodes, which is the most basic form of a triangular robust graph. This configuration is then extended by adding one node at a time, while maintaining the $(2,2)$-robust property at all times, by ensuring that each new node is connected to two or more nodes (that are also connected among themselves). The following theorem shows that a triangular robust graph is expandable.

Theorem 4. *If $\mathcal{G}$ is a triangular robust graph, and $\mathcal{G}'$ is obtained from $\mathcal{G}$ by adding a node k, which is connected to a subset of nodes $S \subset \mathcal{V}$, $|S| \geq 2$ that are connected among themselves. Then $\mathcal{G}'$ is also a triangular robust graph.*

Proof. We check that the conditions of Definition 3 are satisfied by $\mathcal{G}'$.

Condition 1: By the Expansion Lemma (1), $\mathcal{G}'$ is *biconnected* as it is an expansion of $\mathcal{G}$ by adding the node k, which is adjacent to two nodes.
Condition 2: As node k is connected to a set of connected nodes, its neighbors are also connected.

It follows that $\mathcal{G}'$ satisfies the conditions of a triangular robust graph. □

Property 2. By constructing a triangular graph iteratively, we have that the minimum number of edges of a triangular robust graph is $2n-3$, $n \geq 3$. This is the same minimum number of edges for a $(2,2)$-robust graph [20].

Using our inductive construction method, we guarantee that the resulting graph has the minimum number of edges, which is in constrast to the construction method proposed in [10] [2]. Also, Theorem 4 leads to the particularly practical property that triangular robust graphs can be constructed incrementally, in a fully distributed manner. In fact, triangles are geometric configurations that can be easily formed during the deployment of networked robot teams, where each agent has a communication module that is constrained by a fixed radius $R > 0$. These robotic networks determine the set of connections based on the Euclidean distance $\mathcal{E} = \{(i,j)| \ \|v_j - v_i\| \leq R, \ \forall i,j \in \mathcal{V}\}$. For example, a simple formation with three robots maximizes its connectivity and its covered area by positioning the robots at the extremes of a regular triangle. Figure 5 shows a basic example of how an initial formation can be incrementally extended in order to cover a certain area with a triangular robust network.

[2] In the specific case of $(2,2)$-robustness, the algorithm of LeBlanc et al. requires three new edges for every new node (cf. Th. 5), whereas our algorithm requires only two new edges.

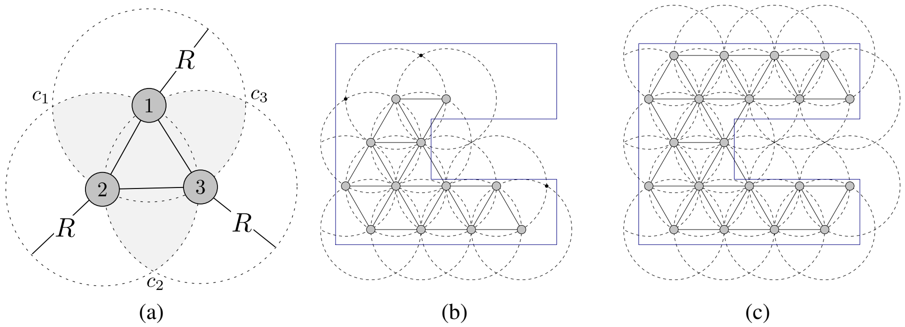
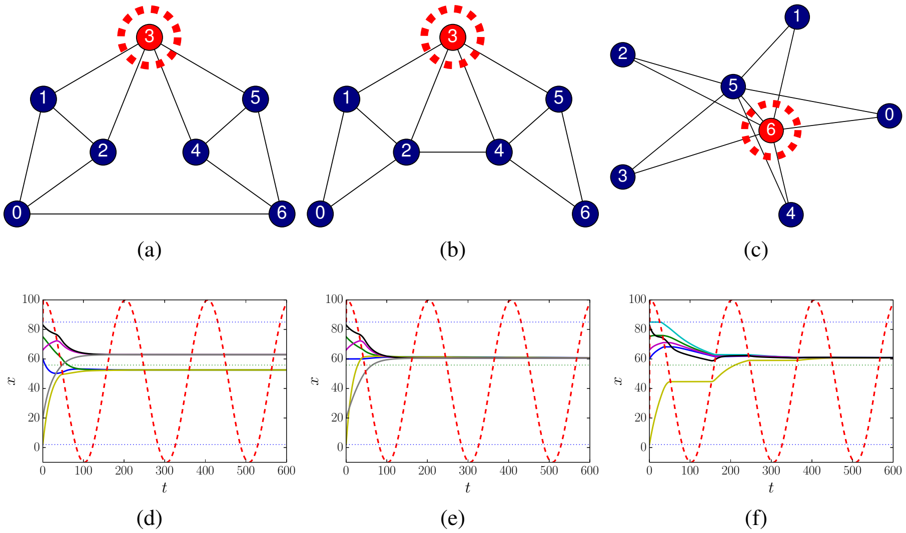


