Ad Hoc Networks 74 (2018) 57–70

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Toward efficient adaptive ad-hoc multi-robot network topologies

Cinara Ghedini[a,*], Carlos H.C. Ribeiro[a], Lorenzo Sabattini[b]

[a] Computer Science Division, Aeronautics Institute of Technology, São José dos Campos, SP, Brazil
[b] Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy

ARTICLE INFO

Article history:
Received 5 September 2017
Revised 4 January 2018
Accepted 27 March 2018
Available online 28 March 2018

Keywords:
Self-organizing networks
Fault-tolerance networks
Coverage problem
Multi-robot networks

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 hazardous 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. Hence, successful control and coordination of a group of wireless-networked robots relies on effective inter-robot communication. In this sense, this work proposes a model that aims at providing more efficient network topologies addressing the issues of connectivity maintenance, collision avoidance, robustness to failure and area coverage improvement. The model performance was experimentally validated considering fault-free and fault-prone scenarios. Results demonstrated the feasibility of having simultaneous controls acting to achieve more resilient networks able to enhancing their sensing area while avoiding collision and maintaining the network connectivity with regard to fault-free scenarios.

1. Introduction

Multi-robots networks open up a wide range of applications possibilities. Robots can now act in 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 fact, in such situations great benefit can be achieved from the assistance of robotic systems [1,2]. Another major application is to deliver a communication infrastructure to clients (e.g. mobile phones, laptops, tablets, sensor, other robots, etc.), when there is no network infrastructure available, either because it is not possible from an economic or a technological point of view, or the previous infrastructure was damaged during a disaster event [3].

In general, these applications require handling teamwork efficiently, *i.e.*, they must be developed to guarantee the capability of constructing a network and performing cooperative work [4]. To maximize autonomy, robots are usually equipped with low-power, short-range network interfaces, which only allow direct communication with their nearby neighbors. Thus, it is virtually impossible for each node to know the entire network topology at any given time. Under these circumstances, the only practical approach to distributed command, control and sensing is to employ an ad hoc wireless networking scheme where the nodes themselves are responsible for routing packets [5]. Hence, successful control and coordination of a group of wireless-networked robots relies on effective inter-robot communication [6].

In addition to the limited processing, storage and bandwidth, the physical devices are prone to failures adding uncertainty and unpredictability in the environments in which they operate. Thus, providing efficient solutions based on ad hoc networks include requirements other than those from the task itself, implying mechanisms to deal with diverse design factors, such as fault tolerance, connectivity maintenance, power consumption, communication efficiency, scalability and coverage.

The ad hoc feature means that the network topology is self-organized, *i.e.* built spontaneously as devices connect, and according to the system dynamics. On the other hand, the system *per se* is not self-organized. Self-organization here refers to a class of systems that are able to change their internal structure and their function in response to external circumstances. Such systems are commonplace in ecological, biological and social systems [7–11]. It is known that although such systems are immersed in dynamic and unpredictable environments, most of them exhibit the ability of environment perception, learning and adaptation, which, in turn, allows them to adjust their behavior in order to maintain the system functionality, even in the presence of disturbances. Because of that, these systems are regarded as intrinsically resilient.

[*] Corresponding author.
E-mail addresses: cinarag@gmail.com (C. Ghedini), carlos@ita.br (C.H.C. Ribeiro), lorenzo.sabattini@unimore.it (L. Sabattini).

https://doi.org/10.1016/j.adhoc.2018.03.012
1570-8705/© 2018 Elsevier B.V. All rights reserved.

Resilience is related to the capacity of a system to absorb disturbance and to reorganize while undergoing change, essentially keeping the same function, structure, identity, and feedbacks [12]. This means that resilient systems are efficient to persist in a very dynamic and unpredictable environments. The indicator of the system performance under perturbation is known as robustness. On the other hand, as there are diverse sources that can trigger adverse or unpredictable events, the properties of the system can be affected in different levels. Thus, a system can be more robust to some events than others. For instance, a communication network can be robust to failures in the network infrastructure but not to unpredictable traffic load, or to keep the integrity of data in cases of improper activities [13].

In this sense, it is expected that new technological applications exhibit some properties akin to self-organization, with scalability and reliability. Over the last decades, a variety of properties have been referred to self-organized systems, such as self-healing, self-organizing, self-configuration, self-adaptation and self-optimization [14]. Specifically, in the context of multi-robot models, some of them are desirable for most applications: (1) the capacity of **(self-)adaptation** to environmental changes; (2) the capacity of **(self-)optimization**, i.e. the ability to improve or maximize/minimize some property as appropriate to fulfill an objective function of a global or local state; and, (3) the capacity of **(self-)configuration**, maintaining or improving the system configuration to optimize performance.

These properties are related to several system components, from topology formation to communication protocols. However, based on the premise that the primary requirement for any application based on multi-robot networks is the existence of a reliable and efficient communication infrastructure, the main subject of concern here is to address these self-properties in the context of *topology formation*. It means providing applications with mechanisms to drive the network to achieve a more efficient and reliable topology that can support several kinds of applications, despite their particular requirements. This is not trivial because the topology emergent from these applications is dynamic; thus, its size and properties are most of the time unknown, costly to estimate and time varying, which impose constraints on the design of solutions, such as a need for relying mostly on information that is locally available and straightforward to obtain, compute and update.

Self-adaptive systems are capable of sensing the environment and reacting in order to revert harmful states or to achieve a better functionality or performance [15]. For possessing adaptability skills, robots must be provided with mechanisms that allow them to analyze data sensed from the environment and to control their motion in order to achieve a desirable state. The design of such mechanisms depends on the target property to be optimized and the system vulnerabilities.

In this sense, regarding an efficient topology formation in multi-robot scenarios, connectivity is the main property of concern. Connectivity maintenance is a very well studied topic in the field of decentralized multi-robot systems, and there are several strategies in the literature for tackling this problem [16–21]. On the other hand, network connectivity can be strongly affected by robot failures, which can lead to a global state of vulnerability, *i.e.*, the network is potentially able to fragment 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. Fig. 1 illustrates a random network topology, highlighting some agents playing important roles in the network communication, emphasizing vulnerable configurations.

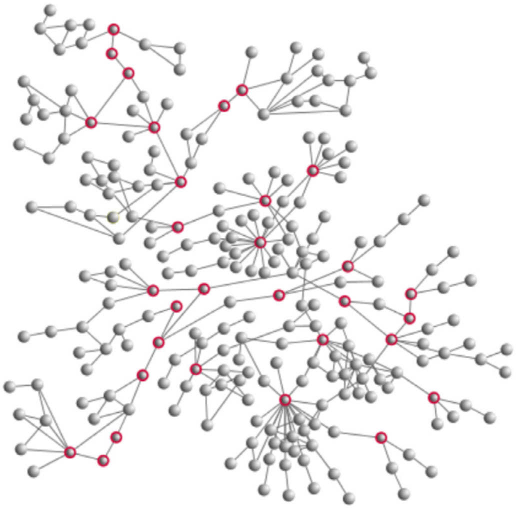


Fig. 1. Illustration of a random network topology highlighting nodes which failure can negatively affect the network connectivity, thus producing inoperative or reduced communication services.

A mechanism, relying on locally available information, for detecting and mitigating vulnerable topological configuration is proposed in [22]. The results demonstrate that the mechanism was able to increase the system resilience by adapting the network topology to accommodate failures, postponing or even avoiding network fragmentation. This model was combined with connectivity maintenance and collision avoidance mechanisms, demonstrating the feasibility of having several control laws acting in order to provide more resilient network topologies [23].

Considering that the combined mechanism above provides a more reliable network regarding connectivity maintenance and failure tolerance, and taking into account that several application based on multi-robot networks perform tasks that demand environment exploration, we propose here an extension of the previous model by addressing the area coverage problem. In our context, the main goal of providing networks with a area coverage mechanism is to improve the capacity of a multi-robot network of sensing the environment, reducing redundant monitoring areas.

The area coverage solution is based on a local Voronoi coverage mechanism presented in [24]. This approach, however, considers a bounded environment, which is not a desirable feature regarding the scenario adopted. For overcoming this limitation, we applied an approach proposed in [25], 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 area coverage model was then combined with the previous model, and evaluated in the same experimental setup. The results show that the model was able to provide networks with some features of **(self-)optimization** w.r.t. topology formation, through improvements in area coverage and robustness to failure.

Naturally, a model that combines several features must include tools for adaptively setting them according to the application requirements and the devices/network states (e.g., battery level, sensor feedback, failure occurrences). For instance, robustness to failure can consider the probability of a node failure by measuring the battery level or/and the frequency of failures. This can result in a desirable behavior and bring some level of **self-configuration** to the system. The combined control law provides means of tuning each mechanism gain settings. We do not consider adaptive setting of gains, but we demonstrate the performance of different gain setting combinations.

