

FINAL REPORT

FAPESP - PROC. NO. 2012/25058-9

Adaptive mechanisms for fault tolerance management in complex network topologies

Researcher: Cinara Guellner Ghedini Hita

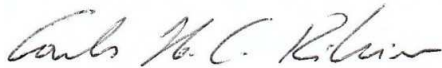
Supervisor: Carlos Henrique Costa Ribeiro

February, 2017

Adaptive mechanisms for fault tolerance management in complex network topologies



Researcher: Cinara Guellner Ghedini Hita



Supervisor: Carlos Henrique Costa Ribeiro

Final technical report proc. no. 2012/25058-9.

Date: 01/08/2015 to 30/11/2016.

São José dos Campos

February, 2017

Abstract

The availability of robust and power-efficient robotic devices boosts their use in a wide range of applications, most of them unfeasible in the recent past due to environmental restrictions or because they are hazard to humans. Nowadays, robots can support or perform missions of search and rescue, exploration, surveillance, and reconnaissance, or provide a communication infrastructure to clients when there is no network infrastructure available. In general, these applications require efficient and multi-objective teamwork. However, agents are prone to failure adding uncertainty and unpredictability. Thus, a robust topology regarding failures is an imperative requirement. In this sense, this research main goal is to design mechanisms for both detecting and mitigating vulnerable topological configurations. This report summarizes the activities and achievements of one-year project extension. During this period, a continuous-time model for a combined control law that aiming at detecting and mitigating vulnerable topological configurations while maintaining the network connectivity w.r.t. non-failure scenarios was developed. Its performance evaluation demonstrated that robots were able to properly respond to faults for different failure time distributions. Moreover, a collision avoidance and a coverage area improvement mechanisms were added to the combined control law for evaluating the concept of self-adaptive network topologies. The control law that combines the mechanisms for connectivity maintenance, robustness to failure improvement, collision avoidance, and coverage enhancing was evaluated through an experimental setup. Results showed the feasibility of having simultaneous controls acting to achieve topological network configurations according to the application requirements.

Contents

1	Introduction	4
2	Preliminaries	6
2.1	Background on network properties	6
2.2	System model	9
3	Connectivity maintenance and robustness to failure improvement control law	12
3.1	Simulations	14
3.1.1	Experimental setup	14
3.1.2	Simulation results	15
4	Failure time distribution evaluation	17
5	Coverage problem	21
5.1	Voronoi coverage	22
5.2	Unbounded Voronoi tessellations	25
5.3	Evaluation metrics	29
5.4	Simulations	31
5.4.1	Experimental setup	31
5.4.2	Simulation results	31
6	Conclusions and future works	34
7	Final remarks and summary of results	37
	References	39

1 Introduction

The availability of robust and power-efficient robotic devices boosts their use in a wide range of applications, most of them unfeasible in the recent past due to environmental restrictions or because they are hazard to humans. Nowadays, robots support or perform missions of search and rescue, exploration, surveillance and reconnaissance, which can take place in dangerous or extreme condition environments, after disaster events, in inhospitable settings or for military purposes. In such situations, great benefit can be achieved from the assistance of robotic systems [36, 39].

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 [21, 26, 6, 19, 1].

However, 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, 12, 18, 24]. 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 w.r.t. failures are mostly disregarded as technical issues.

Considering the problem relevance, we developed a model that intends to overcome the downside of the connectivity maintenance approaches [28, 29] and combining it with an approach that we proposed for enhancing the network robustness to failures based only on locally available information. This approach was introduced in [13] for a discrete-time system and re-

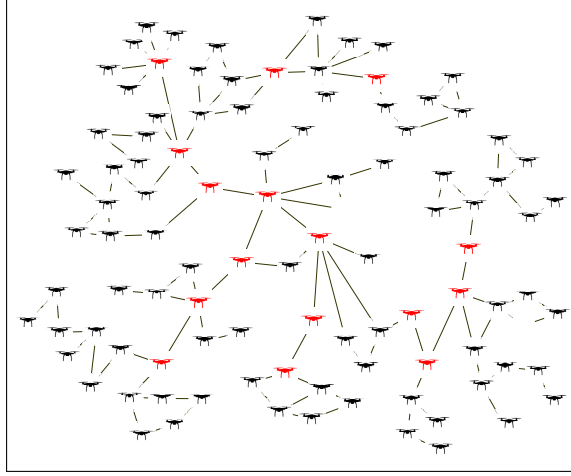


Figure 1: Scenario of a mobile robot network topology.

formulated for continuous-time single integrator robotic systems, and is thus combined with the connectivity maintenance and collision avoidance control strategies, detailed in [9]. The new model guarantees connectivity maintenance w.r.t. non-failure scenarios while both improving the network robustness to failure and ensuring collision avoidance of robots with obstacles and among them. Its performance was confronted with different parameterizations of gains in fault-free and fault-prone environments. The results demonstrate that the combined control law is able to increase the resilience of the system, by adapting the network topology to accommodate failures, postponing or even avoiding network disconnection.

Considering that the combined mechanism provides a more reliable network regarding connectivity maintenance and failure tolerance, there are several additional aspects to enhance the performance of multi-robot systems, such as power consumption, bandwidth, coverage, transmission power, etc. [37]. We have decided to address the coverage area problem because it is related to the topology formation and heading towards a solution for providing efficient network topologies. The main goal of coverage area in our context is to improve the capacity of a multi-robot network of sensing the environment, reducing the redundant monitoring areas.

The developed model is based on a local Voronoi coverage mechanism presented in [5]. This approach, however, considers a bounded environment, which is not a desirable feature regarding the scenario adopted. For overcoming this limitation, it was applied an approach proposed in [30], which focuses on presenting geographic data by means of Voronoi diagrams

without having to specify a bounding box. Its main advantages are that it is based on local information and allows to use the robot range as a parameter for reducing the size of bounding cells. The coverage area model was then combined with the previous model and evaluated by the same experimental setup. The results show that the coverage improvement mechanism was able to provide a more balanced network topology by improving the coverage area and the robustness to failure. Thus, we believe that this new combined model can be used as a reference for providing self-organized network solutions.

This report focuses on detailing these new functionalities, developed over the one-year extension of the postdoctoral project. For that, the necessary background on network properties and the system model are presented in Section 2. Section 3 details the continuous-time model for the algebraic connectivity and robustness to failure combined control law. In addition, an evaluation of the impact of failure time distributions on the combined strategy is presented in Section 4. Section 5 describes the coverage area approach. Conclusions and future works are discussed in Section 6. Section 7 outlines the main activities carried out and the achievements of the project.

2 Preliminaries

2.1 Background on network properties

Hereafter the quantities that are used for evaluating node and network connectivity and robustness to failures are defined.

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 node set and the edge set, respectively. Moreover, let $W \in \mathbb{R}^{N \times N}$ be the weight matrix: each element w_{ij} is a positive number if an edge exists between nodes i and j , zero otherwise. Since \mathcal{G} is undirected, then $w_{ij} = w_{ji}$. Now, 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 of \mathcal{G} , where k_i is the degree of the i -th node of \mathcal{G} , i.e., $k_i = \sum_{j=1}^N w_{ij}$. The (weighted) Laplacian matrix of \mathcal{G} is then defined as $\mathcal{L} = D - W$. As is well known from algebraic graph theory, the Laplacian matrix of an undirected graph exhibits some remarkable properties regarding its connectivity [16]. Let $\lambda_i, i = 1, \dots, N$ be the

eigenvalues of the Laplacian matrix, then:

- The eigenvalues are real, and can be ordered such that $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$.
- 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 [7]. 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:

$$S(\mathcal{G}) = \frac{n_S}{N}, \quad (1)$$

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, the **giant component** of a graph is used to denote its largest connected component, regardless of the network size.

Regarding robustness to failures, it is well-known that failures of nodes playing a central role in the network communication are likely to affect most its connectivity. In particular, referring to connectivity maintenance, the concept of *Betweenness Centrality (BC)* for ranking the network nodes is considered [38]. 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 :

$$BC(i) = \sum_{j < l} \frac{g_{jl}(i)}{g_{jl}}. \quad (2)$$

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 highest value of BC) to the *less central* (i.e., the node with lowest value of BC). Hence, let $[v_1, \dots, v_N]$ be the list of nodes ordered by descending value of BC . Thus, the *robustness level* is introduced as follows.

Definition 2.1 (Robustness level [11]). *Consider a graph \mathcal{G} with N nodes, and let $[v_1, \dots, v_N]$ be the list of nodes ordered by descending value of BC . Let $\varphi < N$ be the minimum index $i \in [1, \dots, N]$ such that, removing nodes $[v_1, \dots, v_i]$ leads to disconnecting the graph, that is, the graph including only nodes $[v_{\varphi+1}, \dots, v_N]$ is disconnected. Then, the robustness level of \mathcal{G} is defined as:*

$$\Theta(\mathcal{G}) = \frac{\varphi}{N}. \quad (3)$$

The robustness level defines the fraction of central nodes that need to be removed from the network to obtain a disconnected network. Small values of $\Theta(\mathcal{G})$ imply that a small fraction of node failures may fragment the network. Therefore, increasing this value means increasing the network robustness to failures. Notice that $\Theta(\mathcal{G})$ is only an estimate of how far the network is from getting disconnected w.r.t. 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 $\Theta(\mathcal{G})$.

From a local perspective, a heuristic for estimating the magnitude of the topological vulnerability of a node by means of information acquired from its 1-hop and 2-hops neighbors is proposed in [11]. 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

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

Thus, 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 w.r.t. connectivity, and **the vulnerability level of a node regarding failures** is given by $P_\theta(v) \in (0, 1)$:

$$P_\theta(v) = \frac{|Path_\beta(v)|}{|\Pi(v)|}. \quad (4)$$

where $|\Pi(v)|$ is the number of v 's 1-hop and 2-hops neighbors, and $|Path_\beta(v)|$ is the number of nodes that are exactly at 2-hops from node v and relying on at most β 2-hops paths to communicate with v .

2.2 System model

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's state be its position $p_i \in \mathbb{R}^m$, and let $p = [p_1^T \dots p_N^T]^T \in \mathbb{R}^{N \times m}$ 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

$$\dot{p}_i = u_i, \quad (5)$$

where $u_i \in \mathbb{R}^m$ is a control input¹. In order to guarantee the connectivity of \mathcal{G} , [28] 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} are defined as follows:

$$w_{ij} = \begin{cases} e^{-(\|p_i - p_j\|^2)/(2\sigma^2)} & \text{if } \|p_i - p_j\| \leq R \\ 0 & \text{otherwise.} \end{cases} \quad \text{with } e^{-(R^2)/(2\sigma^2)} = \Delta \quad (6)$$

where Δ is a small predefined threshold². Define now $\epsilon > 0$ to be the desired lower-bound for the value of λ . The control strategy is then designed to ensure that the value λ never goes below ϵ . As in [28], the following *energy function* can then be utilized for generating the decentralized connectivity maintenance control strategy:

$$V(\lambda) = \begin{cases} \coth(\lambda - \epsilon) & \text{if } \lambda > \epsilon \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The control design drives the robots to perform a gradient descent of $V(\cdot)$, in order to ensure connectivity maintenance. Namely, considering the dynamics of the system introduced in (5), the control law is defined as follows:

$$u_i = u_i^c = -\frac{\partial V(\lambda)}{\partial p_i} = -\frac{\partial V(\lambda)}{\partial \lambda} \frac{\partial \lambda}{\partial p_i}. \quad (8)$$

The connectivity maintenance framework can be enhanced to consider additional objectives. In particular, as shown in [27], the concept of *generalized connectivity* can be utilized for simultaneously guaranteeing **connectivity maintenance and collision avoidance** with environmental obstacles and among the robots. This is achieved considering the following *generalized edge*

¹It 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 [34], and UAVs [22].

