Robust Area Coverage with Connectivity Maintenance

Luca Siligardi†, Jacopo Panerati†, Marcel Kaufmann*, Marco Minelli†,
Cinara Ghedini†, Giovanni Beltrame*, and Lorenzo Sabattini†

Abstract—Robot swarms herald the ability to solve complex
tasks using a large collection of simple devices. However, engi-
eering a robotic swarm is far from trivial, with a major hurdle
being the definition of the control laws leading to the desired
globally coordinated behavior. Communication is a key element
for coordination and it is considered one of the current most
important challenges for swarm robotics. In this paper, we study
the problem of maintaining robust swarm connectivity while
performing a coverage task based on the Voronoi tessellation
of an area of interest. We implement our methodology in a team
of eight Khepera IV robots. With the assumptions that robots have
a limited sensing and communication range—and cannot rely
on centralized processing—we propose a tri-objective control
law that outperforms other simpler strategies (e.g. a potential
based coverage) in terms of network connectivity, robustness to
failure, and area coverage.

I. INTRODUCTION

Lowering costs, collaborative performance, and aug-
mented reliability through inherent redundancy are just some
of the attractive qualities of multi-robot systems. Robot
swarms, specifically, are very large multi-robot systems
composed by moderately simple devices. They carry the
promise of fully exploiting biomimetics and the economy of
scale. Inspired by nature, swarm robotics can benefit from
the application of knowledge from several different fields,
including software engineering and distributed systems [1]. In
multi-robot systems, a crucial element of effective collective
performance is reliable communication. For any two robots
to be able to talk, the underlying communication network
must enforce global connectivity. In the literature, the value
of the algebraic connectivity is often used as a proxy [2] (see
Figure 1). Yet, preserving connectivity per se is typically
not enough: we want robots to actually do something—and
this is often a spatially distributed task. The problem of
optimally deploying a group of mobile robots in a given
environment is generally referred to as a coverage problem.
This class of problems has applications in several domains,
such as multi-robot systems used for search and rescue [3],
deployment of mobile sensor networks [4], or ocean sensing
and sampling [5].

In our previous work [6], we implemented a coverage task
using a contribution based on the Lennard-Jones potential—a
molecular interaction model. Simulation results [7], however,
suggest that a Voronoi tessellation-based coverage approach
could help improve the robustness of the network—that is,
the system’s ability to mitigate the effects of node failures
through predictive actions that avoid vulnerable topological
configurations [8].

The contribution of this paper is two-fold: (i) first, we
implement the methodology from [8]—in particular, its Vor-
onoi tessellation-based coverage control—in a real multi-
robot setup; (ii) second, we compare our new results with
those from a previous robotic implementation exploiting a
simpler potential-based coverage approach. By doing so, we
are able to (i) validate the insights obtained from simulations
in [8] and (ii) establish the superior performance—especially
in terms of robustness—of Voronoi-based coverage w.r.t.
potential-based coverage. In what follows, Sections II to IV
are dedicated to a review of the existing literature and
the background theory of our methodology; Sections V
to VI describe our robotic implementation and experiments;
our findings are reported in Section VII; and Section VIII
concludes the paper.

Fig. 1. Networking in multi-robot systems can be described with the aid of
graph theory through the adjacency A, degree D, and Laplacian L matrices.
The second-smallest eigenvalue of L is also called algebraic connectivity
λ2 and it reveals whether the graph is connected.

II. RELATED WORK

Juliá et al.’s survey paper [9] summarizes the main ap-
proaches proposed in the literature to spread groups of robots
in an environment for exploration purposes. Several methods
start by performing a segmentation of the environment and
subsequently planning the motion of the robots to maximize
the coverage of the area [10]. Semantic information on the most relevant areas can be used, as described in [11], to provide guidance to the robots, thus achieving a non-uniform coverage of the environment. Deployment of the robots in the environment can be achieved by carefully combining attraction and repulsion control laws [12], exploiting, for instance, artificial potential fields [13], such as the one defined by Lennard-Jones [1] (see Equation 13).

While these methods are effective for guaranteeing stability of the group (e.g., cohesiveness, convergence), they generally do not provide any guarantee of uniformity or reliability in area coverage. Hence, several area segmentation methods have been developed [14] to assign each robot to a portion of the entire environment, in such a way that area is covered effectively and dependably. Techniques based on Voronoi tessellation are often used because they provide an optimal partitioning of the environment [15]. Furthermore, decentralized control strategies can be developed to asymptotically achieve area coverage based on centroidal Voronoi tessellation [16]. This makes the technique particularly suitable for multi-robot and swarm systems. Enhanced control strategies have been also developed for deploying robots to cover non-convex environments [17], [18].