Fig. 5 Incremental expansion of a triangular robust network. Panel (a) shows the initial triangular robust network. By placing a subsequent node in the shaded area, the conditions for inductive construction are satisfied. Panels (b) and (c) depict how the network is grown to cover a given area iteratively, for $n = 12$ and $n = 20$, respectively.

4 Consensus in Triangular Robust Networks

In this section, we compare three different formations, and show how consensus is affected by their differing topologies. The formations consist of seven nodes. The nodes' initial values are $x[0] = [60, 75, 2, 85, 66, 83, 20]$. We consider a malicious node that shares an arbitrary (oscillatory) value, attempting to hinder convergence. Figure 6 shows the three different topologies, and demonstrates how the values of the nodes behave after applying the W-MSR algorithm (see Eq. (2)). Panel 6a shows a *biconnected*, *rigid* and *pseudo-triangular* graph, which is known as Laman graph [22]. Although the Laman graph has similar properties, as well as the same number of edges $2n-3$, it is neither $(2, 2)$-robust [3], nor triangular robust (the neighbors of node 3 are not connected among themselves). We can see in Figure 6d that the node values do not reach consensus, with two different values among the normal nodes that remain unchanged as time progresses. A triangular robust network is shown in Figure 6b. It is a simple variation of the previous Laman graph, where the edge $(0, 6)$ is replaced by the edge $(2, 4)$. As we stated earlier in Section 3, a successful convergence will reach consensus to a value within the range of initial values, i.e., a weighted average. This is demonstrated in Figure 6e. Finally, in Figure 6c, we show a triangular robust graph that is not planar (with intersecting edges), and that has two fully connected nodes. This topology represents the worst possible case, since one of the fully connected nodes is the malicious node (node 6 in the graph), with maximal influence over the other nodes in the graph. Also in this case, we see that the nodes reach consensus to a weighted average, as shown in Figure 6f. This particular example shows how the maximum value of the normal nodes does not change when the malicious node's value is the greatest value; instead, we see

[3] Consider the sets $S_1 = \{0, 1, 2\}$ and $S_2 = \{4, 5, 6\}$, since there are not 2-reachable nodes, the conditions for $(2, 2)$-robustness are not satisfied.



(a) (b) (c)

(d) (e) (f)

Fig. 6 Results of the resilient convergence method for different network topologies in the presence of a malicious agent. The network topologies are shown in (a)-(c), with the malicious agent shown in red, encircled by the dashed line. Panel (a) shows a Laman graph, panel (b) shows a planar triangular robust graph, and panel (c) shows a triangular robust graph with two fully-connected nodes. All normal nodes in the network perform the W-MSR update rule. The corresponding convergence behavior is shown in graphs (d)-(f), respectively. The average of the initial values is represented by the green dotted line, and the minimum and maximum of the initial values are depicted by the blue dotted lines. The malicious agent does not follow the update policy by sharing an arbitrary oscillatory value, denoted by the red dashed line.

that the minimum value of the normal nodes increases. An analogous behavior is observed when the malicious node's value the smallest value. As a consequence, the difference between the maximum and the minimum values of the normal nodes is continuously reduced, and the network achieves asymptotic convergence.