²This definition of the edge-weights introduces a discontinuity in the control action, that can be avoided introducing a smooth bump function, as in [8]. However, from an implementation viewpoint, the effect of the discontinuity can be made negligible by defining the threshold Δ sufficiently small.

weights:

$$\omega_{ij} = w_{ij}\gamma_{ij}, \quad (9)$$

$\forall i, j = 1, \dots, N$. In particular, the edge weights w_{ij} represent the standard connectivity property. The multiplicative coefficients γ_{ij} represent the *collision avoidance edge weights*:

Definition 2.2. *The collision avoidance edge weights γ_{ij} exhibits the following properties, $\forall i, j = 1, \dots, N$:*

$$(P1) \quad \gamma_{ij} = \gamma_{ij}(\|p_i - p_j\|) \geq 0.$$

$$(P2) \quad \gamma_{ij} = 0 \text{ if } \|p_i - p_j\| = 0, \text{ and } \gamma_{ij} = 1 \text{ if } \|p_i - p_j\| \geq d_s.$$

$$(P3) \quad \gamma_{ij}(d) \text{ is non-decreasing w.r.t. its argument } d.$$

The parameter $d_s > 0$ represents the *safety distance*: if the distance between two robots is larger than d_s , then the collision avoidance action is not necessary. As shown in [27], the same formalism can be exploited for avoiding collisions with environmental obstacles as well.

Utilizing the generalized edge weights ω_{ij} defined in (9), the *generalized Laplacian matrix* \mathcal{L}^G , whose second smallest eigenvalue φ represents the *generalized connectivity* of the graph can be computed. As shown in [27], guaranteeing positiveness of the generalized connectivity φ simultaneously guarantees maintenance of the algebraic connectivity (i.e., it ensures that λ remains positive) and collision avoidance. This can then be achieved using the control law (8) replacing λ with φ , namely

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

Since φ and its gradient are global quantities, the proposed control law is centralized. Decentralized implementation can be achieved replacing φ and its gradient with their estimates, computed by each robot in a decentralized manner applying the procedure proposed in [28].

This methodology does not consider the fact that robots can unexpectedly fail, thus stopping their activity due to mechanical, electrical or software issues. As described in the Introduction, it is necessary to reduce the effects of robot failures on the overall network connectivity, avoiding

vulnerable topological configurations.

3 Connectivity maintenance and robustness to failure improvement control law

This section describes the unified model that aims at improving the robustness of networks to failures while, in the absence of failures, maintaining connectivity and avoiding collisions. In particular, considering the dynamics introduced in (5), we define the following control law:

$$u_i = \sigma u_i^c + \psi u_i^r, \quad (11)$$

where u_i^c is the generalized connectivity maintenance control law introduced in (10), and u_i^r is the additional control law that will be hereafter defined for improving robustness to failures. Moreover, $\sigma, \psi \geq 0$ are design parameters that represent control gains. Setting either σ or ψ equal to zero leads to removing the effect of one of the control laws. Conversely, if both parameters are greater than zero, both control actions are simultaneously active.

Based on the vulnerability level definition, given in (4), the purpose of the control strategy is to increase the number of links of a potentially vulnerable node i towards its 2-hop neighbors that are in $Path_\beta(i)$, for a given value of β . Hence, define $x_\beta^i \in \mathbb{R}^m$ as the barycenter of the positions of the robots in $Path_\beta(i)$, namely

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

Considering the dynamics of the system introduced in (5), the control law is defined as follows:

$$u_i^r = \xi_i \frac{x_\beta^i - p_i}{\|x_\beta^i - p_i\|} \alpha(t), \quad (13)$$

where $\alpha(t) \in \mathbb{R}$ is the linear velocity of the robots³.

³Pathological situations exist in which (13) is not well defined, namely when $p_i = x_\beta^i$. However, this corresponds to the case where the i -th robot is in the barycenter of its weakly connected 2-hop neighbors: hence, in practice, this never happens when a robot detects itself as vulnerable.

The parameter ξ_i is introduced to take into account the vulnerability state of a node i , i.e., $\xi_i = 1$ if node i identifies itself as vulnerable or $\xi_i = 0$ otherwise. As we aim at setting as vulnerable those robots i exhibiting high values for $P_\theta(i)$, ξ_i is defined as follows:

$$\xi_i = \begin{cases} 1 & \text{if } P_\theta(i) > r \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

where $r \in (0, 1)$ is a random number drawn from a uniform distribution. Namely, if $P_\theta(i) > r$, then the i -th robot considers itself as vulnerable. It is worth noting that (4) provides a decentralized methodology for each robot to evaluate its vulnerability level.

Summarizing, this control law drives vulnerable robots towards the barycenter of the positions of robots in their $Path_\beta$, thus decreasing their distance to those robots and eventually creating new edges in the communication graph.

The control law u_i^c was proven in [28, 27] to guarantee positiveness of the generalized connectivity in a disturbance free environment. The following theorem extends these results considering the presence of the additional robustness improvement control law u_i^r .

Theorem 1. *Consider the dynamical system described by (5), and the control laws described in (10), (11) and (13). Then, if the initial value of $\varphi(t)$, namely $\varphi(0)$, is greater than ϵ , then the value of $\varphi(t)$ will remain positive, as the system evolves, thus implying algebraic connectivity maintenance and collision avoidance.*

Proof. It is possible to show that, under the proposed control law, for constant values of P_θ , the energy function $V(\varphi(p))$ in (7) does not increase over time. As a consequence, it is possible to conclude that the generalized connectivity φ remains greater than zero, as the system evolves. Considering the definition of the generalized edge weights ω_{ij} in (9), this implies that the algebraic connectivity λ will remain positive, while avoiding collisions. This result can be extended to the case where P_θ is time-varying, assuming that variations are sufficiently slow. The proof is analogous to that of [28]. \square

3.1 Simulations

3.1.1 Experimental setup

The proposed 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 initial network topologies were generated over a bounded area of size \mathcal{A}^2 in \mathbb{R}^2 , through a random positioning of N robots, connected according to the communication radius R . For this simulation we consider $N = 20$, $\mathcal{A} = 50$ and $R = 16$.

Given an initial network topology and the combined control law, each node is supposed to move in order to increase the robustness of the network, while keeping the global network connectivity and avoiding collision with randomly placed obstacles. These requirements must be fulfilled despite robot failures. For evaluation purposes, at specific times a disturbance is introduced into the network by removing its most central node according to the updated BC ranking, as defined in (2). The experimental setup considers a set of randomly generated failure times, distributed during the total simulation time.

For emphasizing the importance of the combined control law strategy, its performance was assessed by setting different gain combinations for σ and ψ . This allows a more meaningful analysis considering that the gains calibrate the weight of each control law. For instance, the higher the connectivity coefficient (σ), the higher the connectivity control law effect on keeping a high value of the algebraic connectivity, and likewise for the robustness coefficient (ψ). The considered gain values were $\{0, 0.1, 0.5, 1\}$. Notice that, as demonstrated in Theorem 1, any combination of positive gains leads to the desired behavior.

Regarding the specific control law parametrization, we used $\epsilon = 0.25$, $\beta = 1$ and $\alpha = 0.25R$. Moreover, for assessing the performance of the proposed methodology in a statistically sound manner, the same experiment was carried out for 100 different network topologies. Thus, the results presented here represent an average of 100 networks. The total simulation time was $t = 80$ seconds with the vulnerability level estimation and network properties evaluation being performed at every 1 second.

3.1.2 Simulation results

In this section we discuss the impact of failures on the network properties according to several gain combinations. Figures 2, 3 and 4 present the results for the robustness level, the algebraic connectivity and the size of the giant component, respectively. Each picture depicts a specific chart for each value for the ψ gain and its combination with different values for the σ gain. The vertical axis represents the initial scores for the respective property and its evolution when exposed to continuous failures through the fraction f of nodes removed from the network (horizontal axis).

The robustness level evolution, illustrated in Figure 2, supports the point that it is not suitably relying on the algebraic connectivity as a means of keeping the network connectivity in scenarios where the robots are subject to failures. Notice that the robustness to failure level is not significantly affected by the σ gain setting. On the other hand, the robustness level increases according to the ψ gain value. Considering that $\psi \geq 0.5$ implies that the networks were able to accommodate failures, an intermediate value for ψ may lead to a satisfactory robustness to failure level.

As expected, considering that networks become fragmented after only a few node failures, the algebraic connectivity property quickly approaches zero for small values of ψ . In contrast to the robustness level evolution, the algebraic connectivity exhibits a slight improvement as the σ value increases, which is also expected considering that the mechanism for controlling the network connectivity is based on it. This behavior also corroborates with the claim that the algebraic connectivity property is not suitable to support mechanisms for producing more robust networks.

Let us now consider the size of the giant component under failures (Figure 4). The results demonstrate that, regardless of failures, the action of the robustness control law kept most of the nodes into the giant component. Notice that, for $\psi = 0.5$, the produced giant component is smaller for $\sigma = 1$ than for $\sigma = 0.5$ and $\sigma = 0$. According to the robustness level definition, it means that more nodes can fail, but if they fail, the network will achieve a smaller giant component.