Summarizing, this paper presents a model for providing efficient network topology formation regarding connectivity maintenance, robustness to failures and area coverage . Moreover, and as already mentioned, these properties are related to the interconnection topology among the mobile robots. Therefore, they are global quantities: in order to evaluate them, a global knowledge of the communication network is necessary. However, it is absolutely unpractical to assume that each mobile robot has a global knowledge of the system. Here we consider dynamic topologies composed by mobile agents which will autonomously organize themselves to communicate and integrate information efficiently. Besides, the procedures are always based on local information and independent of fixed or central infrastructure. We believe that this new combined model can be used as a reference for providing self-organized network solutions.

The necessary background on network properties and the system model are presented in Section 2. Sections 3 and 4 detail the continuous-time model for the algebraic connectivity and robustness to failure combined control laws, respectively. Section 5 describes the area coverage approach. The new model performance is presented in Section 6. Conclusions and future works are discussed in Section 7.

2. Preliminaries

2.1. Background on network theory

Consider an undirected graph $\mathcal{G}$, where $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G}) \subseteq \mathcal{V}(\mathcal{G}) \times \mathcal{V}(\mathcal{G})$ 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 ith 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 [26]. Let λ_i, $i = 1, \dots, N$ be the eigenvalues of the Laplacian matrix, then:

- The eigenvalues are real, and can be ordered such that $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$.
- 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 [27]. As the algebraic connectivity goes to zero 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:

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

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 [28]. 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 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:

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

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*. The *robustness level* is then introduced as follows.

Definition 2.1 (Robustness level [22]**).** 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 $\left[v_{\varphi+1}, \dots, v_N\right]$ is disconnected. Then, the *robustness level* of $\mathcal{G}$ is defined as:

$$\Theta(\mathcal{G}) = \tfrac{\varphi}{N}. \tag{3}$$

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 [22]. 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: considering, for instance, $\beta = 1$, then $Path_\beta(v)$ includes only the 2-hop neighbors of node v that are connected through only one path. Hence, the number of nodes included in $Path_\beta(v)$, that is $|Path_\beta(v)|$, is an indicator of how fragile the connection of node v is with the rest of the network. In fact, if the majority of the 2-hop neighbors of v

are connected through a limited number of paths, it means that it is very likely, for v, to be disconnected by the rest of the network, in the case of failure of some critical nodes. Consequently, we define $P_\theta(v)$ as the vulnerability level of a node regarding failures, as follows:

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

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 $p = \left[p_1^T \dots p_N^T\right]^T \in \mathbb{R}^{Nm}$ 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

$$\dot{p}_i = u_i, \tag{5}$$

where $u_i \in \mathbb{R}^m$ is a control input.[1] In order to guarantee the connectivity of $\mathcal{G}$, Sabattini et al. [21] 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:

$$w_{ij} = \begin{cases} e^{-\left(\|p_i - p_j\|^2\right)/(2\sigma^2)} & \text{if } \|p_i - p_j\| \le R \\ 0 & otherwise. \end{cases} \tag{6}$$

with

$$e^{-(R^2)/(2\sigma^2)} = \Delta \tag{7}$$

where Δ is a small predefined threshold.[2]

3. Connectivity maintenance and collision avoidance control laws

Considering the algebraic connectivity λ to be the property to support the connectivity maintenance mechanisms and $\epsilon > 0$ to be the desired lower-bound for the value of λ, the control strategy is then designed to ensure that λ never drops below ϵ. As in [21], the following *energy function* can be used for generating the decentralized connectivity maintenance control strategy:

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

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:

$$u_i = u_i^c = -\frac{\partial V(\lambda)}{\partial p_i} = -\frac{\partial V(\lambda)}{\partial \lambda}\frac{\partial \lambda}{\partial p_i}. \tag{9}$$

The connectivity maintenance framework can be enhanced to consider additional objectives. In particular, as shown in [32], 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 weights*:

$$\omega_{ij} = w_{ij}\gamma_{ij}, \tag{10}$$

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

Definition 3.1. The collision avoidance edge weights γ_{ij} exhibits the following properties, $\forall i, j = 1, \dots, N$:

(P1) $\gamma_{ij} = \gamma_{ij}\left(\|p_i - p_j\|\right) \ge 0$.
(P2) $\gamma_{ij} = 0$ if $\|p_i - p_j\| = 0$, and $\gamma_{ij} = 1$ if $\|p_i - p_j\| \ge d_s$.
(P3) $\gamma_{ij}(d)$ 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 δ_s, then the collision avoidance action is not necessary. As shown in [32], the same formalism can be exploited for avoiding collisions with environmental obstacles as well.

Utilizing the generalized edge weights ω_{ij} defined in (10), 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 [32], 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 (9) replacing λ with φ, namely

$$u_i = u_i^c = -\frac{\partial V(\varphi)}{\partial p_i} = -\frac{\partial V(\varphi)}{\partial \varphi}\frac{\partial \varphi}{\partial p_i}. \tag{11}$$

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 [21].

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, by avoiding vulnerable topological configurations.

4. 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:

$$u_i = \sigma u_i^c + \psi u_i^r, \tag{12}$$

where u_i^c is the generalized connectivity maintenance control law introduced in (11), and u_i^r is the additional control law for improving robustness to failures. The parameters σ, $\psi \ge 0$ are the control gains, setting any of them to zero leads to removing the effect of the corresponding control law.

Based on the vulnerability level given by (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. Let $x_\beta^i \in \mathbb{R}^m$ be the barycenter of the positions of the robots in $Path_\beta(i)$, namely

[1] 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 [29], and UAVs [30].

[2] This definition of the edge–weights introduces a discontinuity in the control action, that can be avoided introducing a smooth bump function, as in [31]. However, from an implementation viewpoint, the effect of the discontinuity can be made negligible by defining the threshold Δ sufficiently small.

$$x^i_\beta = \frac{1}{\left|Path_\beta(i)\right|} \sum_{j \in Path_\beta(i)} p_j, \tag{13}$$

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

$$u^r_i = \xi_i \frac{x^i_\beta - p_i}{\left\| x^i_\beta - p_i \right\|} \alpha(t), \tag{14}$$

where $\alpha(t) \in \mathbb{R}$ is the linear velocity of the robots.[3]

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:

$$\xi_i = \begin{cases} 1 & \text{if } P_\theta(i) > r \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

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^c_i was proven in [21,32] 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^r_i.

Theorem 1. *Consider the dynamical system described by (5), and the control laws described in (11), (12) and (14). 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 both 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 (8) 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 (10), 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 Sabattini et al. [21]. □

5. Coverage problem

The coverage mechanism in the context of this work aims at improving the sensoring capacity of a multi-robot network, reducing the redundant monitoring areas and avoiding uncovered regions in the network. Coverage can be defined as how well or to which extent each point of a deployed network is under the vigilance of a sensor node [33].

In general, robots should be distributed around the environment in a way that the overlapping 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. However, an ad hoc network formation or even the presence of obstacles can generate uncovered regions in the sensing area, which may lead to routing failure and degradation of the quality of service [34,35]. Fig. 2 illustrates the uncovered areas (holes) and redundant monitoring area (overlaps).

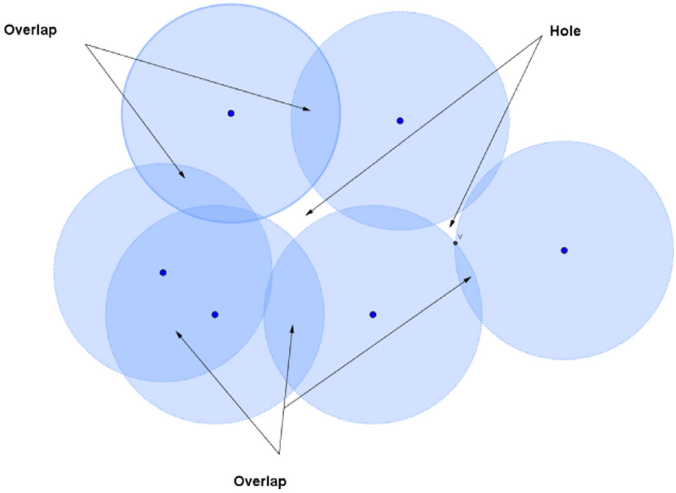


Fig. 2. Hole and overlap areas illustration.

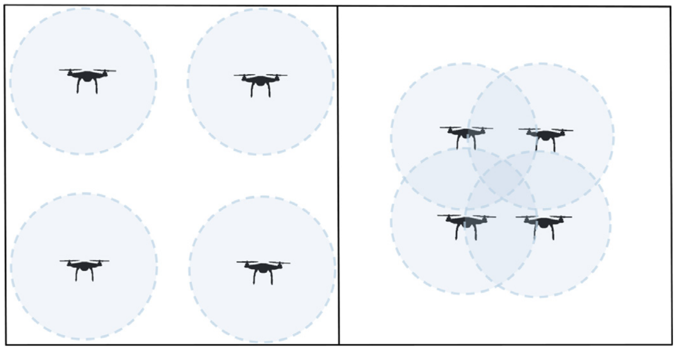


Fig. 3. The impact of having optimized coverage on the network connectivity.

