Toward fault-tolerant multi-robot networks

C. Ghedini
Computer Science Division, Technological Institute of Aeronautics, São José dos Campos, SP, Brazil

C. Ribeiro
Computer Science Division, Technological Institute of Aeronautics, São José dos Campos, SP, Brazil

L. Sabattini
Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy

Applications based on groups of self-organized mobile robots are becoming pervasive in communication networks, monitoring, traffic and transportation systems. Their advantage is the possibility of providing services without the existence of a previously defined infrastructure. However, physical agents are prone to failures that add uncertainty and unpredictability in the environments in which they operate. Therefore, a robust topology regarding failures is an imperative requirement. In this paper, we show that mechanisms based solely on connectivity maintenance are not enough to obtain a sufficiently resilient network, and a robustness-oriented approach is necessary. Thus, we propose a local combined control law that aims at maintaining the overall network connectivity while improving the network robustness via actions that reduce vulnerability to failures that might lead to network disconnection. We demonstrate, from a theoretical point of view, that the combined control law maintains connectivity, and experimentally validate it under diverse failure distributions, from two perspectives: as a reactive and as a proactive mechanism. As a reactive mechanism, it was able to accommodate ongoing failures and postpone or avoid network fragmentation, including cases where failures are concentrated over short time spans. As a proactive mechanism, the network topology was able to evolve from potentially vulnerable with respect to failures to a more robust one.

Keywords: Fault-tolerant networks, Multicooperative robot control, Adaptive networks, Resilient Systems, Complex networks, Multi-Robot networks

1. INTRODUCTION

Network services are becoming pervasive, and network infrastructures are widely available, thus making it possible to access the Internet, cloud technologies, and effective communication services in many circumstances. Nevertheless, there are wide areas of the world where network infrastructure is rudimentary or nonexistent, either because it is not possible from an economic or technological point of view, or because such a structure can get damaged in a disaster event [21].

Mobile robots, if equipped with appropriate communications devices, can be exploited to create an infrastructure network. For instance, interconnected mobile robots can provide rescuing devices in an exploration task, or generic clients (e.g., mobile phones, laptops, tablets, etc.) with communication services. Providing network services for unstructured environments employing interconnected mobile robotic systems involves several issues, from deployment and communication efficiency to topology control. In particular, it is necessary to guarantee the possibility of exchanging data among all the nodes in the network. Along these lines, connectivity maintenance is a well-studied topic in the field of decentralized multi-robot systems. The main approaches provide solutions for ensuring that, if the communication graph is initially connected, then it will remain connected, that is, if a link among two robots is active at time $t = 0$, than it will continue to be active as the system evolves, for time $t > 0$. Examples can be found in [1], [3], [11], [12], [16]. More recently, a few strategies reported in the literature propose a more flexible solution, that is based on a global measure of connectivity, namely the algebraic connectivity. With those strategies, single links are allowed to be added or removed, as long as the overall communication graph remains connected [18], [19].

The literature on connectivity maintenance does not generally consider that robots are prone to failures due to hardware or communication issues and hence does not provide robust solutions in this respect. As it is well known from the literature on Complex Networks — successive failures, particularly of agents playing a central role in the network topology, may easily lead to an inoperative or reduced service [2], [6], [10], [15]. Consider a network topology resulting from a hypothetical scenario of a multi-robot application presented in Figure 1: failures of robots highlighted in red can severely affect the network connectivity. Despite that, the detection and mitigation of vulnerable topological configurations regarding failures are mostly disregarded as technical issues.

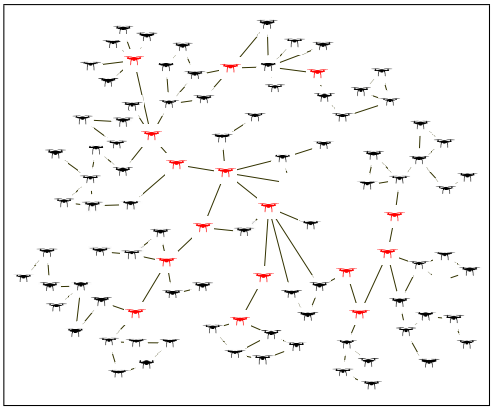


Fig. 1: Scenario of a mobile robot network topology.

In the context of this work, a vulnerable state means that

the network is prone to get disconnected if some nodes fail. Moreover, we consider robustness to failures as the system capacity to mitigate the effects of node failures through predictive actions that avoid topological configurations vulnerable to such effects. This contrasts with the standard definition of robustness used in Robust Control, namely satisfactory (under some performance metric) operation, regardless of bounded parametric uncertainties.

We propose herein a model that aims at overcoming the downside of the connectivity maintenance approaches, using as reference the mechanism proposed in [18] and combining it with a recent approach that aims at enhancing the network robustness to failures based only on locally available information [8]. This approach was introduced in [8] for a discrete-time system, and is here reformulated for continuous-time single integrator robotic systems, and is thus combined with the connectivity maintenance control strategy. The new combined control law was theoretically demonstrated to mantain connectivity and validated in extensive simulations that compare the behavior of the system with that achieved utilizing, in a separate manner, the connectivity maintenance and the robustness improvement control laws. The results demonstrate that the combined control law was able to evolve the network topology to a more robust one regarding failures. In addition, its performance in a failure scenario was assessed, demonstrating its effectiveness to adapt the network topology to accommodate failures, postponing or even avoiding network fragmentation. Preliminary results on this topic were presented in [7], where the proposed combined control strategy was evaluated on a limited set of simulations. In this paper, we extend the simulation setup by evaluating the impact of failure time distribution on the mechanism performance. Furthermore, a formal analysis of the system performance is presented.

The rest of this paper is organized as follows. The background on network properties is presented in Section 2. The system model and the algebraic connectivity control law are discussed in Section 3. The relevance of the problem addressed here is discussed in details in Section 4. Section 5 describes the combined model, and Section 6 presents the simulation model and discusses the results.

2. BACKGROUND ON NETWORK PROPERTIES

We will hereafter define some quantities that can be exploited for evaluating node and network connectivity and robustness to failures.

Consider an undirected graph $\mathcal{G}$, where $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G}) \subset \mathcal{V}(\mathcal{G}) \times \mathcal{V}(\mathcal{G})$ are the vertex set and the edge set, respectively. Moreover, let $W \in \mathbb{R}^{N \times N}$ be the weight matrix: each element w_{ij} is such that

$$w_{ij} = \begin{cases} w_{ij} > 0 & \text{if} \quad (i,j) \in \mathcal{E}(\mathcal{G}) \\ w_{ij} = 0 & \text{otherwise.} \end{cases} \tag{1}$$

Since the graph is undirected, then $w_{ij} = w_{ji}$.