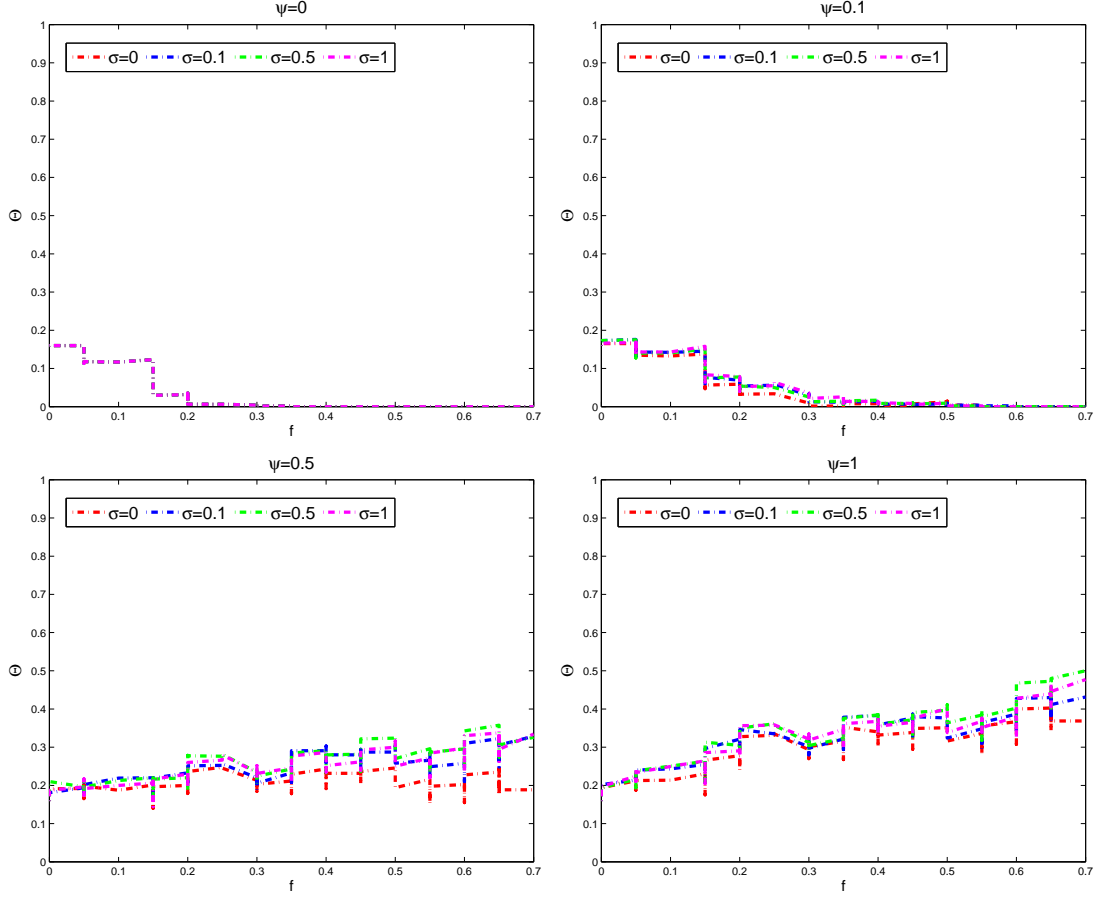


Figure 2: Control law performance - The robustness level (Θ)

For demonstrating the fragility of random networks to failures of elements, Figure 5(a) exhibits snapshots of a network during the simulation process without the proposed control law (i.e., $\psi = 0$ and $\sigma = 0$). These snapshots correspond to simulation time $t = 0$, $t = 20$, $t = 50$ and $t = 80$, depicted by black, blue, green and red nodes, respectively. Notice that at $t = 20$ the network fragmented into two clusters, and as the simulation evolves the number of clusters increases. Different failure distribution might accelerate or delay the network fragmentation, however, cascading failures or a high vulnerable topological configuration are surely harmful to the network connectivity and, as a consequence, to its operation.

Consider now the performance of the combined control law for the same scenario ($\psi = 1$ and $\sigma = 1$), illustrated in Figure 5(b). The resulting network topology is still connected at the end of the simulation despite failures of central elements, emphasizing the effectiveness of the

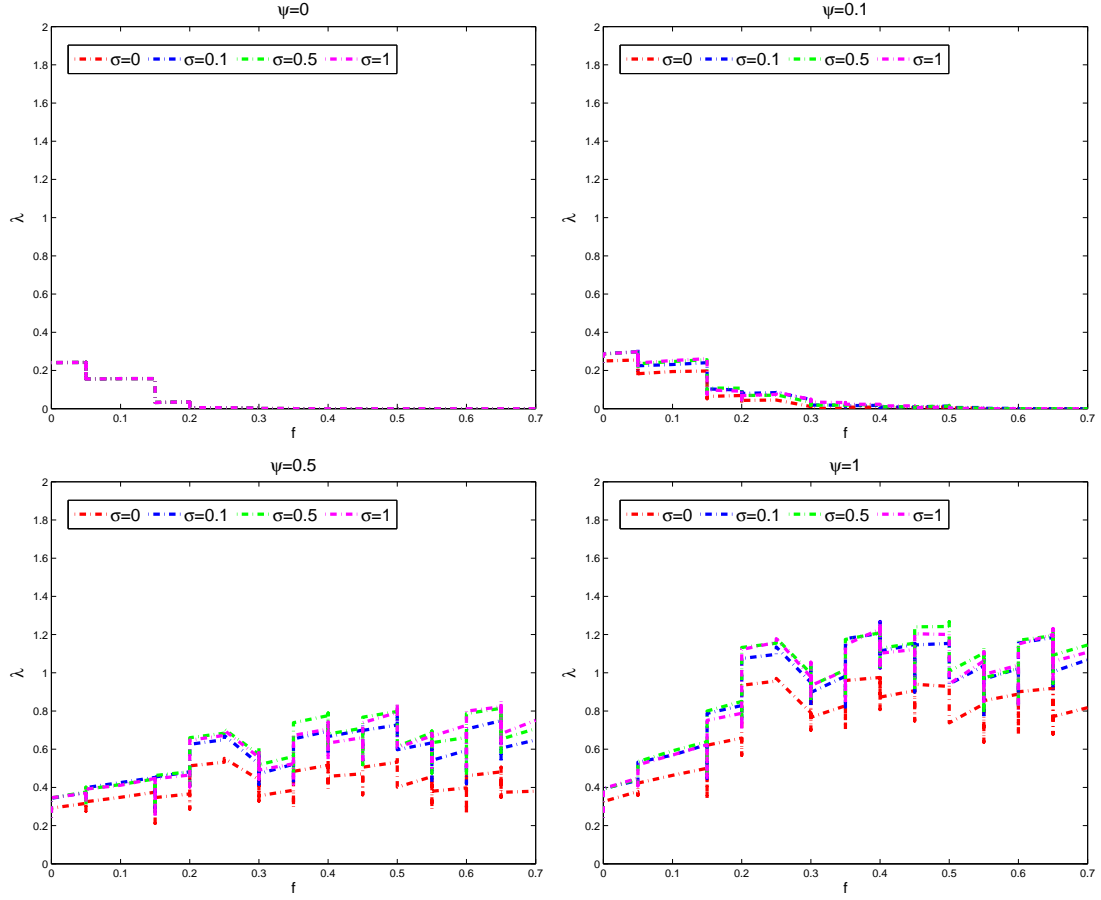


Figure 3: Control law performance - The algebraic connectivity (λ)

combined control law to produce a more resilient network. As already mentioned, the connectivity can not always be guaranteed, but postponing the network fragmentation or increasing the number of elements in the giant component are significant achievements for those applications where connectivity is crucial. Some additional examples can be freely viewed online on <https://youtu.be/ueo7nYEAm24>.

4 Failure time distribution evaluation

In this section, the impact of failure time distribution on the combined control law performance is evaluated. The results presented in Section 3.1.2 considers a uniform failure time distribution. For this experiment, the times in which failures will occur are given by a time vector

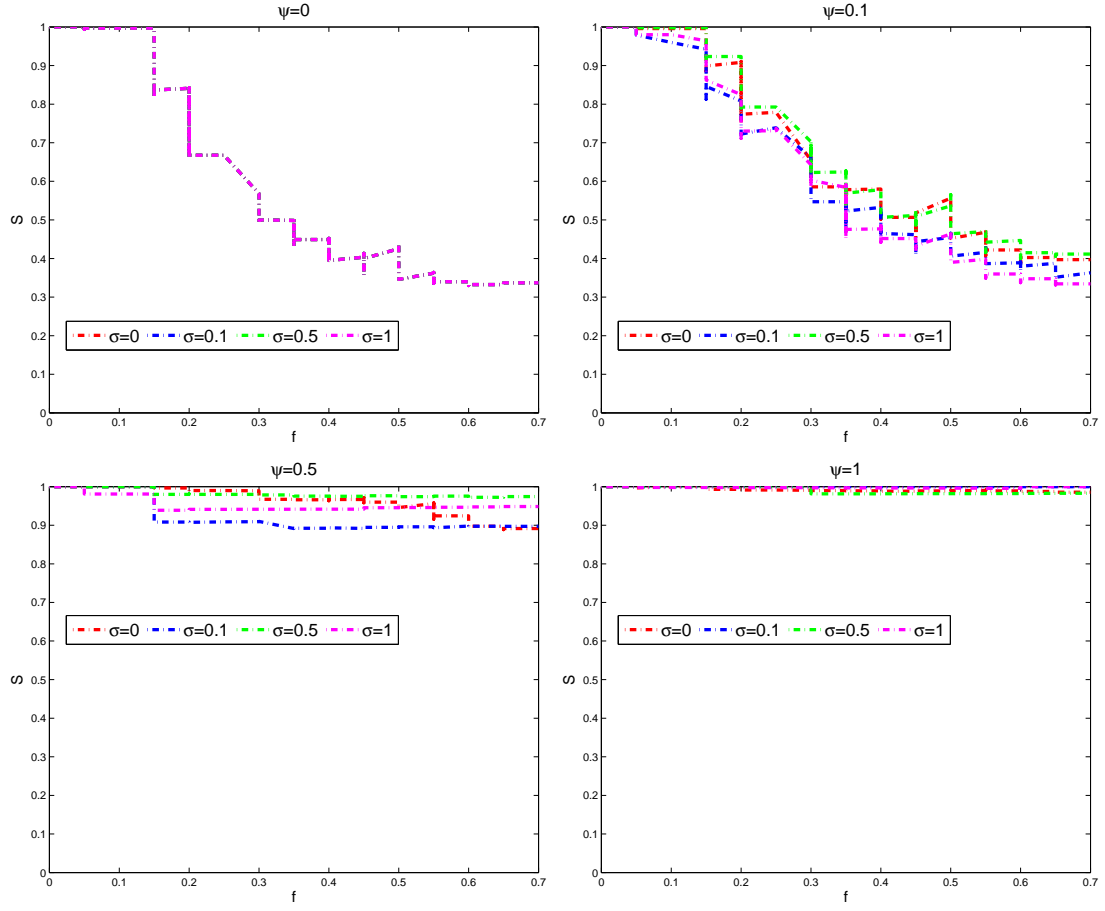


Figure 4: Control law performance - The Giant Component (S)

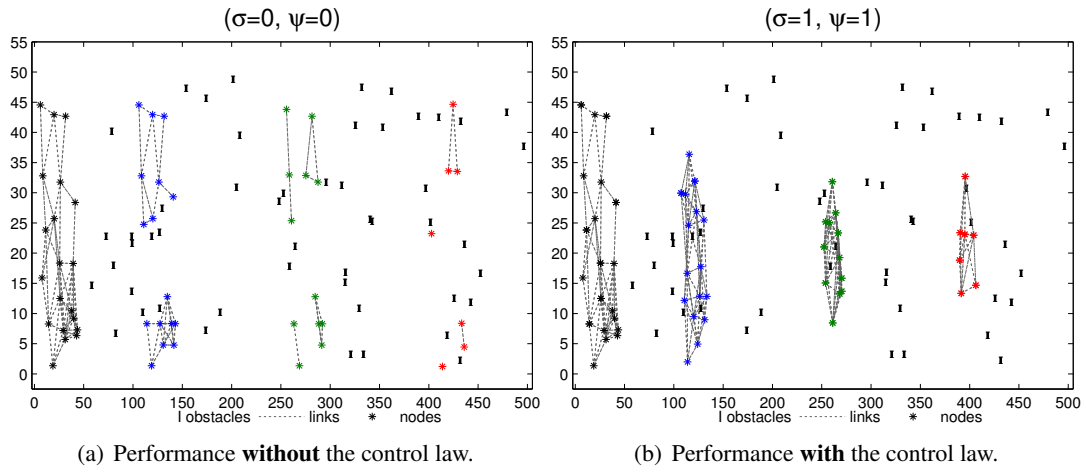


Figure 5: Snapshots of a network topology at $t = 0$ (black), $t = 20$ (blue), $t = 50$ (green) and $t = 80$ (red) simulation times.