As it is often the case in multi-robot applications, area coverage control strategies are usually developed under constraints entailed by the topology of the robots’ communication graph. In particular, preservation of connectivity as the system evolves is often a desired feature. In real-robot systems, exchange of information can be based on limited-range communication devices. Hence, mobility of the robots may lead to disconnections. Few works in the literature, when considering a multi-robot exploration problem, allow the communication graph to temporarily lose connectivity [19]–[21]. The path of the robots is, however, carefully planned so that connection is restored at some point in the future. While these approaches guarantee a certain level of flexibility, the motion of the robots needs to be carefully planned in advance, which can not always be achieved in unknown and harsh environments.

To overcome these issues, we need to impose constraints on the robots’ motion to guarantee the preservation of the communication graph as the system explores. Several strategies can be found in the literature to preserve all the links of an initially connected communication graph [22], [23], or some global properties of the graph itself [24]–[26].

It is worth noting that, when dealing with real-world robotic systems, guaranteeing connectivity maintenance is often not enough—in particular when considering robots that are subject to failures. In fact, connectivity preservation per se does not prevent the system to evolve into configurations where a single point of failure can be detrimental for the overall graph connectivity. It is then necessary to devise robust control strategies to avoid the onset of special nodes whose failure could lead to network fragmentation.

### III. System Model

In this work, we consider a multi-robot system comprising \( N \) robots able to communicate with one other within a communication radius \( R \). The resulting communication topology can be represented by an undirected graph \( G \) in which each robot is a node and each communication link between two robots is an edge. Let each robot’s state be its position \( p_i \in \mathbb{R}^m \) and let \( p = [p_1^T \ldots p_N^T]^T \in \mathbb{R}^{Nm} \) be the state vector of the multi-robot system. For the sake of generalization, we consider weighted graphs, where edge weights \( w_{ij} \) are defined as follows:

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

with \( e^{-(R^2)/(2\sigma^2)} = \Delta \), where \( \Delta \) is a small predefined threshold. Each robot is modelled as a single integrator system, whose velocity can be directly controlled:

\[
  \dot{p}_i = u_i,
\]

where \( u_i \in \mathbb{R}^m \) is the control. Recent research tackle the problem of connectivity for nonholonomic robots [27], [28]. For each robot, the control input must be set so that a global objective can be achieved. In the context of multi-robot environments, very often robots need to cooperatively follow multiple goals. For example, connectivity maintenance is a preponderant factor for information exchange. When dealing with failible robots, it is necessary to avoid vulnerable topological configurations to increase robustness. These objectives were extensively assessed in simulation [7], [8]. In this work, we aim at (i) validating our simulation results in a real robotic set-up and (ii) investigating whether such approach can outperform previous real-robot implementations [6].

### IV. Combined Control Law

The following combined control law aims at providing means for robots in a multi-agent system to coordinate strategies to optimize multiple properties of their interconnect topology. The main goal is to improve the network topology with regard to connectivity maintenance, robustness to failure, and area coverage. Using the kinematic model in Equation 2, we consider each robot to be controlled by means of a control input defined as the superposition of three different terms:

\[
  u_i = \sigma u_i^c + \psi u_i^r + \zeta u_i^a
\]

where \( u_i^c \) is the connectivity maintenance control law; \( u_i^r \) is the control law improving the robustness of the network to robots’ failures; \( u_i^a \) is the control law improving the area coverage; \( \sigma, \psi, \zeta \) are control gains—setting them to zero leads to the removal of the effect of the corresponding law.
A. Connectivity maintenance

In algebraic graph theory, the second-smallest eigenvalue \( \lambda_2 \) of the Laplacian matrix of a graph \( G \) is also called algebraic connectivity \( \lambda_2 \) and it reveals whether/how connected the network is [2]. Imposing \( \epsilon > 0 \) to be the desired lower-bound for the value of \( \lambda_2 \), we design a control strategy to ensure that \( \lambda_2 \) never drops below \( \epsilon \). As in [25], the following energy function can be used for a decentralized implementation:

\[
V(\lambda_2) = \begin{cases} 
\coth(\lambda_2 - \epsilon) & \text{if } \lambda_2 > \epsilon \\
0 & \text{otherwise}
\end{cases}
\]  