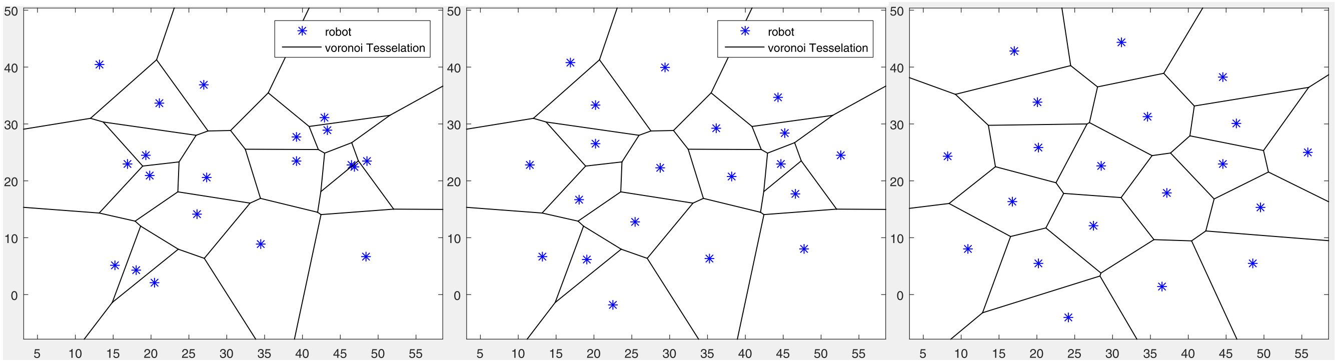
Thus, it must be assumed that given a topological configuration, the robots try to adjust their behavior in order to improve the environment coverage, i.e., to reduce overlapping areas as much as possible without impacting the overall network connectivity and robustness to failures [36,37]. Consider the example depicted in Fig. 3: the area coverage on the left-hand is better than the right-hand one, since a larger portion of the environment is covered. However, the partial overlap in the network in the right-hand figure is necessary to guarantee the preservation of the graph connectivity. Therefore, the efficiency of the control strategy can be directly measured as the portion of the environment that is covered by the mobile robots themselves and the network connectivity. Notice that the connectivity maintenance cannot be ensured because failures are mostly unpredictable. In such cases, the number of nodes still connected in the network must be taken into account.

There are several approaches addressing different coverage problems [34,38,39]. The optimization function is related to the application features and the system requirements [33]. As a consequence, different techniques can be applied. In this sense, Voronoi tessellation is commonly applied in different context of applications, not only as a means to improve the network coverage but also to evaluate its coverage level [24,33,34,40,41]. The Voronoi coverage is discussed in what 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 $P = [p_1, p_2, \ldots, p_n]$ be a set of points in the plane. Define $V(i)$, the

[3] Unlikely situations exist in which (14) is not well defined, namely when $p_i = x^i_\beta$. However, this corresponds to the case where the ith robot is in the barycenter of its weakly connected 2-hop neighbors, in practice this never happens when a robot detects itself as vulnerable.



(a) Original VT

(b) Robots at their voronoi cell center.

(c) VT of the new positions of robots.

Fig. 4. Illustration of a random network topology evolution based on moving robots towards the center of their Voronoi Tessellation (VT) cell.

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

$$V(i) = \{q \mid \|p_i q\| < \|p_j q\|, \forall j \neq i\}, \tag{16}$$

where $\|pq\|$ denotes the Euclidean distance between points p and q [42].

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 [39]. For illustration, Fig. 4a shows the Voronoi tessellation of a random network – blue dots represent the robot positions; Fig. 4b is the same Voronoi tessellation but with each robot positioned at the center of its Voronoi cell; and Fig. 4c is the Voronoi tessellation generated after two iterations 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 environment.

If the requirements demanded by the scenario features include using local information, the strategy for modeling the Voronoi coverage must be based on a local mechanism such as the one presented in [24], 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 $\phi : \Omega \to \mathbb{R} > 0$, the centroid of $V(i)$ is defined as

$$C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} x\,\phi(x)\mathrm{d}x, \tag{17}$$

where M_{V_i} is the cell mass [24]:

$$M_{V_i} = \int_{V_i} \phi(x)\mathrm{d}x. \tag{18}$$

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

$$\mathscr{H}(P) = \sum_{i=1}^{n} \mathscr{H}(p_i) = \sum_{i=1}^{n} \int_{V_i} f(D(p_i, x)),\ \phi(x)\mathrm{d}x, \tag{19}$$

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 [24]. Considering the Euclidean distance $D(p_i, x) = \|x - p_i\|$ as the distance function in (19):

$$\frac{\partial \mathscr{H}(p_i)}{\partial (p_i)} = -M_{V_i}(C_{V_i} - p_i) = 0, \tag{20}$$

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

The Voronoi coverage presented in [24] is based on the Lloyds algorithm [43], a well-established method for generating a centroidal Voronoi tessellation. It encompasses:

1. Creating the Voronoi diagram for generating points P.
2. Computing the centroid $[C_{V_i}]_{i=1}^{n}$ of each Voronoi cell.
3. Setting 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 (20), the simple first-order dynamics can be derived:

$$u_i = \dot{p}_i = -\frac{k}{M_{V_i}} \frac{\partial \mathscr{H}(p_i)}{\partial (p_i)} = k(C_{V_i} - p_i). \tag{21}$$

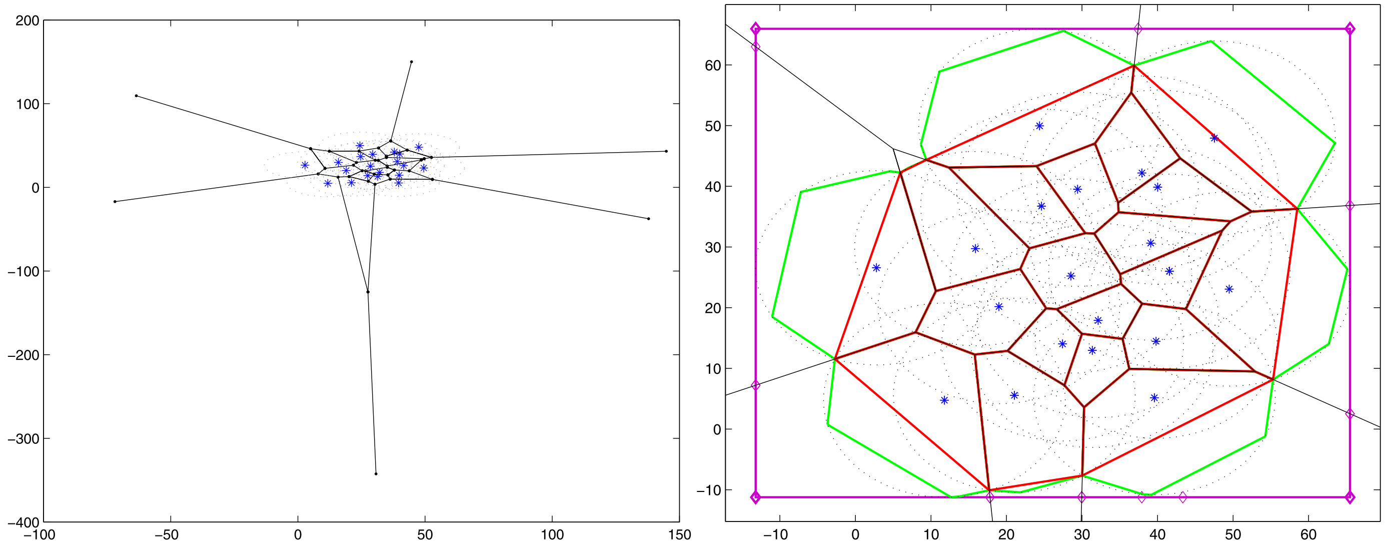
However, regarding the scenario adopted, the goal is not for robots to find an optimal position and remain there, but adaptively explore an unknown and unbounded environment in a way that the total sensing area is improved as much as possible in accordance with other active control laws. Thus, (14) is redefined as:

$$u_i = \sigma u_i^c + \psi u_i^r + \zeta u_i^v, \tag{22}$$

where u_i^v represents the coverage controller (21) 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 neighbors 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 cell ($V(i)$, defined as in (16)).
- Computes the mass of the centroid of its Voronoi cell C_{V_i} (17).
- 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, set 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., robot i does not need to improve its position regarding coverage.