$f_t = (f_{t1}, f_{t2} \dots f_{tn})$, where n is the number of failures, generated according to the procedure described as 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 ($t_f(i) \geq 30$ and $t_f(i) \leq 50$). 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, the failure time ranges were defined 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 1 shows the average failure time distributions generated for each of the specified ranges.

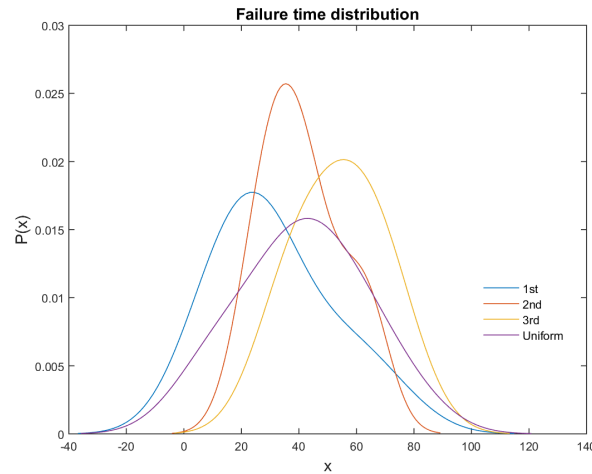


Figure 6: Failure time distributions.

Table 1 presents the setup parameters concerning the distribution experiment. Parameteriza-

tion for u_i^c and u_i^r models are the same applied to the previous experiment (see Section 3.1.1). We ran the simulation considering the combined mechanism as both active ($\sigma=2, \psi=1$) and non-active ($\sigma=0, \psi=0$).

Table 1: 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 7. 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 7(a)). However, the same pattern does not apply to scenarios in which the combined control law is active. As demonstrated in Figure 7(b), despite most failures concentrating at specific ranges, the combined control law was able to suppress their impact.

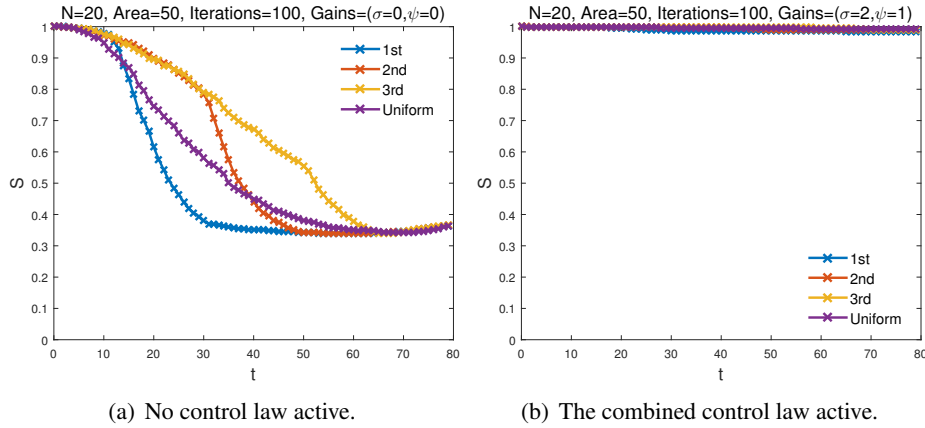


Figure 7: The giant component (S) performance under different time failure distributions.

This feature is reinforced by the robustness level evolution (Figure 8(b)): 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 8(a)).

The actuation of the robustness to failure mechanism can be clearly visualized in Figure 9(b). 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

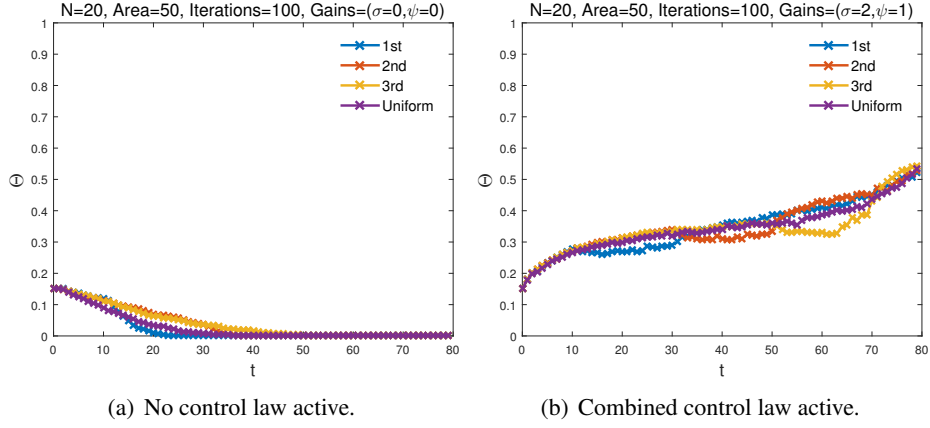


Figure 8: The robustness level (Θ) performance under different time failure distributions.

(4) was effective in providing means for nodes 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 3.

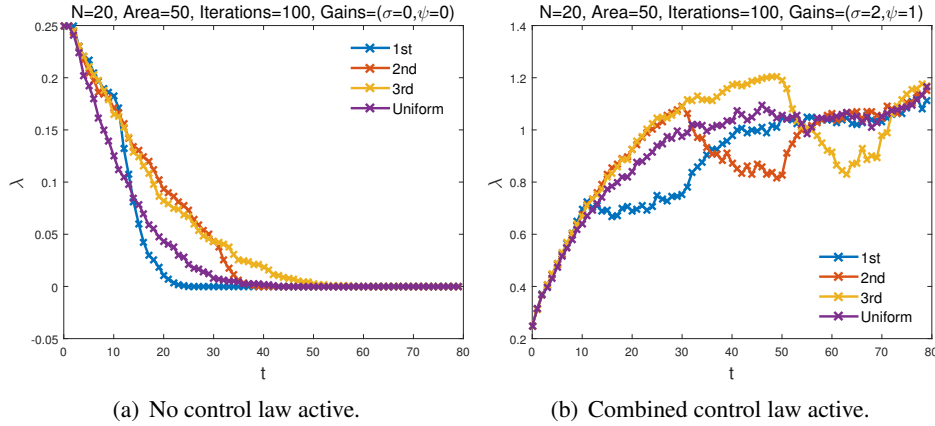


Figure 9: The algebraic connectivity (λ) performance under different time failure distributions.

5 Coverage problem

The coverage mechanism in the context of this work aims at improving the capacity of a multi-robot network of sensing the environment, reducing the redundant monitoring areas and avoiding holes in the network. Coverage can be defined as how well or to how much extent each point of a deployed network is under the vigilance of a sensor node [31].

In general, robots should be distributed around the environment in a way that the overlap

regions, i.e., those spots that are monitored by more than one node, are optimized to improve the total area that can be monitored by the same number of robots. On the other hand, ad hoc network formation or even the presence of obstacles can generate holes in the sensing area, which may lead to routing failure and degrade the quality of service [3, 17]. Figure 10 illustrates holes and redundant monitoring area (overlap areas).

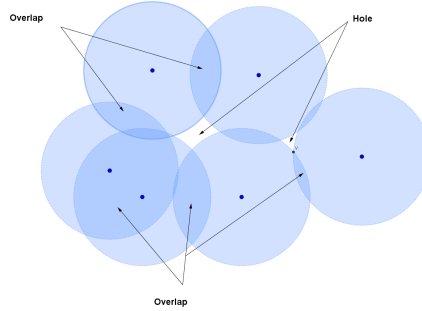


Figure 10: Hole and overlap areas illustration.

There are several approaches addressing different coverage problems [20, 3, 32]. The optimization function is related to the application features and the system requirements [31]. For instance, if the deployment phase is based on a predefined or a random strategy, if there are others optimization goals (e.g., connectivity, energy saving), if the sensing area is known in advance, etc. As a consequence, different techniques can be applied. In this sense, Voronoi tessellation is a widespread technique that supports several context of application approaches [3, 33, 31, 5, 35]. It is commonly applied not only for improving the network coverage but also for evaluating the network coverage level, which is not a trivial task. The Voronoi coverage is discussed as follows.

5.1 Voronoi coverage

A Voronoi tessellation is a subdivision of a plane into cells based on the proximity of a set of points (named sites). Let $P = [p_1, p_2, \dots, p_n]$ be a set of points in the plane. Define $V(i)$, the Voronoi cell for p_i , to be the set of points q in the plane that are closer to p_i than to any other

site. Thus, the Voronoi cell for p_i is defined by

$$V(i) = \{q \mid \|p_i q\| < \|p_j q\|, \forall j \neq i\}, \quad (15)$$

where $\|pq\|$ denotes the Euclidean distance between points p and q [25].

It is known that a cost function representing the sensing cost of a network is locally minimized if each point is positioned at the centroid of its Voronoi cell [32]. For illustration, Figure 11(a) shows the Voronoi tessellation of a random network – blue dots represent the robot positions –; Figure 11(b) the same Voronoi tessellation but with each robot positioned at the center of its Voronoi cell; and Figure 11(c) the new Voronoi tessellation generated after two iteration of the following process: (1) the centroid of each Voronoi cell is computed, (2) each robot is positioned at the centroid of its correspondent cell, and (3) a new Voronoi diagram is generated. Notice that this process generated a network with robots more uniformly distributed over the area.

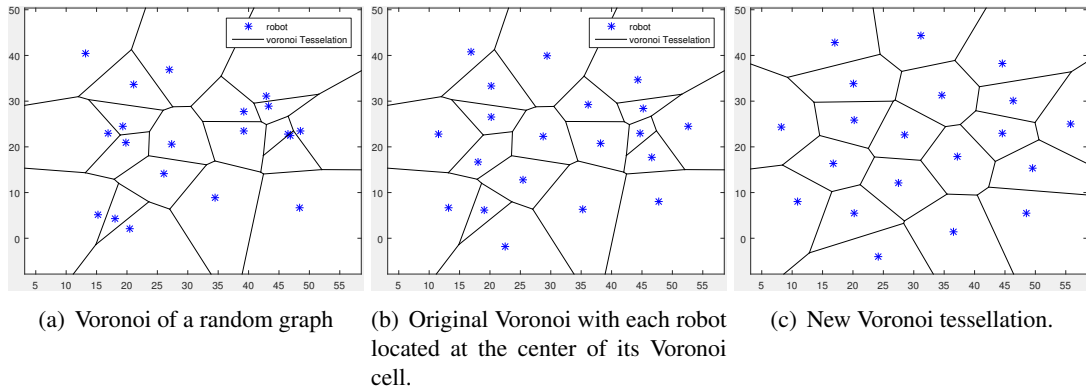


Figure 11: Voronoi tessellation examples.