Thus, let $\mathcal{L} \in \mathbb{R}^{N \times N}$ be the Laplacian matrix of graph $\mathcal{G}$ and $D = \text{diag}(\{k_i\})$ be the degree matrix, where k_i is the degree of the i-th node of the graph, i.e. $k_i = \sum_{j=1}^{N} w_{ij}$. The (weighted) Laplacian matrix of the graph is then defined as $L = D - W$. As is well known from algebraic graph theory, the Laplacian matrix of a graph $\mathcal{G}$ exhibits some remarkable properties regarding its connectivity [9]. Let λ_i, $i = 1, \ldots, N$ be the eigenvalues of the Laplacian matrix, then:

- The eigenvalues can be ordered such that

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N. \tag{2}$$

- Define now $\lambda = \lambda_2$. Then, $\lambda > 0$ if and only if the graph is connected. Therefore, λ is defined as the **algebraic connectivity** of the graph.
- Considering a weighted graph, λ is a non-decreasing function of each edge weight.

Even though the algebraic connectivity is more commonly used in the multi-robot systems literature for assessing the connectedness of a graph, it has been observed that in most real-world complex networks there is a large connected component together with a number of small components containing no more than a few percent of the nodes [4]. As the algebraic connectivity goes to zero as soon as the graph becomes disconnected, it does not provide further information about the network fragmentation. In this sense, the network connectivity $\mathcal{G}$ can be estimated by the relative size of the largest connected component, given by:

$$S(\mathcal{G}) = \frac{n_S}{N}, \tag{3}$$

where n_S is the number of nodes in the largest connected component and N is the number of nodes in the network. The *connected component* of a graph is a set of nodes such that a path exists between any pair of nodes in this set. For very large networks, this component is generally referred to as *giant component*. With a slight abuse of notation, we will hereafter define the **giant component** of a graph as its largest connected component, regardless of the network size.

In addition to the algebraic connectivity and the giant component, we want to evaluate the number of node failures a network can stand before disconnecting. It is known that different nodes have different roles in maintaining the overall network connectivity. In special, the concept of *centrality* is usually exploited for identifying the most important nodes within a graph [13]. Several indicators can be found in the literature for defining centrality. In particular, referring to connectivity maintenance, we will consider the concept of *Betweenness Centrality* (BC) [22], which establishes higher scores for nodes that are contained in most of the shortest paths between every pair of nodes in the network.

For a given node i and pair of nodes j, l, the importance of i as a mediator of the communication between j and l can be established as the ratio between the number of shortest paths linking nodes j and l that pass through node i ($g_{jl}(i)$), and the total number of shortest paths connecting nodes j and l (g_{jl}). Then, the BC of a node i is simply the sum of this value over all pairs of nodes, not including i:

$$BC(i) = \sum_{j<l} \frac{g_{jl}(i)}{g_{jl}}. \tag{4}$$

Once the BC has been computed for all the nodes, it is possible to order them from the *most central* (i.e., the node with the highest BC value) to the *less central* (i.e., the node with the lowest BC value). Hence, let $[v_1, \ldots, v_N]$ be the list of nodes ordered by descending value of BC.

According to the definition of centrality, removing the most central nodes might lead to network fragmentation. We therefore introduce the following definition of *Robustness level*.

Definition 2.1 (Robustness level [8])**.** *Consider a graph $\mathcal{G}$ with N nodes. Let $[v_1, \ldots, v_N]$ be the list of nodes ordered by descending value of BC. Let $\varphi < N$ be the minimum index $i \in [1, \ldots, N]$ such that, removing nodes $[v_1, \ldots, v_i]$ leads to disconnecting the graph, that is, the graph including only nodes $[v_{\varphi+1}, \ldots, v_N]$ is disconnected.*

Then, the level of network ($\mathcal{G}$) robustness *is defined as:*

$$\Theta(\mathcal{G}) = \frac{\varphi}{N}. \tag{5}$$

$\diamond$

The level of robustness defines the fraction of central nodes that need to be removed from the network to obtain a disconnected network, i.e., $S(\mathcal{G}) < 1$. Small values of $\Theta(\mathcal{G})$ imply that the graph can lose connectivity in case of failures of a small fraction of its nodes. Therefore, increasing this value increases the network robustness to failures.

Notice that $\Theta(\mathcal{G})$ is only an estimate of how far the network is from getting disconnected with regard to the fraction of nodes removed. In fact, it might be the case that different orderings of nodes with the same BC produce different values of $\Theta(\mathcal{G})$.

The magnitude of the topological vulnerability of a node can be locally estimated by means of information acquired from its 1-hop and 2-hops neighbors [8]. Let $d(v, u)$ be the shortest path between nodes v and u, i.e., the minimum number of edges that connect nodes v and u. Subsequently, define $\Pi(v)$ as the set of nodes from which v can acquire information:

$$\Pi(v) = \{u \in V(G) : d(v, u) \leq 2\}.$$

Moreover, let $|\Pi(v)|$ be the number of elements of $\Pi(v)$. In addition, define $\Pi_2(v) \subseteq \Pi(v)$ as the set of the 2-hop neighbors of v, that comprises only nodes whose shortest path from v is exactly equal to 2 hops, namely

$$\Pi_2(v) = \{u \in V(G) : d(v, u) = 2\}.$$

Now define $L(v, u)$ as the *number of paths* between nodes v and u, and let $Path_\beta(v) \subseteq \Pi_2(v)$ be the set of v's 2-hop neighbors that are reachable through at most β paths, namely

$$Path_\beta(v) = \{u \in \Pi_2(v) : L(v, u) \leq \beta\}.$$

Notice that β defines the threshold for the maximal number of paths between a node v and each of its u neighbors that are necessary to include u in $Path_\beta(v)$. Therefore, using a low value for β allows to identify the most weakly connected 2-hop neighbors. Hence, the value of $|Path_\beta(v)|$ is an indicator of the magnitude of node fragility regarding connectivity, and **the vulnerability level of a node regarding failures** is given by $P_\theta(v) \in (0, 1)$:

$$P_\theta(v) = \frac{|Path_\beta(v)|}{|\Pi(v)|}. \tag{6}$$

We will hereafter use $\beta = 1$, in order to identify 2-hop neighbors that are connected by a single path, which can represent a critical situation for network connectivity.

3. System model and algebraic connectivity maintenance

In this section we define the system model and summarize the properties of the algebraic connectivity maintenance control law introduced in [18].