(a) Voronoi Tessalation of a random graph

(b) Performance of mechanisms for bounding Voronoi tessalation Cells.

Fig. 5. Illustration of the problem of bounding Voronoi tessellation cells.

This approach imposes some constraints to the scenario adopted here: the procedure for creating a Voronoi tessellation generates some unbounded cells (see Fig. 5a). On the other hand, computing the mass of the centroid for a cell requires it to be bounded. Thus, it is necessary to define a procedure to delimit the unbounded cells. This issue is addressed in Section 5.2.

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 area coverage is the unbounded cells. The technique applied for bounding cells can impact the mechanism performance. Fig. 5b illustrates the output of three techniques applied for bounding the Voronoi cells of the network presented in Fig. 5a. The most straightforward manner, represented by magenta lines, uses 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 define the environment boundaries. The main problem with this approach is that cells can become too large, thus generating biased cell centroids. In addition, it is costly to locally estimate boundaries.

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, represented by red lines, can generate small cells, which, as a consequence, can drive robots towards the network center instead of trying to spread them around the environment.

For overcoming the drawbacks of both techniques, an approach proposed in [25] was adopted. Despite being focused on presenting geographic data through Voronoi diagrams without having to specify a bounding box, it can be applied for bounding Voronoi cells. An additional advantage is that the robot's range can be used for reducing the size or bounding cells, as the technique relies on defining a maximum size for a cell.

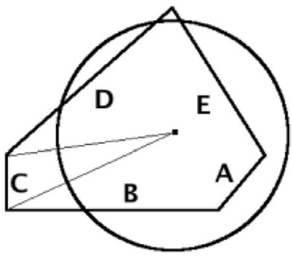


Fig. 6. Mapping edges in a Voronoi Cell [25].

Its performance is illustrated by the green lines in Fig. 5b. Notice that the size and shape of the new Voronoi cells are more balanced and in accordance with the robot range. For exemplifying the procedure, consider the Voronoi cell illustrated in Fig. 6. The circle defines the cell limits for a generating point. Every possible state classification for an edge and the actions to be taken for each of them can be demonstrated by this example. Thus, each edge may:

- 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 part;
- 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:

$$distance_{(P,Center)} = \sqrt{(P_x - Center_x)^2 + (P_y - Center_y)^2}, \tag{23}$$

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 the bounding circle (edge *C* - Fig. 6), 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:

$$NewP_x = Center_x + (P_x - Center_x)\frac{Radius}{distance(P, Center)}, \tag{24}$$

$$NewP_y = Center_y + (P_y - Center_y)\frac{Radius}{distance(P, Center)}. \tag{25}$$

For cutting the edges intersecting the circle (edges *B, D* and *E* - Fig. 6), a straight line corresponding to an edge is computed:

$$\begin{aligned} &y = a * x + b \ when \ x_1 \neq \ x_2, a = (y_2 - y_1)/(x_2 - x1) \\ &\quad and \ b = (y_1 * x_2 - y_2 * x_1)/(x_2 - x_1) \\ &x = a * y + b \ when \ x_1 = x_2, \ a = 0 \ and \ b = x_1, \end{aligned} \tag{26}$$

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:

$$\begin{aligned} y = Center_y \pm \sqrt{Radius^2 - (x - Center_x)^2} \\ if \ Radius^2 \geq (x - Center_x)^2. \end{aligned} \tag{27}$$

The Eqs. (26) and (27) are then combined to find the intersections:

$$Center_y \pm \sqrt{Radius^2 - (x - Center_x)^2} \ = \ a * x + b. \tag{28}$$

Solving the Eq. (28) in order to achieve a quadratic equation of the form $A * x^2 + B * x + C = 0$:

$$\begin{aligned} &A = -1 - a^2 \\ &B = 2 * Center_x - 2 * a * (b - Center_y) \\ &C = Radius^2 - {Center_x}^2 - (b - Center_y)^2, \end{aligned} \tag{29}$$

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 Eq. (26). Thus, *x* and *y* determine the new cell boundaries. According to Sang [25] as the computation for each cell vertex is independent, the computational complexity for bounding a voronoi diagram is $\mathcal{O}(n)$, where *n* is the number of edges in the Voronoi diagram.

6. Simulations

6.1. Experimental setup

The proposed control strategy was validated through a simulation model developed in MATLAB®. It comprises the benchmark networks, the control law parameterization and the protocol for performance evaluation. 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, *R*=16, and 50 randomly deployed obstacles ($N_{obstacles} = 50$).

Given an initial network topology, each node is supposed to explore the environment and to evaluate the measures responsible for computing the control laws, which, in turn, impose its behavior. Considering that the gains calibrate the weight of each control law, a more meaningful analysis is possible by setting different gain combinations. Taking into account that gains for connectivity maintenance and robustness to failures improvement control laws were fairly discussed in [23,44], σ=2 and ψ=1 were adopted. Besides, as connectivity is mandatory – in fault-free environments – its gain is always active (σ=2) and then combined with coverage only (σ=2,ψ=0,ζ=1), robustness only (σ=2,ψ=1,ζ=0) and both coverage and robustness (σ = 2, ψ=1,ζ=1). For example, consider gain settings σ=2,ψ=1,ζ=1. In this case, each robot must move in order to increase both its robustness to failure and sensing area, while keeping the global network connectivity and avoiding collision with obstacles and other robots. The scenario in which none of the control laws is active (σ=0,ψ=0,ζ=0) was also analyzed. Notice that, as demonstrated in Theorem 1, any combination of positive gains leads to the desired behavior.

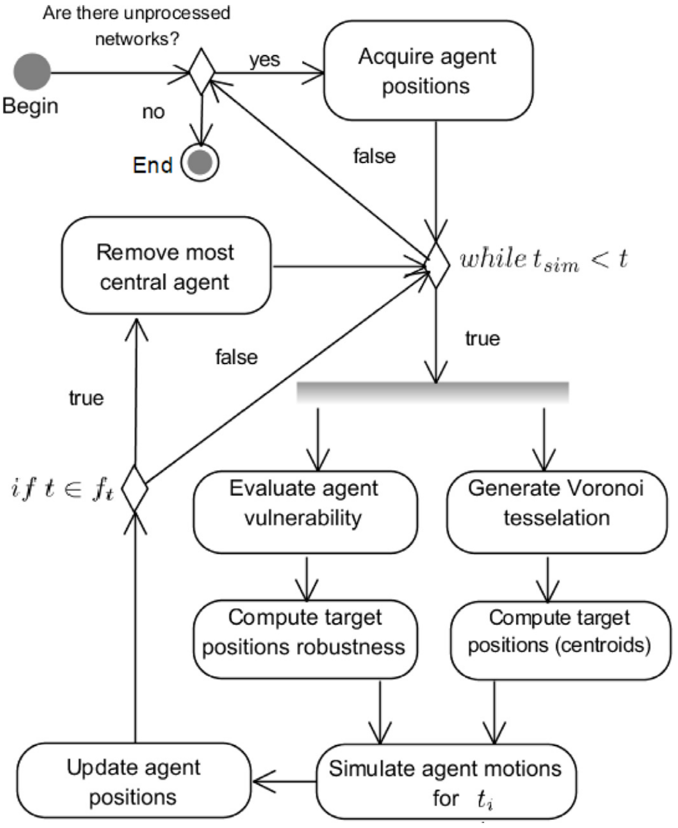

Are there unprocessed networks?
Begin
yes
no
End
Acquire agent positions
false
while $t_{sim} < t$
Remove most central agent
true
false
true
if $t \in f_t$
Evaluate agent vulnerability
Generate Voronoi tesselation
Compute target positions robustness
Compute target positions (centroids)
Update agent positions
Simulate agent motions for t_i

Fig. 7. Activity diagram illustrating the simulation general steps for the fault-prone scenario.

For properly assessing the proposed model effectiveness, fault-free and fault-prone environments were considered. The main difference is that for the fault-prone scenario, 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 f_t of t_n randomly generated failure times, distributed during the total simulation time.

Regarding the specific control law parameterization, we used ϵ=0.25, β=1 and α=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 $N_{networks} = 100$. Thus, the results presented here represent an average of 100 networks. The total simulation time was t_{sim}=80 s with the vulnerability level estimation, the local Voronoi tessellation generation and network properties evaluation being performed at every 1 s. Table 1 summarizes the simulation setup.

Fig. 7 presents a schematic diagram illustrating the main steps of the simulation process. For each network, the simulation runs for t_{sim} seconds ($while \, t_{sim} < t$), but at each t_{sim} seconds, every node updates its target positions based on its vulnerability level and the centroid of its voronoi cell. The motion of robots is simulated

Table 1
Simulation setup parametrization.