According to the requirements demanded by the defined scenario features, such as using local information, the strategy for modeling the Voronoi coverage is based on a local mechanism presented in [5], described as follows. Let $V(i)$ be the Voronoi cell corresponding to a robot located at p_i . A team of n robots at positions $P = [p_i]_{i=1}^n \in \mathbb{R}^{Nn}$ navigate in a bounded polygonal environment $\Omega \Rightarrow \mathbb{R}^N$, where Ω is a closed set with the boundary $\partial\Omega$. Given a

positive density function $\phi : \Omega \rightarrow \mathbb{R} > 0$, the centroid of $V(i)$ is defined as

$$C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} x \phi(x) dx, \quad (16)$$

where M_{V_i} is the cell mass [5]:

$$M_{V_i} = \int_{V_i} \phi(x) dx. \quad (17)$$

An optimal distribution of generating points implies that each of them is at the center of mass of its cell, i.e., $p_i = C_{V_i}, \forall i \in \{1, \dots, n\}$. This centroidal Voronoi tessellation (CVT) is a minimum-energy configuration in the sense that it minimizes the distance between points:

$$\mathcal{H}(P) = \sum_{i=1}^n \mathcal{H}(p_i) = \sum_{i=1}^n \int_{V_i} f(D(p_i, x)) \phi(x) dx, \quad (18)$$

where $D(\cdot)$ is the distance between a point p_i and a location x in Ω , and $\phi(\cdot)$ is the density function that describes the importance of different areas in Ω [5]. Considering the euclidean distance $D(p_i, x) = \|x - p_i\|$ as the distance function in (18):

$$\frac{\partial \mathcal{H}(p_i)}{\partial(p_i)} = -M_{V_i}(C_{V_i} - p_i) = 0, \quad (19)$$

the local minima is reached with a CVT since $C_{V_i} = p_i$ implies $\frac{\partial \mathcal{H}}{\partial(p_i)} = 0, \forall i \in \{1 \dots n\}$.

The Voronoi coverage presented in [5] is based on the Lloyds algorithm [23], a well-stabilished method for generating a centroidal Voronoi tessellation. It encompasses:

1. Create the Voronoi diagram for generating points P .
2. Compute the centroid $[C_{V_i}]_{i=1}^n$ of each Voronoi cell.
3. Set the new target position of each robot as the centroid of its cell.

This process iterates until the computed centroids and the generating points are at the same positions. From the proportional control law (19) the simple first-order dynamics can be derived:

$$u_i = \dot{p}_i = -\frac{k}{M_{V_i}} \frac{\partial \mathcal{H}(p_i)}{\partial(p_i)} = k(C_{V_i} - p_i). \quad (20)$$

However, regarding the scenario adopted, the goal is not for robots to find an optimal position and remain there, but adaptively explore an unknown and unbounded environment in a way that the total sensing area can be improved as much as possible in accordance with other active control laws. Thus, (13) is redefined:

$$u_i = \sigma u_i^c + \psi u_i^r + \zeta u_i^v, \quad (21)$$

where u_i^v represents the coverage controller (20) and ζ its respective gain. The procedure for locally computing the Voronoi cells assumes that each robot knows its own position p_i and its neighbor positions $p_{\mathcal{N}_i} = \{p_j, \forall j \in \mathcal{G} : i, j \in \mathcal{E}(\mathcal{G})\}$. Thus, each robot i :

- Acquires the position of its neighbors $p_{\mathcal{N}_i}$.
- Computes its local Voronoi ($V(i)$ Eq. (15)).
- Computes the mass of the centroid of its Voronoi cell C_{V_i} (16).
- Moves toward the mass of its cell centroid C_{V_i} .

This procedure iterates until a centroidal Voronoi tessellation is achieved or as it is convenient for the application. Here, as the robots are exploring an environment with obstacles, the Voronoi tessellation and the mass of the centroid are recomputed at each t seconds, setting according to the application requirements. If a robot i is at the centroidal position of its Voronoi cell, the weight for u_i^v is zero, i.e., the i robot does not need to improve its position regarding coverage.

This approach imposes some constraints to the scenario adopted here: the procedure for creating a Voronoi tessellation generates some unbounded cells. On the other hand, for computing the mass of the centroid for a cell it needs to be bounded. Thus, it is necessary to define a procedure to delimitate the unbounded cells. This issue is addressed as follows.

5.2 Unbounded Voronoi tessellations

Considering that the coverage approach is based on the generation of each cell centroid, the main issue of using the Voronoi tessellation as a means of improving the network coverage area

is the unbounded cells. Figure 12 presents the corresponding Voronoi tessellation of two random networks.

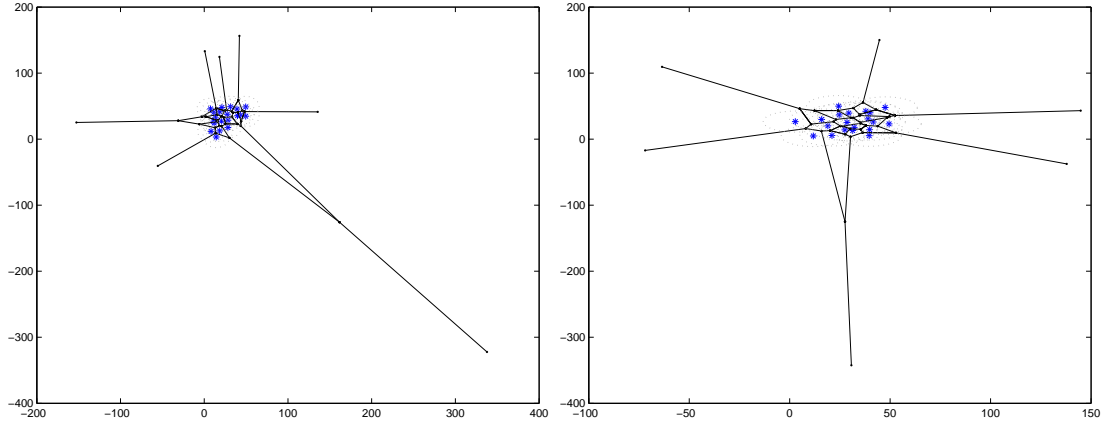


Figure 12: Examples of Voronoi tessellations.

The technique applied for bounding cells can impact the mechanism performance. Figure 13 exemplifies the output of three techniques to bound Voronoi cells of networks presented in Figure 12. The most straightforward manner, represented by magenta lines, is using the positions of robots at the lower and higher positions, regarding x and y coordinates in the plane, as a reference to delimit the environment. Given the reference points, the robot's range (represented by dashed lines) is taken into account to defined the environment boundaries. The main problems with this technique are that its assumes global knowledge of the environment, and some cells can become too large, thus generating biased cell centroids.

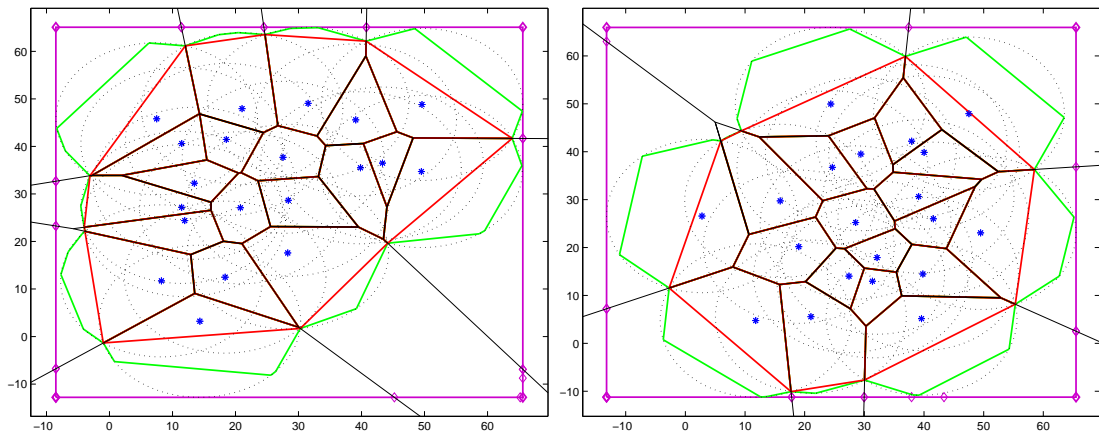


Figure 13: Examples of bounded Voronoi tessellations.

Another technique, that considers local information for computing the cutting points of each cell, takes the two unbounded edges and a circle having the robot position as its center and its range as the radius. The intersection point of each edge and the circle defines the cutting points. The resulting bounded Voronoi cells are represented by red lines in Figure 13. Notice that, this approach can generate small cells, that in consequence can drive robots toward the network center instead of trying to spread them around the environment.

For overcoming the drawbacks of both techniques presented above, an approach proposed in [30] was adopted. Its performance is illustrated by green lines. Notice that the size and shape of the new Voronoi cells are more balanced and in accordance with the robot range.

This approach focuses on presenting geographic data through Voronoi diagrams without having to specify a bounding box. However, the same procedure can be applied for bounding a Voronoi diagram with the advantage of using the robot range for reducing the size or bounding cells, whereas the technique relies on defining a maximum size for a cell. For exemplifying the procedure, Figure 14 illustrates a Voronoi cell for a generating point and a circle that defines the cell limits. All possible state classifications and actions for an edge can be visualized in this example. Each edge may:

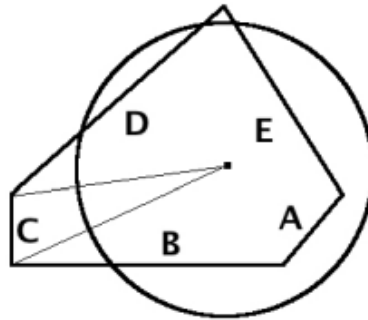


Figure 14: Mapping edges in a Voronoi Cell [30].

- lay completely inside the bounding circle (edge A): it must not change;
- lay completely outside the bounding circle (edge C): it must be replaced by its corresponding circle;
- have one intersection with the bounding circle (edges B and E): the part laying inside the

circle remains unchanged, the rest is replaced by its corresponding circle part;

- have two intersections with the bounding circle (edge D): it part laying inside the circle remains unchanged, the other two parts are replaced by their corresponding circle part.

The first step is to verify if a cell lies inside the bounding circle by computing the distance of each of its vertices to the circle center (in this application, the robot position) and comparing it with the circle radius:

$$distance_{(P, Center)} = \sqrt{(P_x - Center_x)^2 + (P_y - Center_y)^2}, \quad (22)$$

where P_x and P_y are the coordinates of each Voronoi cell point and $Center$ is the center point of the circle.

For those vertices that are completely outside of the bounding circle (edge C - Figure 14), its x and y coordinates are mapped to the circle point, i.e., the distance from the new point to the $Center$ is equal to the predefined radius:

$$NewP_x = Center_x + (P_x - Center_x) * Radius / distance(P, Center), \quad (23)$$

$$NewP_y = Center_y + (P_y - Center_y) * Radius / distance(P, Center). \quad (24)$$

For cutting the edges intersecting the circle (edges B, D and E - Figure 14), a straight line corresponding to an edge is computed:

$$\begin{aligned} y &= a * x + b \text{ when } x_1 \neq x_2, a = (y_2 - y_1) / (x_2 - x_1) \text{ and } b = (y_1 * x_2 - y_2 * x_1) / (x_2 - x_1) \\ x &= a * y + b \text{ when } x_1 = x_2, a = 0 \text{ and } b = x_1, \end{aligned} \quad (25)$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the two vertices. Then, the definition of bounding circle is given by:

$$y = Center_y \pm \sqrt{Radius^2 - (x - Center_x)^2} \text{ if } Radius^2 \geq (x - Center_x)^2. \quad (26)$$

The equations (25) and (26) are then combined to find the intersections:

$$Center_y \pm \sqrt{Radius^2 - (x - Center_x)^2} = a * x + b. \quad (27)$$

Solving the equation (27) in order to achieve a quadratic equation of the form $A*x^2+B*x+C=0$:

$$\begin{aligned} A &= -1 - a^2 \\ B &= 2 * Center_x - 2 * a * (b - Center_y) \\ C &= Radius^2 - Center_x^2 - (b - Center_y)^2, \end{aligned} \quad (28)$$

the values of x are given by the general quadratic equation $x = -B \pm \sqrt{B^2 - 4 * A * C}$, and the values of y by equation (25). Thus, x and y determine the new cell boundaries. The computational complexity is estimated as $\mathcal{O}(n)$, where n is the number of edges in the Voronoi diagram [30].