(4)

Referring to (2), the \( u_i^C \) control law is defined as follows:

\[
u_i^C = -\frac{\partial V(\lambda_2)}{\partial p_i} = -\frac{\partial V(\lambda_2)}{\partial \lambda_2} \frac{\partial \lambda_2}{\partial p_i}
\]

(5)

B. Robustness to robots’ failures

A centralized formulation of a graph’s robustness was introduced in [29]. Let \( \{v_1, \ldots, v_N\} \) be the list of nodes ordered by descending value of betweenness-centrality [29]. Let \( \varphi < N \) be the minimum index \( i \in [1, \ldots, N] \) such that, removing nodes \( \{v_1, \ldots, v_i\} \) leads to disconnecting the graph. Then, the robustness level \( \Theta \) is defined as:

\[
\Theta = \frac{\varphi}{N}
\]

(6)

Our decentralized robustness improvement strategy relies on a metric for estimating the magnitude of the topological vulnerability of a node by means of information acquired from its 1-hop and 2-hops neighbours. The vulnerability level of a node \( i \) regarding failures is given by:

\[
P_b(i) = \frac{|\text{Path}_{\beta}(i)|}{|\Pi(i)|}
\]

(7)

where \( |\Pi(i)| \) is the number of \( i \)'s 1-hop and 2-hops neighbours, and \( |\text{Path}_{\beta}(i)| \) is the number of nodes that are exactly at 2-hops from node \( i \) and relying on at most \( \beta \) 2-hops paths to communicate with \( i \). The purpose of a control strategy is to increase the number of links of a potentially vulnerable node \( i \) towards its 2-hop neighbours that are in \( \text{Path}_{\beta}(i) \). Let \( x_i^b \in \mathbb{R}^m \) be the barycentre of the positions of the robots in \( \text{Path}_{\beta}(i) \), the control law for robustness improvement is defined as follows:

\[
u_i^C = \xi_i \frac{x_i^b - p_i}{\|x_i^b - p_i\|} \propto (C_{V_i} - p_i)
\]

(8)

where \( \alpha(t) \) is the linear velocity of the robots. The parameter \( \xi_i \) is introduced to take into account the vulnerability state of a node \( i \): \( \xi_i = 1 \) if node \( i \) identifies itself as vulnerable, \( \xi_i = 0 \) otherwise—and it is defined as:

\[
\xi_i = \begin{cases} 
1 & \text{if } P_b(i) > r \\
0 & \text{otherwise}
\end{cases}
\]

(9)

where \( r \in (0,1) \) is a random number drawn from a uniform distribution. It is worth noting that (7) provides a decentralized methodology for each robot to evaluate its vulnerability level. The control law \( u_i^C \) was proven in [25], [30] to guarantee positiveness of the generalized connectivity in a disturbance-free environment.

C. Voronoi coverage

In general, the coverage problem states that robots should be distributed in the environment in a way that maximizes the total monitored area. 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. Voronoi tessellations are commonly applied in coverage area problems [17], [31], [32]. A Voronoi tessellation is a subdivision of a plane into cells based on the proximity of a set of points. Let \( P = [p_1, p_2, \ldots, p_n] \) 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

\[
V(i) = \{ q | \|p_i q\| < \|p_j q\|, \forall j \neq i \}
\]

(10)

where \( \|pq\| \) denotes the Euclidean distance between points \( p \) and \( q \) [33]. Let \( V(i) \) be the Voronoi cell corresponding to a robot located at \( p_i \). A team of \( N \) robots at positions \( P = [p_i]_{i=1}^N \in \mathbb{R}^2 \) navigate in a bounded polygonal environment \( \Omega \subseteq \mathbb{R}^2 \), where \( \Omega \) is a closed set with the boundary \( \partial \Omega \). Given a positive density function \( \phi : \Omega \rightarrow \mathbb{R}^+ \), the centroid of \( V(i) \) is defined as

\[
C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} x \phi(x)dx
\]

(11)

where \( M_{V_i} \) is the cell mass [17]:

\[
M_{V_i} = \int_{V_i} \phi(x)dx
\]

Then, the Voronoi-based approach presented in [17] was adopted. The simple first-order dynamics can be expressed as:

\[
u_i^V = \dot{p}_i \propto (C_{V_i} - p_i)
\]

(12)

The overall control in Equation 3 was validated in simulation in fault-free and fault-prone scenarios using different gain combinations [7]. Results demonstrated the feasibility of having simultaneous controls to achieve more resilient networks able to enhance their sensing area while maintaining the network connectivity. The impact of failures on the network connectivity was minimized by having the robustness improvement control law active. It is also important to emphasize that the coverage mechanism based on Voronoi tessellation was able to produce a more robust topology regarding failures compared to scenarios where only the connectivity is active. Nonetheless, for successive failures the robustness mechanism is essential to avoid fragmentation.