	Identification	Description	Values
Environment	$\mathcal{N}$	Number of network nodes	20
	$\mathcal{A}$	Area for obstacles deployment	50^2
	R	Communication radius	16
	$N_{obstacles}$	Number of obstacles	50
	$N_{networks}$	Number of networks	100
Model parametrization			
	$\sigma\ \psi\ \zeta$	control law gain combinations	$\sigma\ \psi\ \zeta$ 0 0 0 2 1 0 2 0 1 2 1 1
	ϵ	Lower-bound for the value of λ	0.25
	β	Threshold for the maximal number of 2-hops paths	1
	α	Linear velocity	$0.25\mathcal{R}$
	t_{sim}	Total simulation time	80s
	t_i	Iteration time	1s
	t_n	Number of failures	14
	f_t	Set of failure times $f_t = (f_{t1}, \ldots, f_{t_n})$	

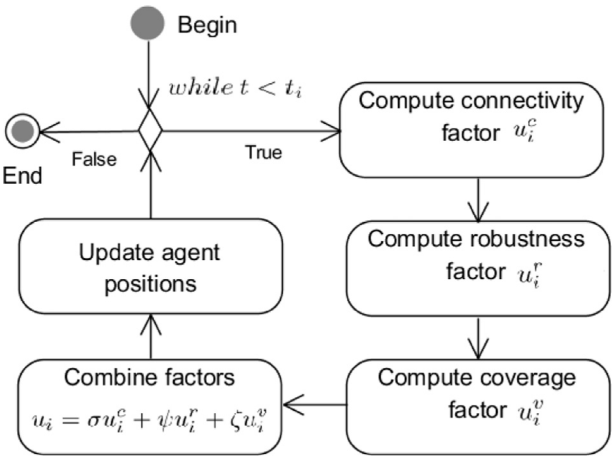

Begin
$while\ t < t_i$
False
True
End
Compute connectivity factor u_i^c
Compute robustness factor u_i^r
Compute coverage factor u_i^v
Combine factors $u_i = \sigma u_i^c + \psi u_i^r + \zeta u_i^v$
Update agent positions

Fig. 8. Activity diagram of the procedure for simulating the robots' motion according to the proposed integrated control law.

through an ordinary differential equation solver. The factor concerning each control law is computed and then integrated for updating the robot positions, as illustrated in Fig. 8. For evaluating the coverage mechanism performance, a technique for estimating the coverage rate, proposed by Kashi et al. [40], was applied. It consists of identifying the boundaries of each nodes 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 outline are positive ones.

6.2. 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 minimized by having the robustness improvement control law active (ψ=1): even when the network connectivity is fairly affected by failures the adaptive mechanism was able to accommodate them. Despite of only one set of randomly generated failure is consid-

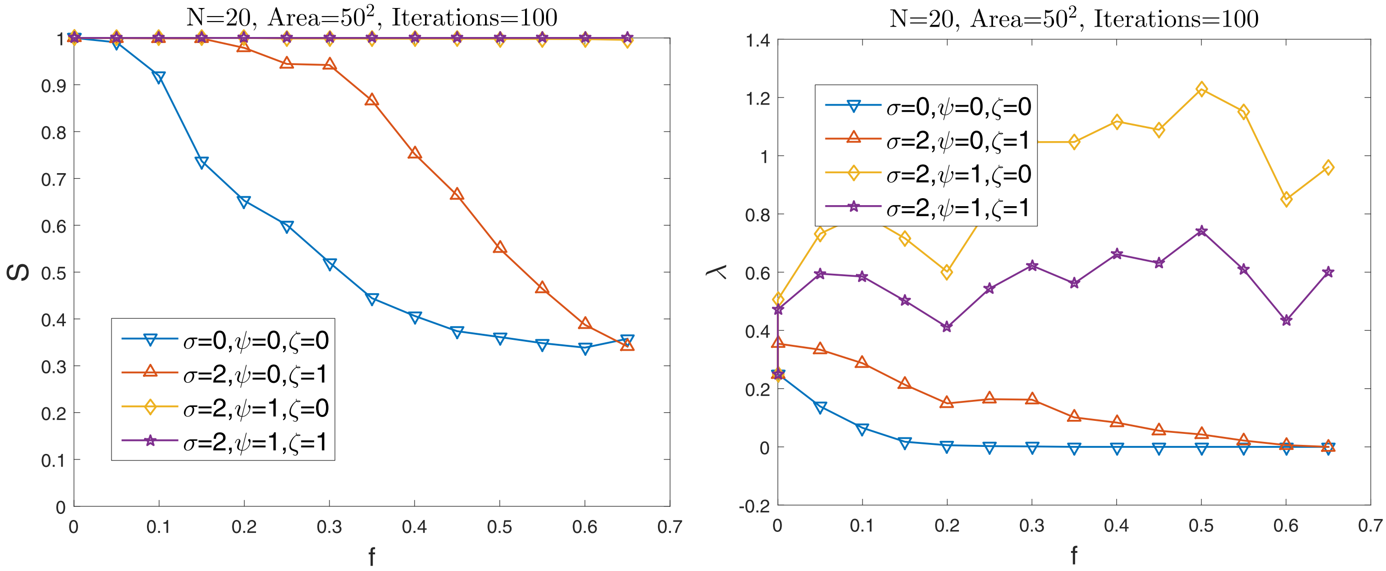


Fig. 9. The giant component (S) and the algebraic connectivity (λ) evolution in a fault-prone scenario.

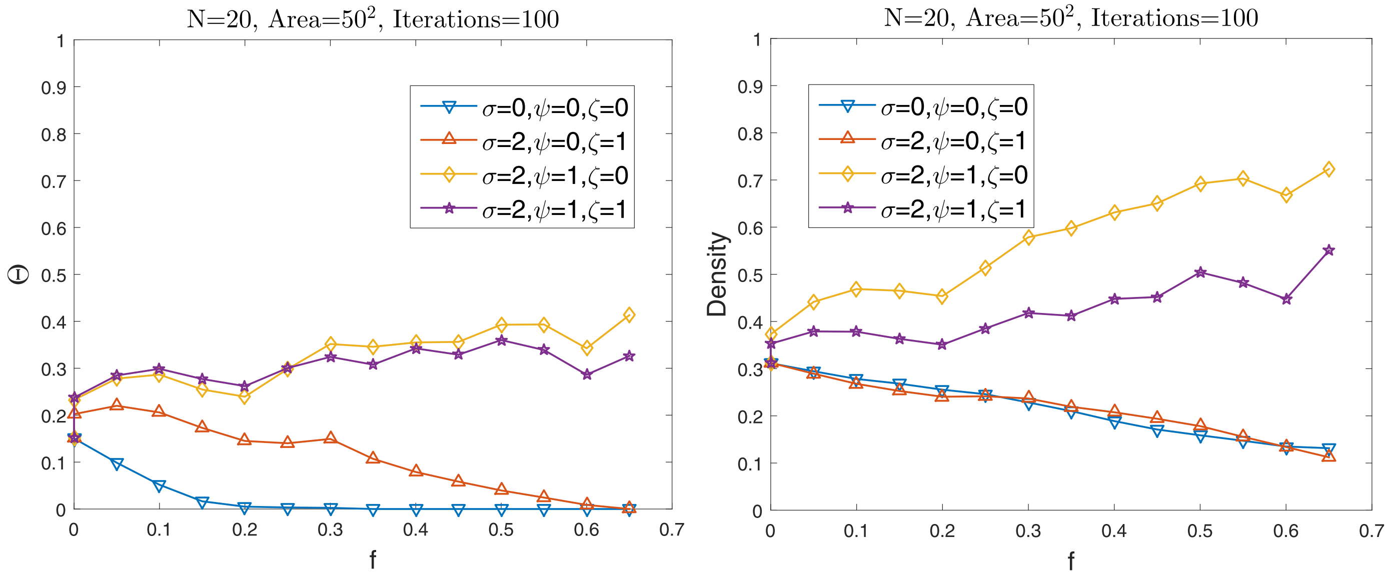


Fig. 10. The robustness level (Θ) and network density evolution in a fault-prone scenario.