5.3 Evaluation metrics

For evaluating the decreasing of the redundant area monitored by the robots, the number of spots covered by each robot is considered. For that, sensing points ($Spoints$) are uniformly distributed around the environment – red points in Figure 15. Then, the overlap area – sensing points that are served for more than one robot – is computed:

$$Spoints_{overlap} = ((\sum_{i=1}^N \#Spoints_c(i)) - \#Spoints_c) / ((\#Spoints - 1) * N), \quad (29)$$

where $\#Spoints_c(i)$ is the number of sensing points served by robot i , $\#Spoints_c$ is the number of sensing points served by at least one robot, $\#Spoints$ is the number of sensing points, and N is the number of robots.

The capacity of the coverage mechanism to spread robots into the environment is given by

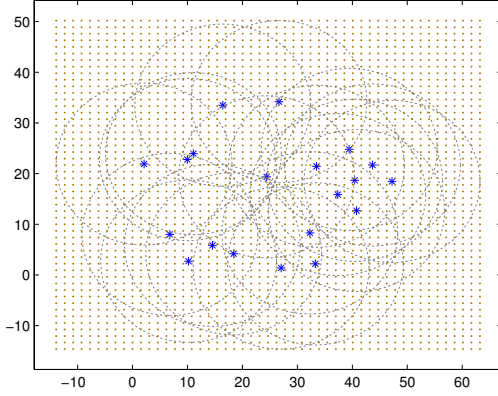


Figure 15: Sensing Points.

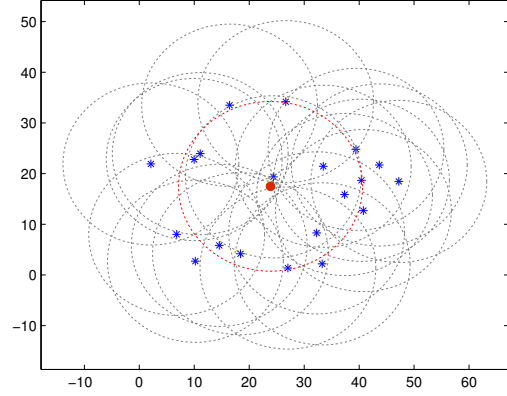


Figure 16: Central point.

the average distance from each robot i to a central point in the network:

$$avgDist(\mathcal{G}) = \frac{1}{N} \sum_{i=1}^N distance(p_i, Center(\mathcal{G})), \quad (30)$$

where $distance(p_i, Center(\mathcal{G}))$ is the Euclidean distance and p_i is the robot's i position. The central point is defined fitting the robot positions into the plane to a circle, and then taking the x and y coordinates of the circle center ($Center(\mathcal{G})$) (see Figure 16).

Although these metrics were able to demonstrate the coverage area evolution, a more precise evaluation was necessary. In this direction, a technique proposed by Kashi et.al. [33] was applied. The approach details a protocol for estimating the coverage rate. It consists of identify the boundaries of each node's neighboring network. Each boundary is then classified as the network outline, holes, and stains. The area comprising each type of boundary is then computed considering the polygon made by nodes inside each boundary. The total area is simply the sum of each boundary type, where holes are negative values and areas of stains and the network boundary are positive ones. This technique involves several concepts and equations that are not detailed here, (see [33]).

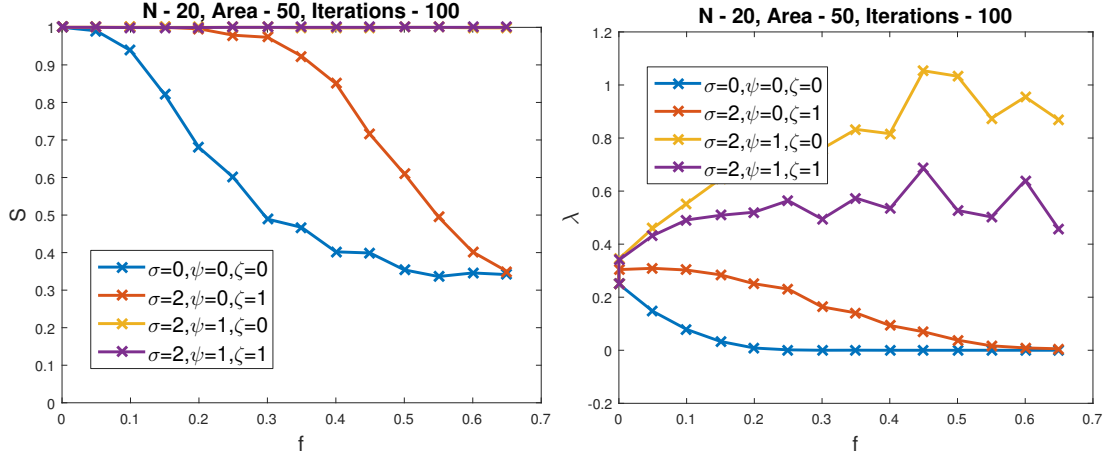


Figure 17: The giant component and the algebraic connectivity evolution.

5.4 Simulations

5.4.1 Experimental setup

The experimental setup used for evaluating the coverage performance is the same presented in Section 3.1.1, except for the gain settings. Taking into account that gains for connectivity maintenance and robustness to failures improvement control laws were fairly discussed in [9, 10], $\sigma = 2$ and $\psi = 1$ were adopted. For evaluating the impact of the coverage mechanism on the previous model performance, its corresponding gain was defined as $\zeta = 1$. As the connectivity is mandatory – in a fault-free environment – its gain is always active $\sigma = 2$ and then combined with coverage only ($\sigma = 2, \psi = 0, \zeta = 1$), robustness only ($\sigma = 2, \psi = 1, \zeta = 0$) and both coverage and robustness ($\sigma = 2, \psi = 1, \zeta = 1$). The scenario in which none of the control laws is active ($\sigma = 0, \psi = 0, \zeta = 0$) was also analyzed.

5.4.2 Simulation results

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

Regarding the fault-prone environment scenario, the impact of failures on the network connectivity was fairly minimized by having the robustness improvement control law active ($\psi = 1$). Figure 17 shows the evolution of the giant component (on the left) and the algebraic connectivity (on the right) by the fraction of robots that remain into the network.

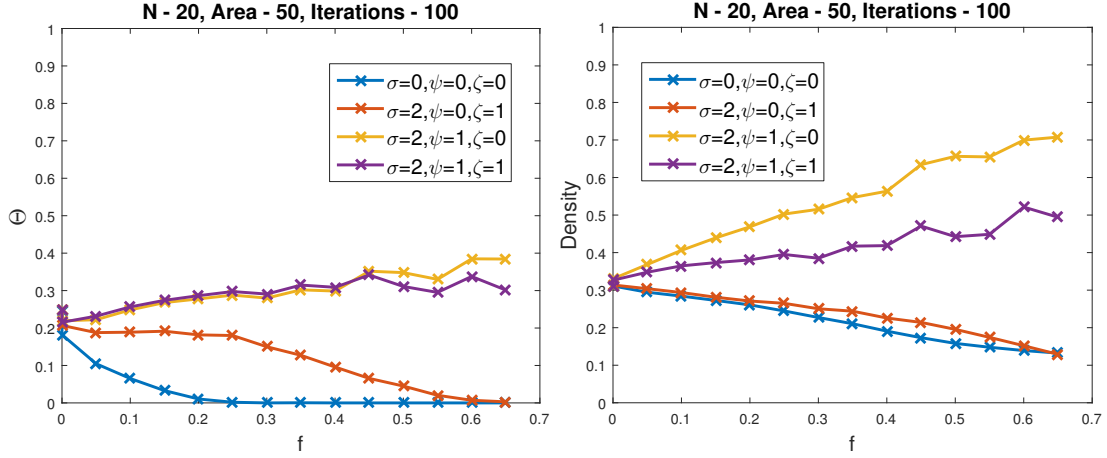


Figure 18: The robustness level and network density evolution.

The robustness level performance also demonstrates that networks could tolerate more robot failures before fragmentation, even in the presence of the additional coverage area control law (Figure 18). It is important to notice that, despite similar performance regarding the robustness level for $\psi = 1$, the network density (on the right in Figure 18) was not too affected because of the coverage mechanism action ($\zeta = 1$), which is a desirable behavior.

This is also reinforced by the coverage area and the giant component analysis: despite the total coverage area, depicted in Figure 19, have decreased when the robustness control law was active, most of the robots were still connected (Figure 17). Besides, the coverage area averaged by the number of robots indicates that robots are more spread out in the environment, i.e., the area served by each robot, on average, increased (on the right in Figure 19). Figure 20 shows the results for the average distance to the network center (on the left) and the overlapped areas (on the right) emphasizing the previous analysis.

Figures 21 to 24 depict the corresponding results for fault-free environments. They demonstrate that the action of the coverage area control law was able to mitigate the effect, imposed by the robustness improvement control law, of letting robots close to each other, without negatively impact the network robustness level (Figure 22).

In general, the experiment's achievements point out to the feasibility of setting the model gains according to the application requirements: to enhance the coverage area regardless the robustness to failure improvement ($\sigma = 2, \psi = 0, \zeta = 1$) or to focus on providing more resilient

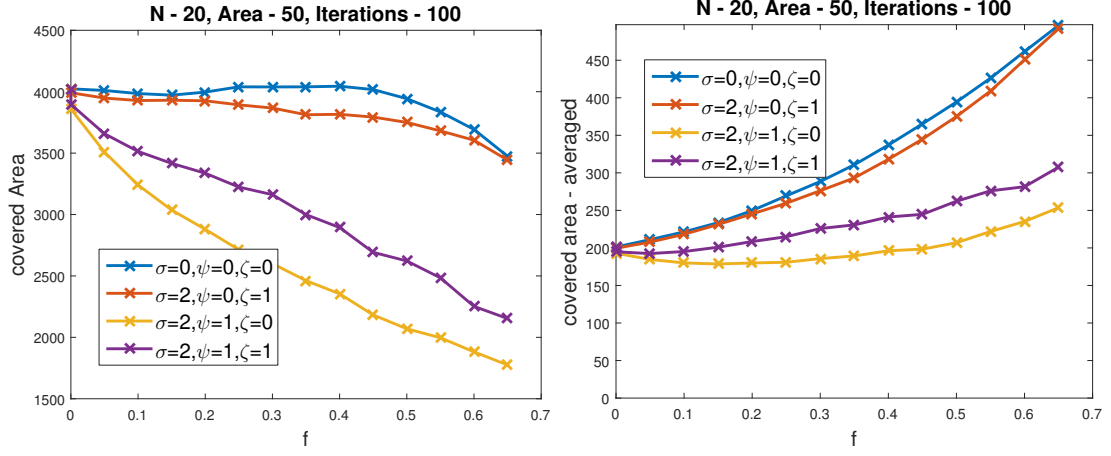


Figure 19: The coverage area and the average coverage area evolution.