D. Lenard-Jones coverage

In [6], the control strategy from Equation 3 was implemented and evaluated for the first time in real robots substituting the Voronoi-based coverage contribution with a simpler potential-based diffusion approach. Rather than \( u_i^C \), the control law exploited contribution \( u_i^L \) defined as:

\[
u_i^L = -\tau \sum_{k} \left( \frac{a \cdot \delta^2}{x_{ik}^{a+2}} - 2 \cdot \frac{b \cdot \delta}{x_{ik}^{b+1}} \right)
\]

(13)

where \( x_{ik} \) is the distance between robots \( i, k \) and parameters \( \tau, i, \text{and } \delta \) represent the depth and distance from the origin.
of the potential’s minimum, respectively; \( a \) and \( b \) were set to 4 and 2. The considerations concluding Subsection IV-C suggest that Voronoi-based coverage has the power to outperform this potential-based coverage approach. However, because the two approaches can result in different geometrical configuration at convergence (see Figure 2), one of the objectives of this work is to discover whether they can produce comparable area coverage.

![Figure 2](image1)

**Fig. 2.** Voronoi tessellation of a uniform random distribution of points (a) and an incomplete triangular lattice (b); disk coverage of a uniform random distribution of points (c) and an incomplete triangular lattice (d).

![Figure 3](image2)

**Fig. 3.** ARGoS multi-physics simulator (a); K-team Khepera IV robot (b); OptiTrack Prime 13 camera (c); MIST Laboratory’s robotic arena (d).

**V. IMPLEMENTATION AND PORTING TO BUZZ**

Coding of the control law described in Section IV was done in Buzz\(^1\). Buzz [34] is a multi-paradigm scripting language designed to create composable programs for collections of heterogeneous robots. One of the advantages of using Buzz is the ability to rely on virtual stigmergy [35] for information dissemination. Preliminary simulation and debugging were carried out in the multi-physics environment of ARGoS [36] (see Figures 3-a and 4). Thanks to its virtual machine (VM), Buzz can run on multiple software and hardware platforms, thus, we were able to seamlessly transfer the Buzz code from ARGoS simulator to our robot of choice (i.e., K-team Khepera IV). With regard to the implementation’s nitty-gritty, since robots can only use local coordinates, we elected a leader robot to spread the information necessary for all the robots to reason in a virtual common coordinates system. This is effectively done using Dijkstra’s algorithm to compute the shortest paths between the leader and every other node. Both Dijkstra’s algorithm and Voronoi tessellation have naive implementations running in quadratic time complexity w.r.t. the number of robots \( O(N^2) \) and more sophisticated implementations running in \( O(N \log(N)) \) time.

![Figure 4](image3)

**Fig. 4.** Final configurations of a multi-robot system comprising eight footbot robots using different combinations of gains for Equation 3. They show the slightly different equilibria entailed by variations in these parameters. All simulations starts from a common initial pose (top-left) in ARGoS.

**VI. ROBOTIC EXPERIMENTS**

Going from simulations to hardware, sensible performance degradation can occur—the so-called “reality gap” [37]. Hence, the importance of actual robotic implementations. Our robot model lies on the assumptions that (i) robots have a limited communication range, (ii) they can only perceive information in their local coordinate systems, and (iii) they can exploit the “situated communication” model [38]. The robotic platform of our experiments is the K-team Khepera

---

\(^1\)https://github.com/MISTLab/Buzz
The monitored arena (see Figure 3-d) is a 2 meters by 2 meters square with eight Khepera IV robots comprising the robot team. Each robot has 512MB of RAM and can be controlled through Buzz, running on an 800MHz ARM Cortex-A8 Processor with an onboard 14cm diameter and 6cm height cylindrical two-wheel robot. More importantly, each Khepera IV is a fully-fledged Linux computer running on a 800MHz ARM Cortex-A8 Processor with 512MB of RAM and it can be controlled through Buzz's Virtual Machine. In all of our experiments, we used a multi-robot team comprising eight Khepera IVs.