Consider a multi-robot system composed of N robots that are able to communicate with other robots within the same communication radius R. The resulting communication topology can be represented by an undirected graph $\mathcal{G}$ where each robot is a node of the graph, and each communication link between two robots is an edge of the graph. Let each robot state be its position $p_i \in \mathbb{R}^m$, and let $p = \left[p_1^T \ldots p_N^T\right]^T \in \mathbb{R}^{N \times m}$ be the state vector of the multi-robot system. Let each robot be modeled as a single integrator system, whose velocity can be directly controlled, namely

$$\dot{p}_i = u_i, \tag{7}$$

where $u_i \in \mathbb{R}^m$ is a control input. It is worth remarking that, by endowing a robot with a sufficiently good cartesian trajectory tracking controller, it is possible to use this simple model to represent the kinematic behavior of several types of mobile robots, like wheeled mobile robots [20], and UAVs [14].

In order to guarantee the connectivity of $\mathcal{G}$, [18] proposes an approach to solve the connectivity maintenance problem in a decentralized manner, utilizing the algebraic connectivity property. For this purpose, consider a weighted graph, where the edge weights w_{ij} introduced in (1) are defined according to the following:

Definition 3.1. *The edge weights w_{ij} exhibits the following properties, $\forall i, j = 1, \ldots, N$:*

(P1) $w_{ij} \geq 0$.
(P2) $w_{ij} = w_{ij}(\|p_i - p_j\|)$.
(P3) $w_{ij} = 1$ *if* $\|p_i - p_j\| = 0$.
(P4) $w_{ij} = 0$ *if* $\|p_i - p_j\| \geq R$.
(P5) $w_{ij}(d)$ *is non-increasing with respect to its argument* d.

$\diamond$

Define now $\epsilon > 0$ to be the desired lower-bound for the value of λ. The control strategy will then be designed to ensure that the value λ never goes below ϵ. An *energy function* is utilized for generating the decentralized connectivity maintenance control strategy, namely:

Definition 3.2. *An energy function*

$$V(\lambda) = V(\lambda(p)) : \mathbb{R}^{Nm} \mapsto \mathbb{R}$$

exhibits the following properties:

(P1) It is continuously differentiable $\forall \lambda > \epsilon$.
(P2) It is non-negative.
(P3) It is non-increasing with respect to λ, $\forall \lambda > \epsilon$.
(P4) It approaches a constant value, as λ *increases.*
(P5) It suddenly increases, as λ *approaches* $\epsilon > 0$*, namely*

$$\lim_{\lambda \to \epsilon} V(\lambda(p)) = \infty.$$

$\diamond$

As an example, in [18] the following energy function and edge weights were used, respectively:

$$V(\lambda) = \begin{cases} \coth(\lambda - \epsilon) & \text{if } \lambda > \epsilon \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

$$w_{ij} = \begin{cases} e^{-(\|p_i - p_j\|^2)/(2\sigma^2)} & \text{if } \|p_i - p_j\| \leq R \\ 0 & otherwise, \end{cases} \quad (9)$$

where the scalar parameter σ is chosen to satisfy the threshold condition $e^{-(R^2)/(2\sigma^2)} = \Delta$, with Δ defined as a small predefined threshold. This definition of the edge–weights introduces a discontinuity in the control action, that can be avoided introducing a smooth bump function, as in [5]. However, from an implementation viewpoint, the effect of the discontinuity can be made negligible by defining the Δ sufficiently small.

The designed control law drives the robots to perform a gradient descent of $V(\cdot)$, in order to ensure connectivity maintenance. Considering the dynamics of the system introduced in (7), the control law is defined as follows:

$$u_i = u_i^c = -\frac{\partial V(\lambda)}{\partial p_i} = -\frac{\partial V(\lambda)}{\partial \lambda}\frac{\partial \lambda}{\partial p_i}. \quad (10)$$

Since λ and its gradient are global quantities, the proposed control law is centralized. Decentralized implementation can be achieved replacing λ and its gradient with their estimates, computed by each robot in a decentralized manner utilizing the procedure proposed in [18].

This methodology does not consider the fact that robots can unexpectedly fail, thus stopping their activity due to mechanical, electrical or software issues. As pointed out before, the possible drawback of robot failure in the overall network connectivity needs to be avoided.

4. Problem Definition: Robustness to Failures

The topological properties resulting from applications relying on dynamic networks are most of the time unknown, costly to estimate and change over time, adding uncertainty to the system behavior. Despite such variability, in general, these networks are often able to maintain most of the nodes into the giant component. However, this is not necessarily the case when such topological variability is biased towards nodes of high centrality that either leave or fail, or when the system is facing successive or cascading failures, conducting networks to a global state of vulnerability where its operation is completely compromised [6].

As an example, consider nodes 12 and 17 of the random network in Figure 2a. They are clearly playing a crucial role in the network communication. If any of them fails, the network efficiency to communicate will be degraded. If both fail, the network fragments into two clusters, as illustrated in Figure 2b. Besides, node 16 is in a vulnerable configuration since its communication with the entire network depends on node 9. Thus, for applications where the interaction among nodes is critical to the system functionalities, such *harmful* topological configurations should be avoided to reduce the impact of possible node failures.

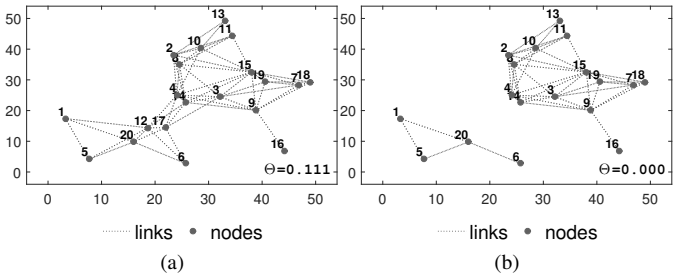


Fig. 2: A random network topology: initial configuration (a) and after two node failures (b).

As described in Section 3, control strategies exist that aim at keeping nodes connected [17], [18], but those *per se* do not ensure robustness to failures. For demonstrating that, the approach based on the algebraic connectivity maintenance described in Section 3 was applied to the network in Figure 2a, resulting in the topology illustrated in Figure 4a (see Section 6.1 for model parameterization details). Despite a slight improvement in the network robustness, from Θ=0.11 to Θ=0.17, the vulnerable configurations highlighted before still persist.

We therefore argue that algebraic connectivity is not the most suitable property for evaluating or controlling the capacity of the network to accommodate failures. For reinforcing that, notice that the network shown on the right in Figure 3, even exhibiting a larger algebraic connectivity than the network on the left, is more affected by the central node failure.