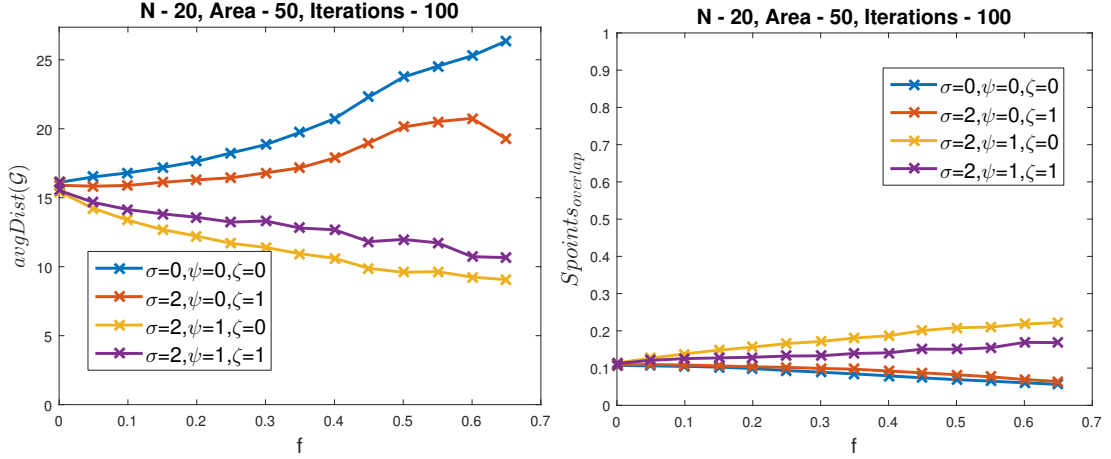


Figure 20: The average distance to the network center and overlapped areas evolution.

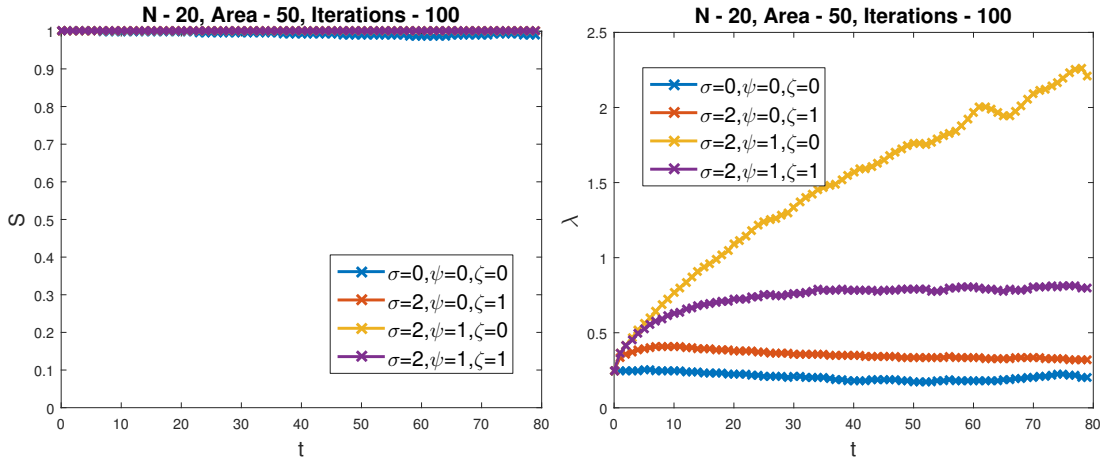


Figure 21: The giant component and the algebraic connectivity evolution.

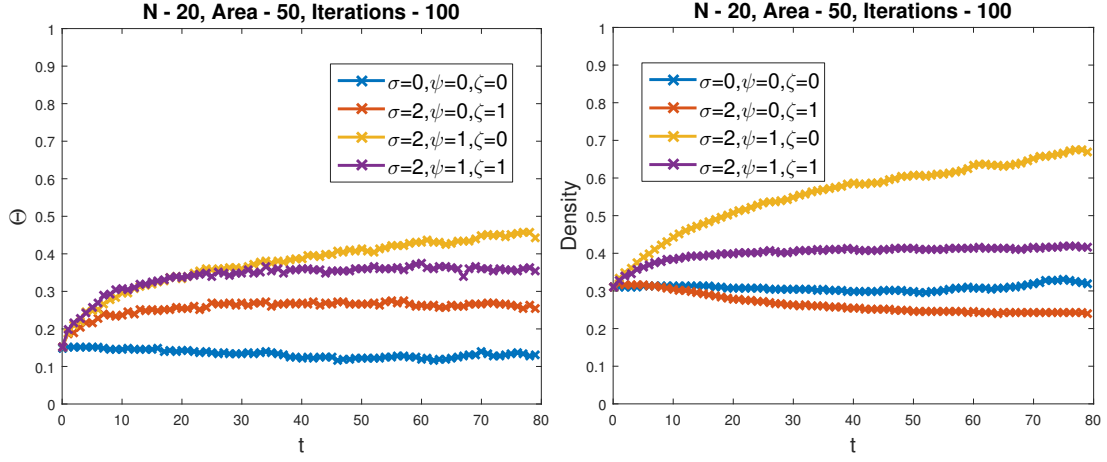


Figure 22: The robustness level and network density evolution.

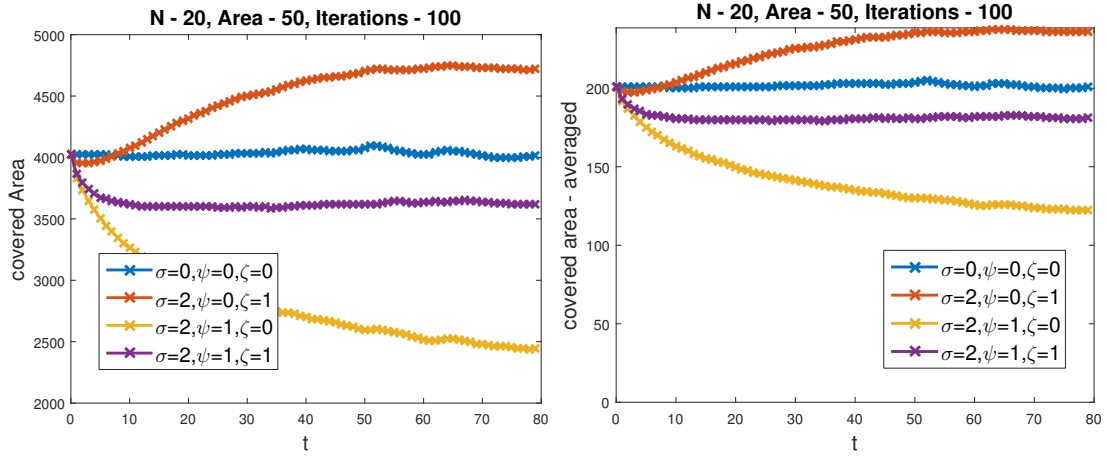


Figure 23: The coverage area and the average coverage area evolution.

networks despite the tendency of robots to get quite close to each other ($\sigma = 2, \psi = 1, \zeta = 0$). On the other hand, it is possible for both controllers operating together to delivery a more balanced network topology.

6 Conclusions and future works

In this report we presented the activities carried out during the one-year regular postdoctoral scholarship extension. The overall purpose of this research was to extend the proposed mechanism that aims at improving the robustness of complex networks to failure and attacks. The new functionalities intended to provide ad-hoc multi-robot networks with means to produce efficient

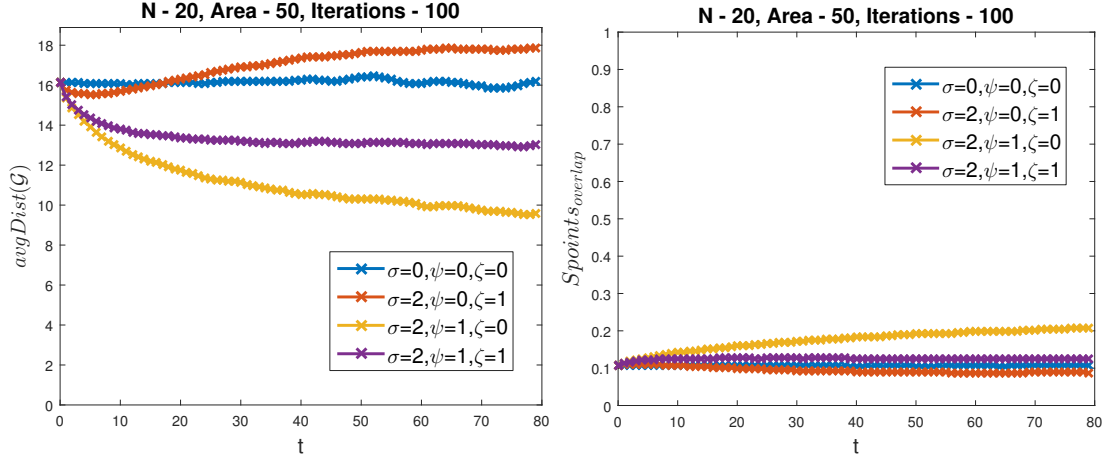


Figure 24: The average distance to the network center and overlapped areas evolution.

topology formation. In these lines, at first, a continuous-time model for the robustness to failures mechanism were implemented. Then, three mechanisms were incorporated to the model: (1) connectivity maintenance, (2) collision-avoidance and, (3) coverage area improvement.

The connectivity maintenance and the collision avoidance allow robots to keep connected while avoiding to collide with each other and with obstacles. For evaluating the combined model performance, a new experimental setup was developed, in which robots are exploring an environment with obstacles. The impact of failure time distribution on the mechanism performance was also evaluated. The results corroborate with the formal proof that, for fault-free environments, the connectivity is always guaranteed and the collisions are avoided. For fault-prone environments the combined mechanism was able to postpone or even avoid networks fragmentation. This approach was presented at two international conferences [9, 10]. An extended version of [10] was submitted to the special issue of Networks journal (Wiley), dedicated to the RNDM 2016 (International Workshop on Resilient Networks Design and Modeling), as invited paper.

The process of enhancing the robustness to failures does not take into account how robots are far from each other. Therefore, the capacity of robots to sense the environment can be improved by employing mechanisms to increase the coverage area. In this sense, we developed a model that intends to accomplished that. The resulting model imposed some challenges because the existing approaches that address the coverage problem exhibit some constraints on the requirement of using local information without previous knowledge of the environment. We

have adopted an approach for generating Centroidal Voronoi Tessellations (CVT) and combined it with a strategy for bounding Voronoi tessellations. This solution is a significant contribution because it provides a local strategy for unknown and unbounded environments.