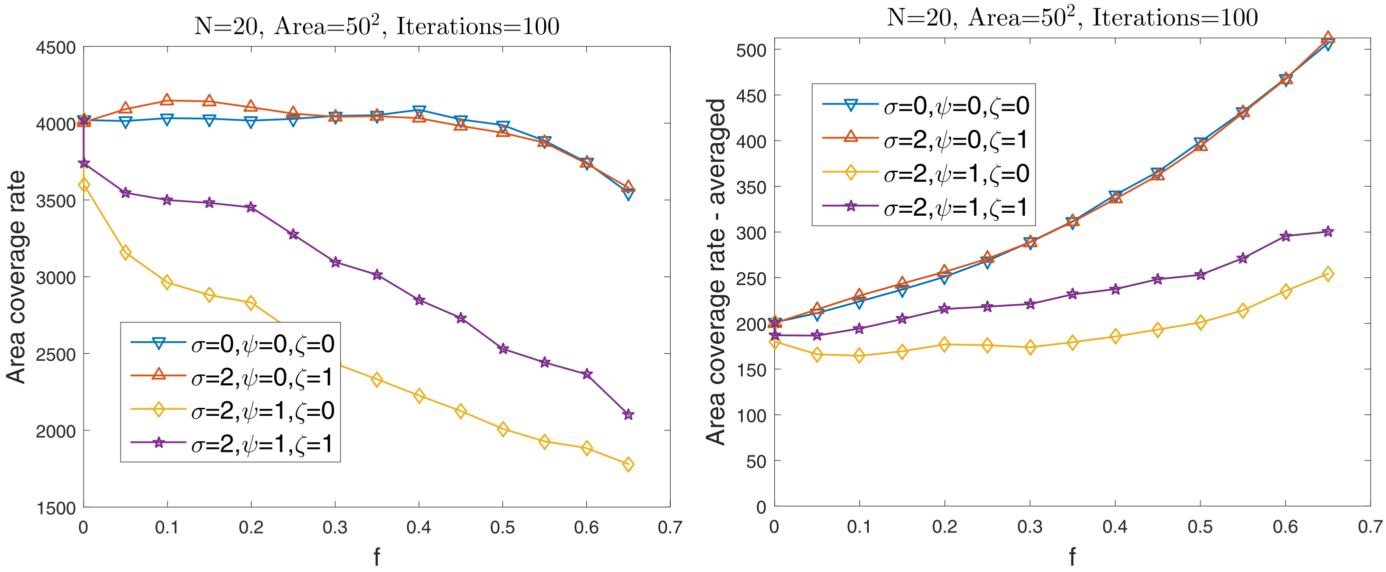


Fig. 11. The area coverage rate and the average coverage area rate evolution in a fault-prone scenario.

ered here, in [45] the effectiveness of the robustness control law was demonstrated under diverse failure time distributions, including cases where those failures are concentrated over short time spans.

Fig. 9 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.

The robustness level performance also demonstrates that networks could tolerate more robot failures before fragmentation, even in the presence of the additional area coverage control law (Fig. 10). It is important to notice that, despite similar performance regarding the robustness level for ψ=1, the network density (on the right in Fig. 10) was not too affected because of the coverage mechanism action (ζ=1), which is a desirable behavior.

This is also reinforced by the area coverage and the giant component analysis: despite the total area coverage, depicted in Fig. 11 having decreased when the robustness control law was active, most of the robots were still connected (Fig. 9). Besides, the area coverage averaged by the number of robots indicates that they are more spread out in the environment, i.e., the area served by each robot, on average, increased (on the right in Fig. 11).

Figs. 12 and 14 depict the corresponding results for fault-free environments. They demonstrate that the action of the area coverage control law was able to mitigate the effect of letting robots close to each other, without negatively impacting the network robustness level (Fig. 13).

In general, the experiment achievements point out to the feasibility of setting the model gains according to the application requirements: to enhance the area coverage regardless the robustness to failure improvement (σ=2,ψ=0, ζ=1) or to focus on providing more resilient networks despite the tendency of robots to get closer to each other (σ=2,ψ=1,ζ=0). On the other hand, it is possible for both controllers to operate together to deliver a more balanced network topology (σ=2,ψ=1,ζ=1).

Fig. 15 illustrates the performance of the proposed model regarding each gain setting evaluated for the same initial network and time failure distribution. Each topology color represents the network topology at specific times [$t_i = 20, 40, 60, 80$] in a fault-prone scenario. The impact of each control law on the resulting

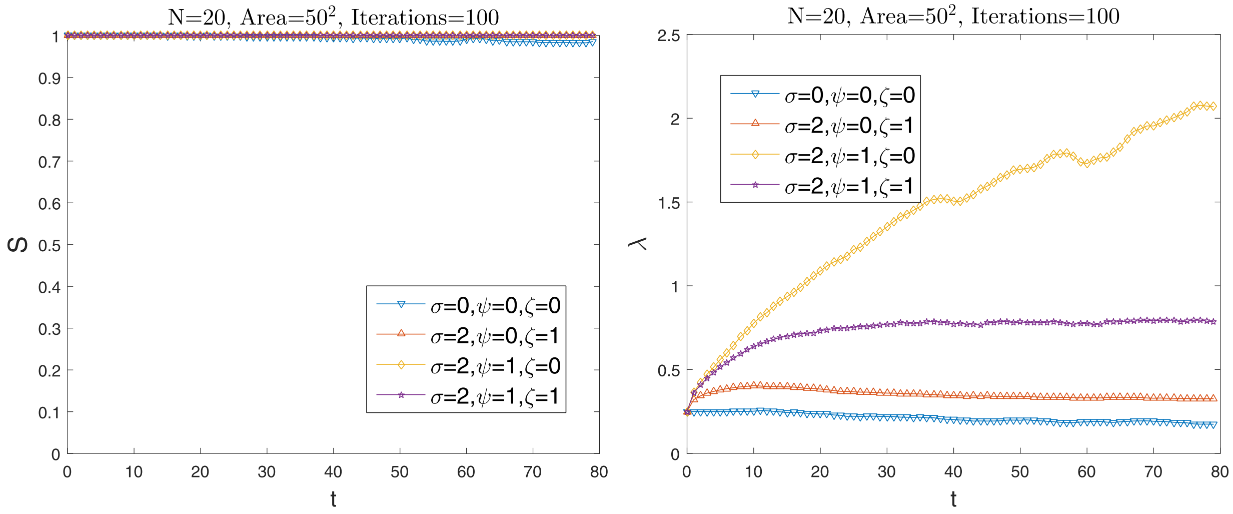


Fig. 12. The giant component (S) and the algebraic connectivity (λ) evolution in a fault-free scenario.

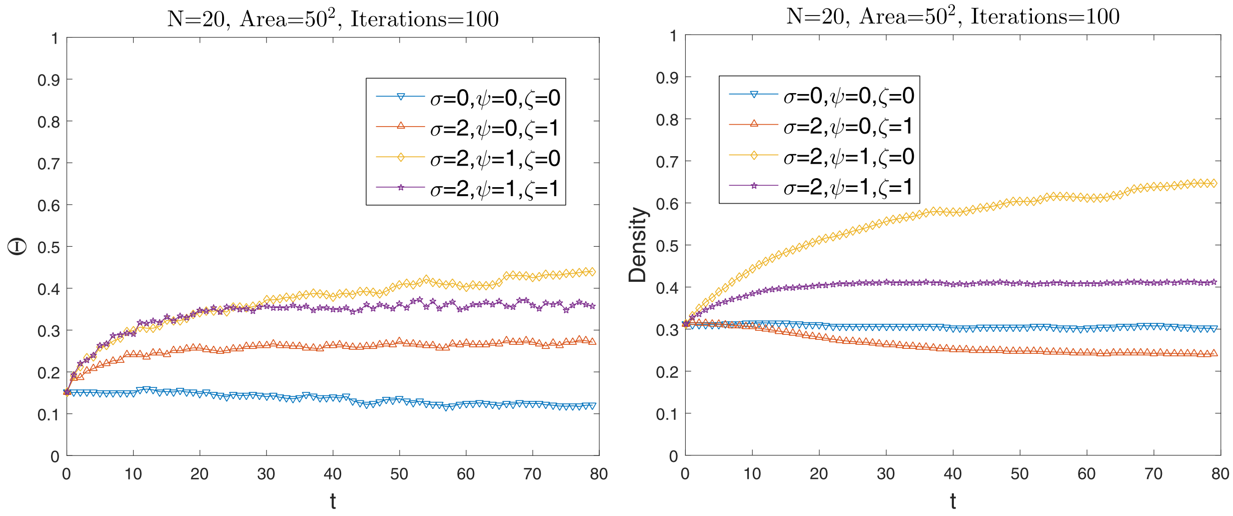


Fig. 13. The robustness level (Θ) and network density evolution in a fault-free scenario.