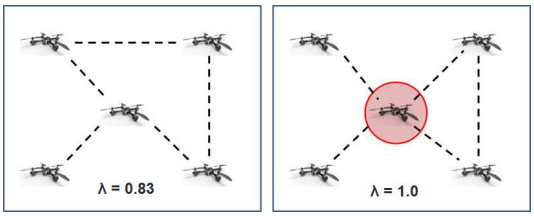
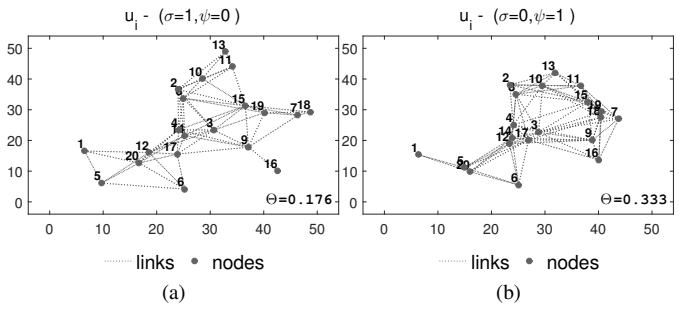


Fig. 3: Network with larger algebraic connectivity (right) is more affected by central node failure.

In this regard, a continuous-time model based on the approach that aims at improving the network robustness proposed in [8] was also applied to the same network (Figure 4b) — see Section 5 for details. Notice that node 16 is now also directly connected to nodes 3, 17 and 18; and node 6 to node 14, improving the robustness from Θ=0.11 to Θ=0.33 and suppressing the vulnerable topological configurations. The downside of this strategy is not ensuring the network

connectivity, i.e., the network can become disconnected during (or as a result of) the adaptive process.



(a) (b)

Fig. 4: Resulting network topology for the algebraic connectivity control law (a) and for the robustness improvement control law (b).

Summarizing, connectivity is a crucial requirement in decentralized multi-robot systems: in order to achieve a common objective, robots may need to exchange information. On the other hand, robots are likely to fail. The examples discussed in this Section demonstrate the importance of a robust connectivity maintenance strategy regarding failures. Given this evidence, in the next Section we aim at solving the following problem:

Problem. *Given a multi-robot system, design a local estimation procedure that allows each robot to assess its vulnerability level, based on locally available information, and subsequently exploit this estimate for controlling the motion of the robots in such a way that the overall robustness to failures is increased.*

5. Control law for connectivity maintenance and robustness to failure improvement

This section describes the unified model that aims at improving the topological robustness of networks to failures while ensuring the connectivity maintenance. In particular, considering the dynamic introduced in (7), we define the following control law

$$u_i = \sigma u_i^c + \psi u_i^r, \tag{11}$$

where u_i^c is the connectivity maintenance control law introduced in (10), and u_i^r is an additional control law that aims at improving robustness to failures. Moreover, $\sigma, \psi \geq 0$ are design parameters that represent control gains. Naturally, setting either σ or ψ to zero removes the effect of the corresponding control law. Conversely, if both parameters are larger than zero, both control actions are simultaneously active.

Based on the definition of the node vulnerability level, given by (6), we now introduce a control law u_i^r that leads to improving the network robustness.

Assume that the i-th robot identifies itself as vulnerable. In this case, the aim of the control strategy is to increase the number of links towards the 2-hop neighbors of i that are in $Path_\beta(i)$. Hence, define $x_\beta^i \in \mathbb{R}^m$ as the barycenter of the positions of the robots in $Path_\beta(i)$, namely

$$x_\beta^i = \frac{1}{|Path_\beta(i)|} \sum_{j \in Path_\beta(i)} x_j. \tag{12}$$

Considering the dynamics of the system introduced in (7), the control law is defined as follows:

$$u_i^r = \frac{x_\beta^i - x_i}{\left\| x_\beta^i - x_i \right\|} \alpha, \tag{13}$$

where $\alpha \in \mathbb{R}$ is the linear velocity of the robots, that we assume constant for the sake of simplicity.

This control law drives vulnerable robots towards the barycenter of the positions of robots in their $Path_\beta$, thus decreasing the distance between them and eventually creating new edges in the communication graph. It is worth noting that (6) provides a decentralized methodology for each robot to evaluate its vulnerability level.

The control law in (13) is applied in a probabilistic manner, with higher probability for those robots i whose value $P_\theta(i)$ is high. This is obtained comparing $P_\theta(i)$ with a random number $r \in (0, 1)$: if $P_\theta(i) > r$, then the i-th robot considers itself as vulnerable, and applies the control law in (13).

We will hereafter analyze the performance of the proposed combined control law introduced in (11). In particular, the control law u_i^c was proven in [17], [18] to guarantee positiveness of the generalized connectivity. The following theorem shows that, even in the presence of the additional robustness improvement control law u_i^r, generalized connectivity maintenance is always guaranteed, $\forall \sigma > 0$, thus ensuring that the algebraic connectivity remains positive.

Theorem 5.1. *Consider the dynamical system described by* (7)*, and the control laws described in* (10)*,* (11) *and* (13)*. Then, if the initial value of $\lambda(t)$, namely $\lambda(0)$, is greater than ϵ, then the value of $\lambda(t)$ will remain positive, as the system evolves, thus implying algebraic connectivity maintenance.*

Proof. In order to prove the statement, consider an energy function $V(\lambda(p))$ defined according to Definition 3.2. In particular, its time derivative can be computed as follows:

$$\dot{V}(\lambda(p)) = (\nabla_p V(\lambda(p)))^T \dot{p} = \sum_{i=1}^{N} \left(\frac{\partial V(\lambda(p))}{\partial p_i} \right)^T \dot{p}_i \tag{14}$$

Considering the control law introduced in (11), we obtain the following:

$$\dot{V}(\lambda(p)) = \sum_{i=1}^{N} \left(\frac{\partial V(\lambda(p))}{\partial p_i} \right)^T (\sigma u_i^c + \psi u_i^r). \tag{15}$$

Considering then the definition of u_i^c given in (10), then the following holds:

$$\dot{V}(\lambda(p)) = \sum_{i=1}^{N} \left(\frac{\partial V(\lambda(p))}{\partial p_i} \right)^T \left(-\sigma \frac{\partial V(\lambda(p))}{\partial p_i} + \psi u_i^r \right). \tag{16}$$

It is worth noting that

$$\frac{\partial V(\lambda(p))}{\partial p_i} = \frac{\partial V(\lambda(p))}{\partial \lambda(p)} \frac{\partial \lambda(p)}{\partial p_i}. \tag{17}$$

Subsequently, it is possible to rewrite (16) as follows:

$$\dot{V}(\lambda(p)) = \frac{\partial V(\lambda(p))}{\partial \lambda(p)} \sum_{i=1}^{N} \left(\frac{\partial \lambda(p)}{\partial p_i}\right)^T \left(-\sigma \frac{\partial V(\lambda(p))}{\partial \lambda(p)} \frac{\partial \lambda(p)}{\partial p_i} + \psi u_i^r\right). \tag{18}$$