5 Conclusions and Future Work

In this paper, we proposed a particular network topology that is resilient to a malicious member node. We showed that this topology provides robustness, and guarantees asymptotic consensus in the presence of illegitimate information originating from one of the nodes. Especially, we showed that it satisfies two important properties for distributed systems: *(i)* it can be constructed incrementally by adding one node at a time such that the conditions for attachment can be computed locally, and *(ii)* its robustness can be verified through a distributed method by using only neighborhood-based information. The topology is fully scalable, and its robustness property can be validated in polynomial time. Also, its geometric properties lend themselves elegantly to applications of coverage and formation control for networked robot teams. In future work, we intend to study how our current method can be extended to withstand more than one malicious agent, i.e., $(F+1, F+1)$-robust graphs for $F > 1$. This is challenging because prior approaches assume unbounded communication radii (i.e., unlimited edge lengths between nodes that are being deployed in physical space). Furthermore, we will investigate the applicability of our methods to mobile problem settings, such as robust formation control.

Acknowledgements We gratefully acknowledge the support of the Colombian Innovation Agency (COLCIENCIAS), and the Brazilian agencies CAPES, CNPq, FAPEMIG. We also acknowledge the support of ONR grants N00014-15-1-2115 and N00014-14-1-0510, ARL grant W911NF-08-2-0004, NSF grant IIS-1426840, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

1. A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3, Dec 2002, pp. 2953–2958 vol.3.
2. R. O. S. R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the 2003 American Controls Conference*, 2003.
3. W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, April 2007.
4. H. Zhang and S. Sundaram, "Robustness of complex networks with implications for consensus and contagion," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 3426–3432.
5. N. A. Lynch, *Distributed algorithms*. Morgan Kaufmann, 1996.
6. J. Hromkovič, *Dissemination of information in communication networks: broadcasting, gossiping, leader election, and fault-tolerance*. Springer Science & Business Media, 2005.
7. F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2012.
8. S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2011.
9. H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *2012 American Control Conference (ACC)*, pp. 5855–5861.
10. H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, April 2013.
11. F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 650–663, 2010.
12. P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2480–2496, 2008.
13. H. Park and S. Hutchinson, "A distributed robust convergence algorithm for multi-robot systems in the presence of faulty robots," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2980–2985.

14. ——, “An efficient algorithm for fault-tolerant rendezvous of multi-robot systems with controllable sensing range,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 358–365.
15. J. Cortés, S. Martínez, and F. Bullo, “Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions,” *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.
16. W. Ren, R. W. Beard, and E. M. Atkins, “A survey of consensus problems in multi-agent coordination,” in *American Control Conference (ACC)*. IEEE, 2005, pp. 1859–1864.
17. R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
18. H. Zhang and S. Sundaram, “Robustness of information diffusion algorithms to locally bounded adversaries,” in *American Control Conference (ACC)*, 2012, pp. 5855–5861.
19. D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001.
20. H. J. LeBlanc and X. D. Koutsoukos, “Algorithms for determining network robustness,” in *Proceedings of the 2nd ACM international conference on High confidence networked systems*. ACM, 2013, pp. 57–64.
21. M. Ahmadi and P. Stone, “A distributed biconnectivity check,” in *Distributed Autonomous Robotic Systems 7*. Springer, 2006, pp. 1–10.
22. R. Haas, D. Orden, G. Rote, F. Santos, B. Servatius, H. Servatius, D. Souvaine, I. Streinu, and W. Whiteley, “Planar minimally rigid graphs and pseudo-triangulations,” in *Proceedings of the nineteenth annual symposium on Computational geometry*. ACM, 2003, pp. 154–163.