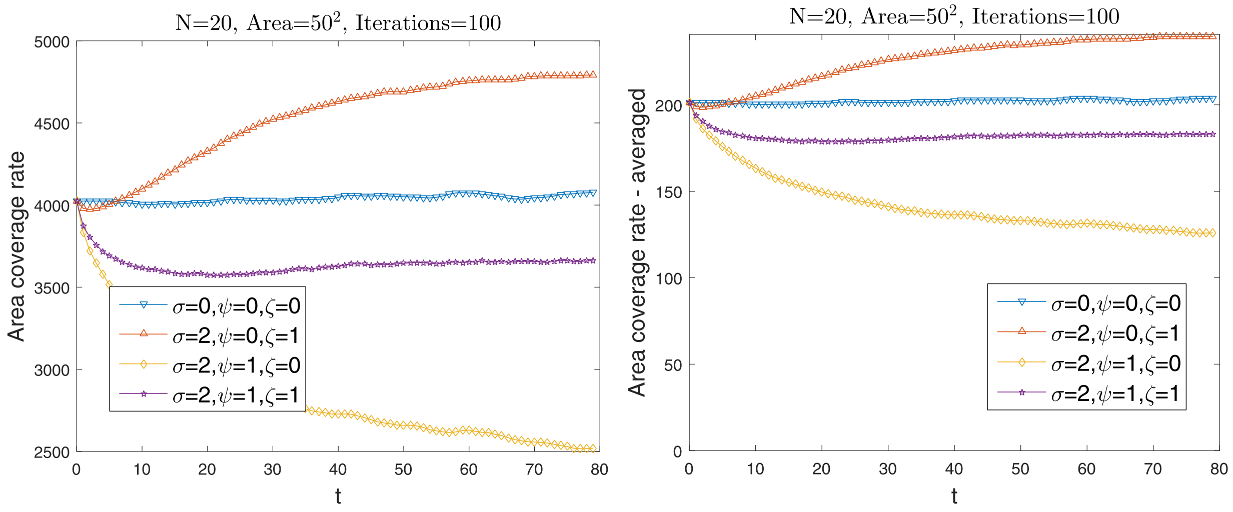


Fig. 14. The area coverage rate and the average area coverage rate evolution a fault-free scenario.

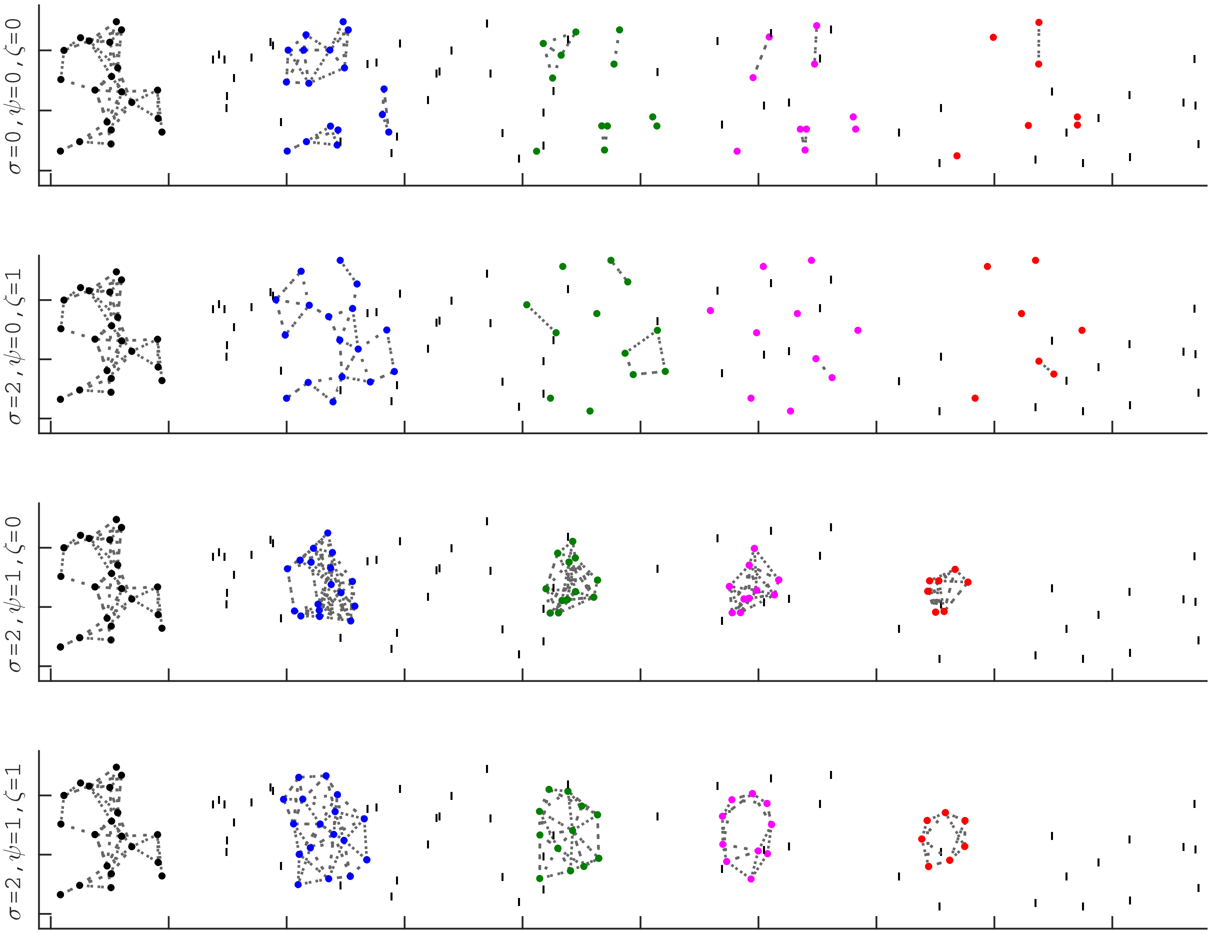


Fig. 15. Snapshots of a network configuration at different timestamps [$t_i = 20, 40, 60, 80$] in a fault-prone scenario for the same initial network topology and failure time distribution. Each figure presents the performance of a gain setting, from top to bottom: (σ=0,ψ=0,ζ=0), (σ=2,ψ=0,ζ=1), (σ=2,ψ=1,ζ=0) and (σ=2,ψ=1,ζ=1).

network topology is clear. The robustness to failures improvement mechanism plays an important role on delivering a more resilient network, which is not fulfilled by the coverage and connectivity mechanisms. In turn, the area coverage mechanism is able to increase the network efficiency in sensing a larger portion of the environment.

Some additional examples can be freely viewed online on http://www.comp.ita.br/~airgroup/papers/01/.

7. Conclusions and future work

In this paper we presented a model that combines several control laws that aim at providing ad-hoc multi-robot networks with means to produce efficient topology formations regarding the application requirements. The continuous-time model encompasses mechanisms for connectivity maintenance, collision-avoidance, robustness to failures w.r.t connectivity improvement, and area coverage enhancing.

A previous model combining connectivity maintenance, collision avoidance and robustness to failures improvement was already proposed and validated in [23]: the connectivity maintenance and the collision avoidance allow robots to keep connected while avoiding to collide with each other and with obstacles, and the robustness to failures is able to deliver more resilient networks. Here, we propose an extension of this model by adding a mechanism that aims at enhancing the area covered by the robots. 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. For overcoming that, we adopted an approach for generating Centroidal Voronoi Tessellations (CVT) based on local information, 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.

Simulation results demonstrated the feasibility of combining several strategies as a means of providing more efficient network topology formations. The gain settings are the main responsible for tuning the model behavior according to the expected outcomes.

As future work, we intend to propose adaptive gain setting according to the network state that can significantly improve the model already proposed, heading towards self-adaptive network topologies. Another interesting evolution is the design of an efficiency-based model regarding some cost function, such as communication efficiency or energy-consumption. This means aggregating mechanisms to assess the impact of the integrated model actions to some parameter of efficiency. Thus, the system might try not only to adjust the topology formation to the best configuration, but also to maintain a system efficiency property at a desired level.

Regarding multi-robot applications, it is necessary to validate the proposal presented here on real robot platforms.

Acknowledgments

The authors thank FAPESP for the financial support to carry out this research (procs. no. 2012/25058-9, 2014/ 13800-8 and 2013/13447-3).

References

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, D. Burschka, Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue, Robot. Autom. Mag., IEEE 19 (3) (2012) 46–56.

[2] U. Witkowski, M.A.M. El Habbal, S. Herbrechtsmeier, A. Tanoto, J. Penders, L. Alboul, V. 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.

[3] G. Xu, W. Shen, X. Wang, Applications of wireless sensor networks in marine environment monitoring: A survey, Sensors 14 (9) (2014) 16932–16954.

[4] Z. Wang, L. Liu, M. Zhou, Protocols and applications of ad-hoc robot wireless communication networks: An overview, Future 10 (2005) 20.

[5] A.F. Winfield, Distributed Autonomous Robotic Systems 4, Springer Japan, Tokyo, pp. 273–282.

[6] W. Ye, R.T. Vaughan, G.S. Sukhatme, J. Heidemann, D. Estrin, M.J. Mataric, Evaluating control strategies for wireless-networked robots using an integrated robot and network simulation, in: Proceedings ICRA. IEEE International Conference on Robotics and Automation, vol. 3, 2001, pp. 2941–2947.

[7] E. Almaas, A.L. Barabasi, Power laws in biological networks, in: E.V. Kroonin, Y.I. Wolf, G.P. Karev (Eds.), Power laws, scale-free networks and genome biology, Springer Science, 2006, pp. 1–11.