Thus, considering the definition of u_i^r given in (13), from (18) we obtain the following inequality:

$$\dot{V}(\lambda(p)) \leq \frac{\partial V(\lambda(p))}{\partial \lambda(p)} \sum_{i=1}^{N} \left(-\sigma \frac{\partial V(\lambda(p))}{\partial \lambda(p)} \left\|\frac{\partial \lambda(p)}{\partial p_i}\right\|^2 + \left\|\frac{\partial \lambda(p)}{\partial p_i}\right\| \psi\alpha\right). \tag{19}$$

Consider now the definition of the energy function given in Definition 3.2. Since the energy function is non-increasing with respect to its argument λ, then the following holds:

$$\frac{\partial V(\lambda(p))}{\partial \lambda(p)} < 0. \tag{20}$$

Moreover, since the energy function is continuously differentiable, and it approaches infinity as λ approaches ϵ, then the following property holds:

$$\lim_{\lambda \to \epsilon} \left\|\frac{\partial V(\lambda(p))}{\partial \lambda(p)}\right\| = \infty. \tag{21}$$

According to (20), the inequality in (19) can be rewritten as follows:

$$\sum_{i=1}^{N} \left(-\sigma \frac{\partial V(\lambda(p))}{\partial \lambda(p)} \left\|\frac{\partial \lambda(p)}{\partial p_i}\right\|^2 + \left\|\frac{\partial \lambda(p)}{\partial p_i}\right\| \psi\alpha\right) \geq 0. \tag{22}$$

If the following condition holds

$$\sum_{i=1}^{N} \left\|\frac{\partial \lambda(p)}{\partial p_i}\right\|^2 \neq 0, \tag{23}$$

then the inequality in (22) can be rewritten as follows:

$$-\frac{\partial V(\lambda(p))}{\partial \lambda(p)} \geq \frac{\psi\alpha \sum_{i=1}^{N} \left\|\frac{\partial \lambda(p)}{\partial p_i}\right\|}{\sigma \sum_{i=1}^{N} \left\|\frac{\partial \lambda(p)}{\partial p_i}\right\|^2}. \tag{24}$$

According to (21), a value $\bar{\lambda} > \epsilon$ always exists such that (24) is satisfied, $\forall \lambda \in (\epsilon, \bar{\lambda}]$.

Thus, $\forall \lambda \in (\epsilon, \bar{\lambda}]$ the function $\dot{V}(\lambda(p)) \leq 0$. Then, $\forall \lambda \in (\epsilon, \bar{\lambda}]$, the energy function $V(\lambda(p))$ does not increase over time.

Let $\lambda(0) > \epsilon$ be the initial value of λ. If $\lambda(0) > \bar{\lambda}$, then the value of λ will always be lower-bounded by $\bar{\lambda}$. Conversely, if $\lambda(0) \leq \bar{\lambda}$, then the value of λ will increase, until $\lambda \geq \bar{\lambda}$, and then it will never go below $\bar{\lambda}$. Namely, let

$$\hat{\lambda} = \min\left(\lambda(0), \bar{\lambda}\right). \tag{25}$$

Then, the value of λ will never go below the value of $\hat{\lambda} > \epsilon$.

In case $\left\|\frac{\partial \tilde{\lambda}_2}{\partial p_i}\right\| = 0$, the condition in (23) is not verified. However, in this case

$$\dot{\lambda} = \frac{\partial \lambda}{\partial p_i} \dot{p}_i = 0. \tag{26}$$

This implies that the value of λ is constant over time, thus it is lower-bounded by its initial value. In both cases, $\lambda > \epsilon$. Therefore, it is possible to conclude that the generalized connectivity λ remains greater than zero as the system evolves. Considering the definition of the edge weights w_{ij} given in Definition 3.1, this implies that the algebraic connectivity λ will remain positive, which implies that connectivity maintenance is always guaranteed.

6. Experimental results

6.1. Simulation model

The proposed unified control strategy was validated using a simulation model, developed in Matlab. The model encompasses the benchmark networks, the control law parameterization, and the protocol to evaluate its performance.

The benchmark networks were generated through a random positioning of a variable number N of robots in $\mathbb{R}^2$, over a bounded area of size S^2. Given the communication radius R, the communication network is then generated, based on the relative positioning of the robots. For this simulation we set $N = 20$, $S = 50$ and $R = 16$.

The control law performance was evaluated considering a fault-free environment, where networks can evolve from a potentially vulnerable topology to a more robust one and a fault-prone environment, where networks are experiencing disturbances (*i.e.*, node failures). For both scenarios, at discrete time intervals t_i, the network properties are measured. For the second, in addition to the properties computation, the most central node regarding the BC ranking, as defined in (4), is removed from the network. Repeated experiments were carried out in order to assess the performance of the proposed methodology in a statistically sound manner (# in Table I indicates the number of repetitions). The model parameterization settings are presented in Table I.

TABLE I: Simulation settings

model			u_i^c	u_i^r	
t	t_i	#	ϵ	β	α
10	1	200	0.25	1	$0.15R$

For the combined control law strategy evaluation, we assume that the robustness to failure must not overpower the network connectivity to produce a more robust network. Thus, the combination σ=2 and ψ=1 was used for demonstrating the combined control law performance. The results were also confronted with each individual control law. For σ=0 and ψ=1 combination, only the robustness improvement control law is active, for σ=1 and ψ=0 only the algebraic connectivity control law is active. We also evaluate the scenario where no control law is active (σ=0, ψ=0). Take into account that any combination of positive gains leads to the desired behavior, the gain settings should be defined according to the application requirements.

6.2. Robustness to failures evaluation

This section presents the simulation results according to the model defined above. Regarding the robustness to failures

improvement in fault-free environment scenario, Figure 5 illustrates, for each gain combination, the robustness level and the algebraic connectivity values at each time interval. As expected, the simulation results demonstrate the robustness improvement for ψ=1, i.e., when the robustness control law is active. Besides, the combined control law achieved a better performance than the robustness improvement control law, with the advantage of ensuring network connectivity during the adaptive process.

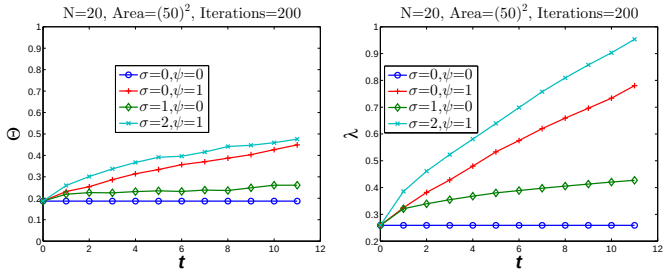


Fig. 5: The robustness level (Θ) and the algebraic connectivity (λ) evolution.