The tracking of the robots' positions is achieved through four OptiTrack Prime 13 cameras (shown in Figure 3-c). The monitored arena (see Figure 3-d) is a 2 meters by 2 meters square. The OptiTrack system captures the x and y positions of the robots, as well as their yaw. This information, in an absolute frame of reference, is then translated into local coordinates before being fed to each Khepera IV. Situated communication is emulated over a Wi-Fi network using Blabbermouth. This software accesses the information made available by OptiTrack and distributes it to the robots, respecting the communication range—65cm in our experiments—and visibility constraints to re-create a fully distributed environment. This introduces a simplification, in that robots cannot obstruct each other's field of view.

We selected the most relevant combinations of gains (σ, ψ, ζ)—whose static [39] and time-varying optimizing [6] we studied before—and run over 20 new experiments lasting ~20′ each. These experiments were evaluated against four performance metrics, i.e., algebraic connectivity λ₂, robustness Θ, area coverage (in m²), and the optimization metric introduced in [6] to find the ideal gains σ, ψ, ζ:

\[
\text{performance} = \lambda_2(t)A(t)
\]

where λ₂(t) and A(t) are the algebraic connectivity and the area coverage at time t, respectively.
Lennard-Jones Potential  Voronoi Tessellation

Performance Coverage (m²)

\[ \chi_2 \Theta \]

Cover. (m²) Performance

\[ \begin{array}{c|c|c|c|c}
\text{Lennard-Jones Potential} & \text{Voronoi Tessellation} \\
\hline
\text{0.82} & \text{0.5} \\
0 & \theta \\
\text{3.71} & \text{3.6} \\
\text{2.49} & \text{3.56} \\
\end{array} \]

Fig. 9. Sample means \( \mu \) and population standard deviations \( \sigma \) (as error bars) of algebraic connectivity \( \chi_2 \), robustness \( \Theta \), coverage (in m²), and performance metric in experiments using Lennard-Jones potential-based (left) and Voronoi tessellation-based (right) coverage.

In this work, we presented the results of the implementation in a real-life multi-robot system—comprising eight Khepera IV robots—of a combined control law for the improvement of area coverage and network robustness that also preserves algebraic connectivity of a robotic team. The control law contribution improving area coverage exploited the Voronoi tessellation of the robots’ arena. We compared our results with those of a previous implementation exploiting a different, potential-based approach to area coverage. In Section VII, we explained how our results both substantiate the finding achieved through simulation in [8] and suggest that Voronoi-based coverage outperforms the potential-based coverage in terms of robustness and connectivity with negligible detriment to the explored area.

VIII. CONCLUSIONS AND FUTURE WORK

A selection of most significant experiment runs is presented in Figures 5 to 9. Cumulative statistics for all the experiments are given in Table I. The figures, as well as the table, also contain an excerpt of the results of [6]. These were obtained using the same control law, gains \( \sigma, \psi, \zeta \), and the Lennard-Jones potential-based coverage law in Equation 13.

The results confirm the simulation findings in [7]. One of the advantages of using Voronoi coverage is the improvement of the network robustness (see Figure 6). By increasing the coverage gain, the robustness level also improves—yet, Voronoi coverage per se would not be able to accommodate successive failures and this is when the robustness contribution matters. A possible explanation is that Voronoi coverage control, unlike Lennard-Jones coverage, is more gentle towards unevenly spaced configurations. On the other hand, we remarked that Voronoi tessellation-based coverage is sensitive to the way the cells are bounded. With \( \zeta = 1 \), a low \( \sigma \) can lead to a good coverage performance at the expense of the value of \( \chi_2 \) without resulting in loss of connectivity. Another (expected) result is the overt trade-off between algebraic connectivity and area coverage (see Figures 5 and 7). In fact, \( \sigma = 2 \) has a negative impact on the area the robots can cover. We also observe that, with the setup from Section VI, \( \sim 4m^2 \) is the coverage upper bound, where the robotic team starts to lose connectivity. With regard to the area coverage performance of the potential-based approach and the Voronoi tessellation-based one, we found that these are comparable—possibly with a slight advantage for the former (see Figure 7). Both connectivity and robustness—especially the latter—improves when using Voronoi-based coverage instead. In Table I we also report the results of the Student’s t-test confirming the statistical significance of the comparisons in Figure 9 for \( \chi_2 \) and \( \Theta \), while implying that the difference in area coverages is likely inconsequential.