[8] S. Wuchty, Small worlds in rna structures, Nucl. Acids Res. 31 (2003) 1108–1117.

[9] D.J. Watts, Small Worlds : The Dynamics of Networks between Order and Randomness (Princeton Studies in Complexity), Princeton University Press, 2003.

[10] P.S. Dodds, R. Muhamad, D.J. Watts, An experimental study of search in global social networks, Science 301 (5634) (2003) 827–829.

[11] J.H. Jones, M.S. Handcock, An assessment of preferential attachment as a mechanism for human sexual network formation, Proc. R. Soc. London B 270 (1520) (2003) 1123–1128, doi:10.1098/rspb.2003.2369.

[12] S. Carpenter, B. Walker, M. Anderies, N. Abel, From metaphor to measurement: resilience of what to what? Ecosystems 4 (8) (2001) 765–781.

[13] J. Rak, Resilient Routing in Communication Networks, first ed., Springer Publishing Company, Incorporated, 2015.

[14] A. Berns, S. Ghosh, Dissecting self-* properties, in: Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, 2009, pp. 10–19, doi:10.1109/SASO.2009.25.

[15] G. Di Marzo Serugendo, M.-P. Gleizes, A. Karageorgos, Self-organising Software: From Natural to Artificial Adaptation, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 7–32. 10.1007/978-3-642-17348-6_2

[16] Y. Cao, 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, 2010, pp. 4744–4755.

[17] M.A. Hsieh, A. Cowley, V. Kumar, C.J. Talyor, Maintaining network connectivity and performance in robot teams, J. Field Rob. 25 (1) (2008) 111–131.

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

[19] D.V. Dimarogonas, K.H. Johansson, Bounded control of network connectivity in multi–agent systems, IET Contr. Theor. Appl. 4 (2010) 1751–8644.

[20] F. Morbidi, A. Giannitrapani, D. Prattichizzo, Maintaining connectivity among multiple agents in cyclic pursuit: A geometric approach, in: Proceedings of the IEEE International Conference on Decision and Control, 2010, pp. 7461–7466.

[21] L. Sabattini, N. Chopra, C. Secchi, Decentralized connectivity maintenance for cooperative control of mobile robotic systems, Int. J. Rob. Res. (SAGE) 32 (12) (2013) 1411–1423.

[22] C. Ghedini, C. Secchi, C.H.C. Ribeiro, L. Sabattini, Improving robustness in multi-robot networks, in: Proceedings of the IFAC Symposium on Robot Control (SYROCO),Salvador, Brazil, 2015.

[23] C. Ghedini, C.H.C. Ribeiro, 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, 2016.

[24] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, D. Rus, Voronoi coverage of non-convex environments with a group of networked robots., in: ICRA, IEEE, 2010, pp. 4982–4989.

[25] E.T.K. Sang, Voronoi diagrams without bounding boxes, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, II-2/W2, 2015.

[26] C. Godsil, G. Royle, Algebraic Graph Theory, Springer, 2001.

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

[28] S. Wasserman, K. Faust, D. Iacobucci, Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences), Cambridge University Press, 1994.

[29] R. Soukieh, I. Shames, B. Fidan, Obstacle avoidance of non-holonomic unicycle robots based on fluid mechanical modeling, in: Proceedings of the European Control Conference, Budapest, hungary, 2009.

[30] D. Lee, A. Franchi, H. Son, C. Ha, H. Bulthoff, P. Robuffo Giordano, Semi-autonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles, IEEE/ASME Trans. Mechatron. 18 (4) (2013) 1334–1345.

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

[32] P. Robuffo Giordano, A. Franchi, C. Secchi, H.H. Bülthoff, A passivity-based decentralized strategy for generalized connectivity maintenance, Int. J. Rob. Res. 32 (3) (2013) 299–323.

[33] A. Sangwan, R.P. Singh, Survey on coverage problems in wireless sensor networks, Wireless Pers. Commun. 80 (4) (2015) 1475–1500, doi:10.1007/s11277-014-2094-3.

[34] M. Argany, M.A. Mostafavi, F. Karimipour, C. Gagné, A GIS Based Wireless Sensor Network Coverage Estimation and Optimization: A Voronoi Approach, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 151–172. 10.1007/978-3-642-25249-5_6

[35] G. Dai, H. Lv, L. Chen, B. Zhou, P. Xu, A novel coverage holes discovery algorithm based on voronoi diagram in wireless sensor networks, Int. J. Hybrid Inf. Technol. 9 (3) (2016) 622–642.

[36] A. Gusrialdi, T. Hatanaka, M. Fujita, Coverage control for mobile networks with limited-range anisotropic sensors, in: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, 2008, pp. 4263–4268, doi:10.1109/CDC.2008.4739007.

[37] J. Stergiopoulos, A. Tzes, Voronoi-based coverage optimization for mobile networks with limited sensing range - a directional search approach, in: 2009 American Control Conference, 2009, pp. 2642–2647, doi:10.1109/ACC.2009.5160709.

[38] C.-F. Huang, Y.-C. Tseng, The coverage problem in a wireless sensor network, Mobile Netw. Appl. 10 (4) (2005) 519–528, doi:10.1007/s11036-005-1564-y.

[39] M. Schwager, J. McLurkin, J.-J. E. Slotine, D. Rus, From Theory to Practice: Distributed Coverage Control Experiments with Groups of Robots, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 127–136. 10.1007/978-3-642-00196-3_15

[40] S. Sedighian Kashi, M. Sharifi, Coverage rate calculation in wireless sensor networks, Computing 94 (11) (2012) 833–856, doi:10.1007/s00607-012-0192-1.

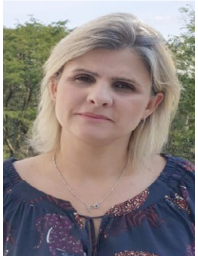
[41] T.-W. Sung, C.-S. Yang, Voronoi-based coverage improvement approach for wireless directional sensor networks, J. Netw. Comput. Appl. 39 (2014) 202–213. https://doi.org/10.1016/j.jnca.2013.07.003.

[42] D.M. Mount, Computational Geometry, Dept. of Computer Science - University of Maryland, 2005.

[43] S. Lloyd, Least squares quantization in pcm, IEEE Trans. Inf. Theor. 28 (2) (2006) 129–137, doi:10.1109/TIT.1982.1056489.

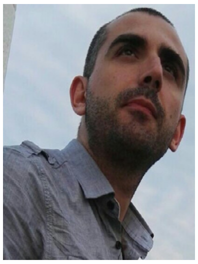
[44] C. Ghedini, C.H.C. Ribeiro, L. Sabattini, Improving the fault tolerance of multi-robot networks through a combined control law strategy, in: Proceedings of the International Workshop on Resilient Networks Design and Modeling (RNDM),Halmstadt, Sweden, 2016.

[45] C. Ghedini, C. Ribeiro, L. Sabattini, Toward fault-tolerant multi-robot networks, Networks 70 (4) (2017) 388–400, doi:10.1002/net.21784.



Cinara Ghedini is a research associate at the company Energias. She received her M.Sc. in Computer Science in 2000 and her D.Sc. in Engineering and Computer Science in 2012. Between 2012 and 2014 she held a fellow research position at the Aeronautics Institute of Technology, Brazil. In 2015 she had been a Visiting Researcher at University of Modena and Reggio Emilia, Italy. Her main research interests involve complex networks, machine learning, failure tolerant systems and multi-agent robotics.

Carlos H.C. Ribeiro holds a M.Sc. degree from Federal University of Rio de Janeiro and a Ph.D. degree in Electrical and Electronical Engineering from Imperial College London, 1998. Since 1998 he has been working at the Aeronautics Institute of Technology, Brazil, where he is associate professor and pro-rector for undergraduate studies. His research interests include Machine Learning (especially algorithms for autonomous learning), Multiagent Robotics, Complex Networks and Engineering Education. Dr. Ribeiro is a researcher of the Brazilian National Scientific Council (CNPq) since 2001.



Lorenzo Sabattini is an Assistant Professor at the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy, since 2013. He received his B.Sc. and M.Sc. in Mechatronic Engineering from the University of Modena and Reggio Emilia (Italy) in 2005 and 2007 respectively, and his Ph.D. in Control Systems and Operational Research from the University of Bologna (Italy) in 2012. In 2010 he has been a Visiting Researcher at the University of Maryland, College Park, MD (USA). His main research interests include multi-robot systems, decentralized estimation and control control, and mobile robotics. He is one of the founding cochairs of the IEEE RAS Technical Committee on Multi-Robot Systems: he has served as the corresponding co-chair since its foundation, in 2014. He has been serving as Associate Editor for the IEEE Robotics and Automation Letters since 2015, and for IEEE Robotics and Automation Magazine since 2017.