Figures 6 and 7 demonstrate the combined control law efficiency to evolve the initial network topologies (on the left) to a more robust one (on the right). In figure 6, notice that nodes 13, 15 and 17 are notably vulnerable. On the other hand, the communication in the network illustrated in Figure 7 is relying on nodes 11, 15 and 16. Failures of any of these nodes may lead the network to a critical state of connectivity. For both examples, the harmful topological configurations were overcome by applying the combined control law.

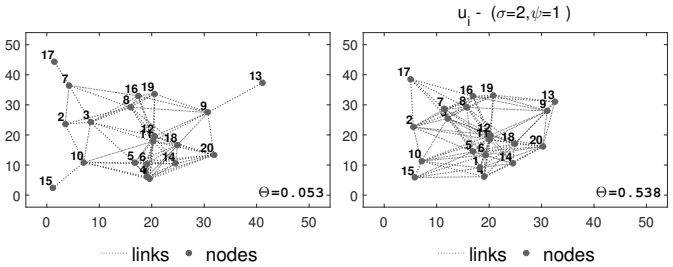


Fig. 6: Example of the combined control law performance.

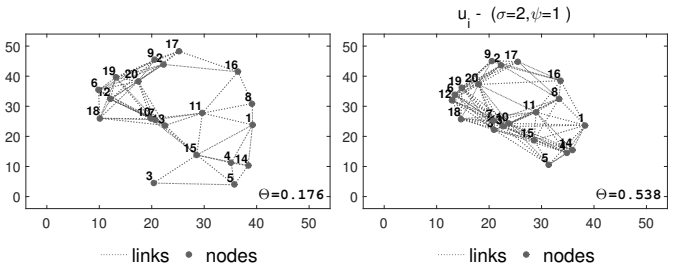


Fig. 7: Example of the combined control law performance.

It is also observed that improving the network robustness to failure can result in dense networks. For evaluating the feasibility of providing more robust networks but still preserving or increasing the area covered by the robots, we conducted preliminary tests where we introduced an additional control term to (13), aiming at improving area coverage. Such an additional control term leads to increasing inter-robot distances, and is activated when an appropriately introduced gain ζ is non-zero. The evolution of the network density under such a a control law is depicted in Figure 8. We consider different scenarios: while we assume that the connectivity preserving control law is always active (σ=2), in the first scenario we combine it with only area coverage improvement control law (σ=2, ψ=0, ζ=1), in the second scenario we combine it with the robustness improvement control law (σ=2, ψ=1, ζ=0), and it the third scenario we combine it with both control laws (σ=2, ψ=1, ζ=1). Notice that the robustness mechanism provides denser networks in contrast to less dense ones produced by the coverage mechanism. On the other hand, combining robustness and coverage generated more balanced topological configuration: this suggests that a model comprising several (possibly conflicting) control laws leads to achieving more efficient network topologies.

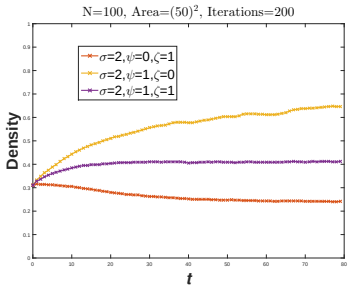


Fig. 8: Network density evolution.

For the fault-prone environment scenario, the combined control strategy was able, on average, to maintain the robustness level and the algebraic connectivity values (Figure 9).

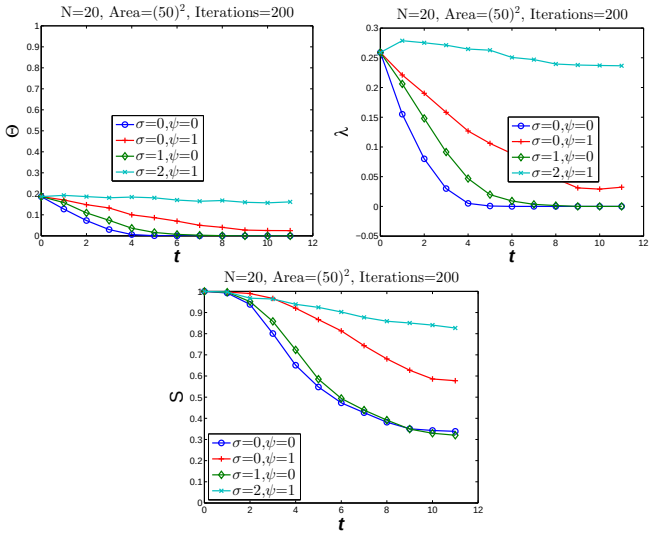


Fig. 9: The robustness level (Θ), the algebraic connectivity (λ) and the giant component (S) evolution under failures.

The giant component results highlight the impact of node failures on its value when there is no adaptive mechanism

active and when the combined control law is active, blue line (σ=0, ψ=0) and turquoise line (σ=2, ψ=1), respectively. For the first, a fraction of 0.32 nodes are into the giant component after 12 seconds of simulation, in contrast to a fraction of 0.84 for the latter, clearly demonstrating the evolution to more resilient networks, able to react to unexpected failures through the adaptive mechanism. It is important to emphasize that in failure situations there is no way of ensuring that the network will remain connected because the failure probability distribution is usually unknown. Thus, the giant component evolution analysis is crucial to properly evaluate the mechanism performance.

The importance of failure mitigation is emphasized by showing the resulting topologies for networks in Figures 6 and 7 for σ=0, ψ=0 (on the left) and σ=2, ψ=1 (on the right) gain settings, illustrated in Figures 10 and 11, respectively. In such cases, the combined control law was able to maintain the networks connected even in the case of unexpected failures of central nodes, demonstrating its capacity to be reactive, despite a potentially vulnerable initial topological configuration.

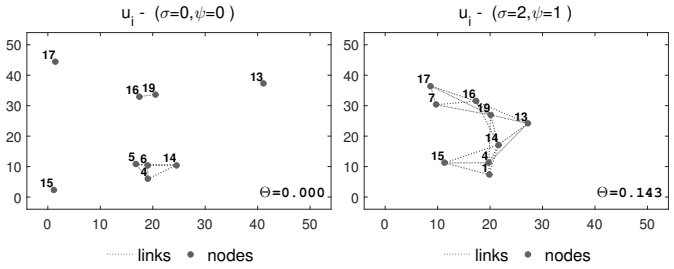


Fig. 10: Example of the combined control law performance under failures.

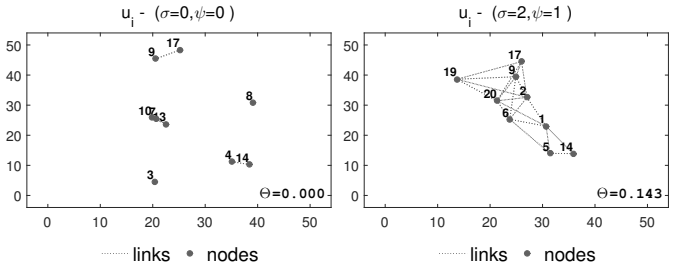


Fig. 11: Example of the combined control law performance under failures.