The coverage area mechanism was then added to the previous model. Preliminary results demonstrated the feasibility of a model, relying on local information, that combines several strategies as a means of providing more efficient network topology formation. The gain settings are the main responsible for tuning the model behavior according to the expected outcomes. This new combined model is being detailed through a paper, which is intended to be submitted to a journal.

In general, we can emphasize as our main achievements the model evolution and its validation on a simulation testbed. As future work, adaptively setting gains according to the network state can significantly improve the model already proposed and heading towards a self-adaptive network topologies. Regarding the multi-robot application, it is necessary to validate the proposal presented here on real robot platforms.

In addition to the parametrization and the fine-tuning of the integrated model, the addition of a cost function and the awareness about the environment state are some of the main points to be addressed. More specifically, we can emphasize as potential future works:

- The addition of a cost function to evaluate the trade-off between the adaptation gain and its cost.
- The design of an efficiency-based model. This means aggregating mechanisms to assess the impact of the integrated model actions to the network efficiency. This information can be used for decision making. Thus, the system will try not only to adjust the topology robustness and connectivity, but also to maintain the system efficiency at a desired level.
- The model implementation in a real environment setup. The possible candidates are the e-puck mobile robots and flying robots (e.g., quadcopters).

7 Final remarks and summary of results

As final remarks we want to highlight the main achievements of the entire postdoctoral project. The research primary goal was to develop mechanisms that aiming at providing more robust network topologies regarding failures of elements. We started with a model for static network proposed in [14]. This model validation was extended for considering larger network models and additional real complex networks, and its performance was confronted with a random self-regenerating mechanism. This work yielded a publication in the peer-reviewed journal *Advances in Complex Systems* [13].

This approach was the base for the development of a hierarchical solution based on the community structure. The first aspect highlighted was that central nodes are probably those connecting communities and, therefore, information about the community structure can be worthwhile to design more resilient networks, w.r.t. failures and attacks tolerance. This work was published and presented at PESARO - The Fifth International Conference on Performance [15].

At this point we started to consider more real application scenarios. A natural choice was multi-robot network applications, since robots are prone to failure. In this sense, a collaboration with UNIMORE (Università degli Studi di Modena e Reggio Emilia) started as a concept that aimed at integrating two existing approaches regarding network connectivity: connectivity maintenance and robustness to failures. For fitting to the new model requirements and features, a new approach to mitigate the impact of failures on the network connectivity was proposed and presented at the 11th Symposium on Robot Control - SYROCO 2015 [11].

This model was then combined with the algebraic connectivity. The results was presented at the International Workshop on Resilient Networks Design and Modeling [10]. Given that this model was extensively validated, the collision avoidance mechanism was added to it. For supporting this functionally, a new experimental setup was developed in MATLAB[®]. The proposal was presented at the International Symposium on Distributed Autonomous Robotic Systems (DARS) [9]⁴.

This model is under assessment on collaborative UAV (Unmanned Aerial Vehicle) scenario

⁴Some additional examples can be freely viewed online on <https://youtu.be/ueo7nYEA24>.

through a master degree thesis. The main goal is to improve the stealth of UAV networks. As the environment is potentially vulnerable for UAVs, the robustness to failure is a desirable feature. An outline of the proposal was presented at *Simpósio de Ciência e Tecnologia do Instituto de Estudos Avançados* [4], and a paper with the first achievements was submitted to the Computer on the Beach Symposium, which is supported by the Brazilian Computer Society (Portuguese: Sociedade Brasileira de Computação, SBC).

Concerning the coverage approach, further validation and possible extensions are under development in the context of a master degree thesis. We intend to address the future works highlighted in Section 6 and reinforce our collaboration with UNIMORE through a new research project to be submitted.

Figure 7 summarizes the main activities carried out during the project extension.

Activity	2015					2016										
	08	09	10	11	12	01	02	03	04	05	06	07	08	09	10	11
Modelling and implementation of the continuous time model	X	X	X	X	X	X	X	X	X	X						
Modelling and implementation of new experimental setup for the continuous time model		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Paper submission, review and camera ready RNDM - 2016				X	X			X				X				
Collision avoidance implementation			X	X												
Paper submission, review and camera ready – DARS 2016					X	X			X					X		
Preparing paper for RNDM special issue journal (Networks)															X	X
Modelling and implementation of the distribution failure time model						X	X									
Modelling and implementation of the coverage approach										X	X	X	X	X	X	X
Prepare the final technical report																X

Figure 25: Activities carried out from August/2015 to November 2016.

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 Transactions on Automatic Control*, 55:2828–2833, 2010.
- [2] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.
- [3] Meysam Argany, Mir Abolfazl Mostafavi, Farid Karimipour, and Christian Gagné. *A GIS Based Wireless Sensor Network Coverage Estimation and Optimization: A Voronoi Approach*, pages 151–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [4] Nicolas P. Borges, Cinara. G. Ghedini, and Carlos H. C. Ribeiro. Improving mission success rate in collaborative uav networks based on a stealth approach. In *Simpósio de Ciência e Tecnologia do Instituto de Estudos Avançados*, volume 1, pages 190–195, November 2016.
- [5] Andreas Breitenmoser, Mac Schwager, Jean claude Metzger, and Daniela Rus. Voronoi coverage of non-convex environments with a group, 2010.
- [6] Y. Cao and W. Ren. Distributed coordinated tracking via a variable structure approach – part I: consensus tracking. part II: swarm tracking. In *Proceedings of the American Control Conference*, pages 4744–4755, 2010.
- [7] Pin-Yu Chen and Kwang-Cheng Chen. Information epidemics in complex networks with opportunistic links and dynamic topology. In *Global Telecommunications Conference (GLOBECOM)*, pages 1–6. IEEE, Dec 2010.
- [8] K. D. Do. Formation tracking control of unicycle-type mobile robots with limited sensing ranges. *IEEE Transactions on Control Systems Technology*, 16:527–538, 2008.
- [9] C. Ghedini, C. H. C. Ribeiro, and L. Sabattini. A decentralized control strategy for resilient connectivity maintenance in multi-robot systems subject to failures. In *Proceedings of the*

International Symposium on Distributed Autonomous Robotic Systems (DARS), London, UK, nov. 2016.

- [10] C. Ghedini, C. H. C. Ribeiro, and L. Sabattini. Improving the fault tolerance of multi-robot networks through a combined control law strategy. In *Proceedings of the International Workshop on Resilient Networks Design and Modeling (RNDM)*, Halmstadt, Sweden, sep. 2016.
- [11] C. Ghedini, C. Secchi, C. H. C. Ribeiro, and L. Sabattini. Improving robustness in multi-robot networks. In *Proceedings of the IFAC Symposium on Robot Control (SYROCO)*, Salvador, Brazil, aug. 2015.
- [12] Cinara Ghedini and Carlos H. C. Ribeiro. Rethinking failure and attack tolerance assessment in complex networks. *Physica A: Statistical Mechanics and its Applications*, 390(23–24):4684–4691, November 2011.
- [13] Cinara Ghedini and Carlos H. C. Ribeiro. Improving resilience of complex networks facing attacks and failures through adaptive mechanisms. *Advances in Complex Systems*, 17(02):1450009, 2014.
- [14] Cinara G. Ghedini. *A Proposal Towards Resilient Complex Networks Through Evaluation and Adaptation Mechanisms*. PhD thesis, Instituto Tecnológico de Aeronáutica - ITA, 2012.
- [15] Cinara G. Ghedini and Carlos H. C. Ribeiro. Using community structure information to improve complex networks robustness. In *PESARO - The Fifth International Conference on Performance*, pages 8–14. IARIA, April 2015.
- [16] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.
- [17] Luyi Chen Binbin Zhou Ping Xu Guoyong Dai, Hexin Lv. A novel coverage holes discovery algorithm based on voronoi diagram in wireless sensor networks.

- [18] Zhiwei He, Shuai Liu, and Meng Zhan. Dynamical robustness analysis of weighted complex networks. *Physica A: Statistical Mechanics and its Applications*, 392(18):4181 – 4191, 2013.
- [19] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Talyor. Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*, 25(1):111–131, 2008.
- [20] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4):519–528, 2005.
- [21] M. Ji and M. Egerstedt. Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 2007.
- [22] D. Lee, A. Franchi, H.I. Son, C. Ha, H.H. Bulthoff, and P. Robuffo Giordano. Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles. *IEEE/ASME Transactions on Mechatronics*, 18(4):1334–1345, Aug 2013.
- [23] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006.
- [24] M. Manzano, E. Calle, V. Torres-Padrosa, J. Segovia, and D. Harle. Endurance: A new robustness measure for complex networks under multiple failure scenarios. *Computer Networks*, 57(17):3641 – 3653, 2013.
- [25] David M. Mount. Computational geometry. *Dept. of Computer Science - University of Maryland*, 2005.
- [26] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie. Maintaining limited-range connectivity among second-order agents. In *Proceedings of the American Control Conference*, pages 2134–2129, 2006.
- [27] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bühlhoff. A passivity-based decentralized strategy for generalized connectivity maintenance. *The International Journal of Robotics Research*, 32(3):299–323, 2013.

- [28] L. Sabattini, N. Chopra, and C. Secchi. Decentralized connectivity maintenance for cooperative control of mobile robotic systems. *The International Journal of Robotics Research (SAGE)*, 32(12):1411–1423, October 2013.
- [29] L. Sabattini, C. Secchi, and N. Chopra. Decentralized estimation and control for preserving the strong connectivity of directed graphs. *IEEE Transactions on Cybernetics*, 2014.
- [30] Erik Tjong Kim Sang. Voronoi diagrams without bounding boxes. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-2/W2, 2015.
- [31] Anju Sangwan and Rishi Pal Singh. Survey on coverage problems in wireless sensor networks. *Wireless Personal Communications*, 80(4):1475–1500, 2015.
- [32] Mac Schwager, James McLurkin, Jean-Jacques E. Slotine, and Daniela Rus. *From Theory to Practice: Distributed Coverage Control Experiments with Groups of Robots*, pages 127–136. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [33] Saeed Sedighian Kashi and Mohsen Sharifi. Coverage rate calculation in wireless sensor networks. *Computing*, 94(11):833–856, 2012.
- [34] R. Soukieh, I. Shames, and B. Fidan. Obstacle avoidance of non-holonomic unicycle robots based on fluid mechanical modeling. In *Proceedings of the European Control Conference*, Budapest, hungary, 2009.
- [35] Tien-Wen Sung and Chu-Sing Yang. Voronoi-based coverage improvement approach for wireless directional sensor networks. *Journal of Network and Computer Applications*, 39:202 – 213, 2014.
- [36] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I.L. Grix, 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(3):46–56, Sept 2012.
- [37] Zhigang Wang, Lichuan Liu, and MengChu Zhou. Protocols and applications of ad-hoc robot wireless communication networks: An overview. *future*, 10:20, 2005.

- [38] Stanley Wasserman, Katherine Faust, and Dawn Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.
- [39] Ulf Witkowski, Mohamed Ahmed Mostafa El Habbal, Stefan Herbrechtsmeier, Andry Tanoto, Jacques Penders, Lyuba Alboul, and Veysel Gazi. Ad-hoc network communication infrastructure for multi-robot systems in disaster scenarios. In *Proceedings of IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance (RISE 2008)*, Benicassim, Spain, 2008.