Finally, for exemplifying the network evolution during the attack simulation process, Figure 12 exhibits snapshots of the initial network shown in Figure 7 at t_i=4 simulation time. It is possible to observe the connectivity maintenance control law ineffectiveness to maintain the network nodes connected, i.e., the network is fragmented for σ=1 and ψ=0. Moreover, the robustness control law (σ=0, ψ=1) was not able to overcome the vulnerable configuration, represented by the red nodes, i.e., those that defined themselves as vulnerable according to (7) and a random probability (see Section 5). In contrast, the combined mechanism (σ=2, ψ=1) clearly provided a more resilient network. Some additional examples can be freely viewed online on https://www.youtube.com/watch?v=dThj_-BdkJw&t=264s

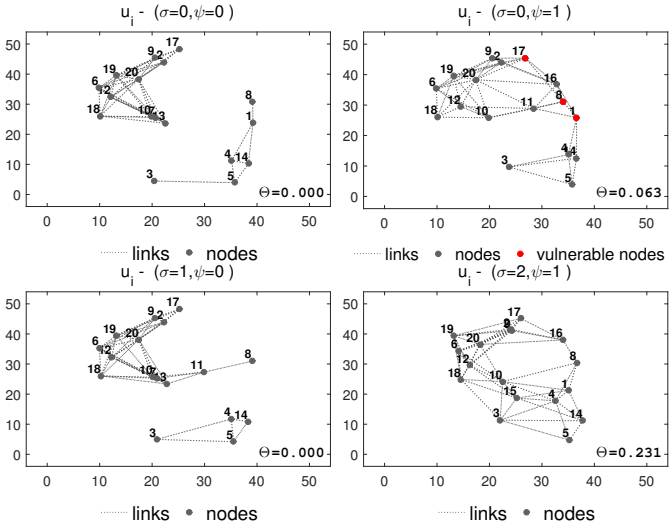


Fig. 12: Network topology at 4 seconds of simulation.

6.3. Failure distribution evaluation

The results presented in Section 6.2 consider that the most central node regarding the BC ranking fails every 1 second along the total simulation time of 10 seconds. In this section, we want to evaluate the impact of failure time distribution on the combined control law performance. For that, each robot is now supposed to move forward into the environment increasing its robustness to failures, when necessary, while keeping the global network connectivity. On the other hand, the times in which failures will occur are given by a time vector $f_t = (f_{t1}, f_{t2}...f_{tn})$, where n is the number of failures. This set of failure times is generated according to the procedure described in what follows.

Let t_f be the total simulation time, t_e the network evaluation time interval, f_n the fraction of network nodes to fail, and f_d the fraction of node failures that must occur into a fault range time interval $t_r = [tr_i, tr_f]$. The fault range interval defines the time interval in which a exactly fraction f_d of nodes fail. For instance, setting $N = 100$, $t_i = 0$, $t_f = 80$, $f_n = 0.7$, $f_d = 0.8$, and $t_r = [30, 50]$ means that 70 nodes will fail during the entire simulation with 56 (80% of 70) failing between 30 and 50 seconds of simulation ($t_f(i) >= 30$ and $t_f(i) <= 50$). It is important to highlight that the remaining failures times are generated at random outside the correspondent range specification.

For each parametrization setting, a set of failure times f_t is generated, i.e., for each network, its properties evolution regarding a specific time interval t_r are averaged over $\#t_r$ sets of different times according that range.

For properly evaluating the impact of different time failure distributions on the mechanism performance, we predefined failure time ranges regarding the moment when they occurred: beginning ($1st$), middle ($2nd$) and ending ($3rd$) of the simulation time. The results were confronted with a uniform random time failure distribution. Figure II shows the

averaged distribution time failures generated for each of the specified ranges.

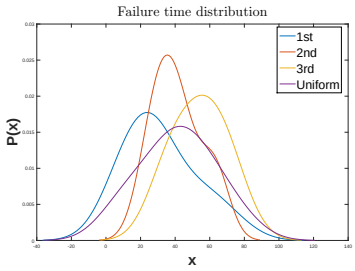


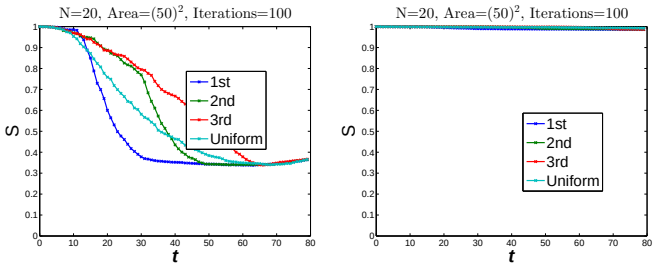
Fig. 13: Failure time distributions.

Table II presents the setup parameters concerning the distribution experiment. Parameterization for u_i^c and u_i^r models are the same applied to the previous experiment (see Table I). We ran the simulation considering the combined mechanism as both active (σ=2,ψ=1) and non-active (σ=0,ψ=0).

TABLE II: Time distribution experiment settings

t_f	t_e	t_r	$\#t_r$	f_n	f_d	N
80	1	$[10,30][30,50][50,70]$	50	0.7	0.6	100

The impact of each time failure distribution on the giant component (S) is presented in Figure 14. Its value decreases according to the occurrence of faults, i.e., the more perturbations at the beginning, the sooner the lowest value of S is reached (See Figure 14a). However, the same pattern does not apply to scenarios in which the combined control law is active. As demonstrated in Figure 14b, despite most failures concentrating at specific ranges, the combined control law was able to suppress their impact.



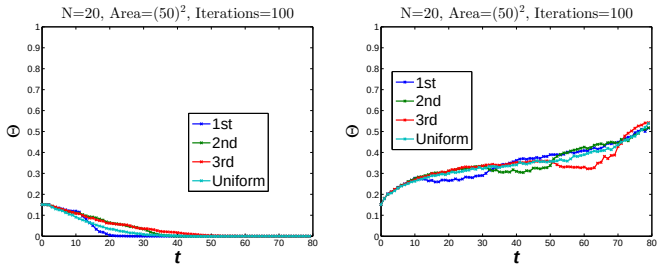
(a) No control law active (σ=0,ψ=0).

(b) Both control laws active (σ=2,ψ=1).

Fig. 14: The giant component (S) performance under different time failure distributions.

This feature is reinforced by the robustness level evolution (Figure 15b): even with faults occurring at very close times, networks were able to maintain their ability to withstand failures without disconnecting, as opposed to the case of networks with no active mechanism (Figure 15a).

The robustness to failure mechanism contribution is emphasized in Figure 16b. The algebraic connectivity of networks (λ), and as a consequence their connectivity, is hardly affected by consecutive failures. However, notice that the vulnerability level measure (6) was effective in providing means for nodes

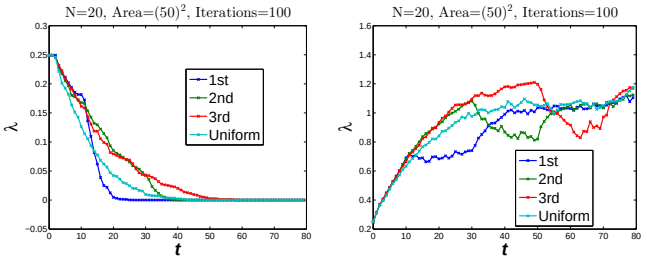


(a) No control law active (σ=0,ψ=0).

(b) Both control laws active (σ=2,ψ=1).

Fig. 15: The robustness level (Θ) performance under different time failure distributions.

to detect their possible vulnerable states, after perturbations in the neighborhood. Similar effectiveness is achieved by the robustness to failure improvement strategy, described in Section 5.



(a) No control law active (σ=0,ψ=0).

(b) Both control laws active (σ=2,ψ=1).

Fig. 16: The algebraic connectivity (λ) performance under different time failure distributions.

7. Conclusions

We proposed in this paper a mechanism that locally and autonomously combines connectivity maintenance and robustness with respect to node failures in a multi-robot setting. Both aspects are important in the design of cooperative multi-robot systems, the first for guaranteeing communication among all robots, the second for reducing the network vulnerability in the case of failures that can happen as a consequence of hardware unreliability or environmental hostility. The combined control law performance was experimentally validated from two perspectives: as a reactive mechanism, it was able to accommodate ongoing failures and postpone or avoid network fragmentation, and as a proactive mechanism, the network topology evolved towards a more robust topology regarding node failures. The effectiveness of the proposed combined control law was demonstrated under several failure time distributions, including cases where those failures are concentrated over short time spans. As ongoing work, we are now implementing the algorithms in a realistic multi-robot setting, also considering additional constraints such as obstacle avoidance. For future work, we intend to extend our approach considering additional requirements, such as coverage, energy consumption and heterogeneous robotic teams.

ACKNOWLEDGMENTS

The authors thank FAPESP for the financial support to carry out this research (procs. no. 2012/25058-9 and 2014/13800-8). Carlos H. C. Ribeiro also thanks CNPq (proc. no. 303738/2013-8) and FAPESP (proc. no. 2013/13447-3).

REFERENCES

[1] A. Ajorlou, A. Momeni, and A.G. Aghdam, A class of bounded distributed control strategies for connectivity preservation in multi–agent systems, IEEE Trans Automatic Control 55 (2010), 2828–2833.

[2] R. Albert, H. Jeong, and A.L. Barabasi, Error and attack tolerance of complex networks, Nature 406 (July 2000), 378–382.

[3] Y. Cao and W. Ren, Distributed coordinated tracking via a variable structure approach – part I: consensus tracking. part II: swarm tracking, Proc Am Control Conference, 2010, pp. 4744–4755.

[4] P.Y. Chen and K.C. Chen, Information epidemics in complex networks with opportunistic links and dynamic topology, Global Telecommunications Conference (GLOBECOM), IEEE, December 2010, pp. 1–6.

[5] K.D. Do, Formation tracking control of unicycle–type mobile robots with limited sensing ranges, IEEE Trans Control Syst Technology 16 (2008), 527–538.

[6] C. Ghedini and C.H.C. Ribeiro, Rethinking failure and attack tolerance assessment in complex networks, Physica A: Stat Mechanics its Appl 390 (November 2011), 4684–4691.

[7] C. Ghedini, C.H.C. Ribeiro, and L. Sabattini, Improving the fault tolerance of multi-robot networks through a combined control law strategy, Proc Int Workshop Resilient Networks Design Modeling (RNDM), Halmstadt, Sweden, September 2016.

[8] C. Ghedini, C. Secchi, C.H.C. Ribeiro, and L. Sabattini, Improving robustness in multi-robot networks, Proc IFAC Symp Robot Control (SYROCO), Salvador, Brazil, August 2015.

[9] C. Godsil and G. Royle, Algebraic graph theory, Springer, 2001.

[10] Z. He, S. Liu, and M. Zhan, Dynamical robustness analysis of weighted complex networks, Physica A: Stat Mechanics its Appl 392 (2013), 4181 – 4191.

[11] M.A. Hsieh, A. Cowley, V. Kumar, and C.J. Talyor, Maintaining network connectivity and performance in robot teams, J Field Robotics 25 (2008), 111–131.

[12] M. Ji and M. Egerstedt, Distributed coordination control of multiagent systems while preserving connectedness, IEEE Trans Robotics 23 (August 2007), 693–703.

[13] D. Koschützki, K.A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski, "Centrality indices," Network analysis, Springer, 2005, pp. 16–61.

[14] D. Lee, A. Franchi, H. Son, C. Ha, H. Bulthoff, and P. Robuffo Giordano, Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles, IEEE/ASME Trans Mechatronics 18 (August 2013), 1334–1345.

[15] M. Manzano, E. Calle, V. Torres-Padrosa, J. Segovia, and D. Harle, Endurance: A new robustness measure for complex networks under multiple failure scenarios, Comput Networks 57 (2013), 3641 – 3653.

[16] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie, Maintaining limited–range connectivity among second–order agents, Proc Am Control Conference, 2006, pp. 2134–2129.

[17] P. Robuffo Giordano, A. Franchi, C. Secchi, and H.H. Bülthoff, A passivity-based decentralized strategy for generalized connectivity maintenance, Int J Robotics Res 32 (2013), 299–323.

[18] L. Sabattini, N. Chopra, and C. Secchi, Decentralized connectivity maintenance for cooperative control of mobile robotic systems, Int J Robotics Res (SAGE) 32 (October 2013), 1411–1423.

[19] L. Sabattini, C. Secchi, and N. Chopra, Decentralized estimation and control for preserving the strong connectivity of directed graphs, IEEE Trans Cybernetics 45 (October 2015), 2273–2286.

[20] R. Soukieh, I. Shames, and B. Fidan, Obstacle avoidance of non-holonomic unicycle robots based on fluid mechanical modeling, 2009 Eur Control Conference (ECC), August 2009, pp. 3269–3274.

[21] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue, Robotics Automation Magazine, IEEE 19 (September 2012), 46–56.

[22] S. Wasserman, K. Faust, and D. Iacobucci, Social network analysis: Methods and applications (structural analysis in the social sciences), Cambridge University Press, November 1994.