# Consensus-Based Distributed Connectivity Control in Multi-Agent Systems

Karlo Griparic, *Member, IEEE*, Marsela Polic, *Student Member, IEEE*, Marko Krizmancic, *Student Member, IEEE*, and Stjepan Bogdan, *Senior Member, IEEE*

***Abstract*—This paper addresses a distributed connectivity control problem in networked multi-agent systems. The system communication topology is controlled through the algebraic connectivity measure, the second smallest eigenvalue of the communication graph Laplacian. The algebraic connectivity is estimated locally in a decentralized manner through a trust-based consensus algorithm, in which the agents communicate the perceived quality of the communication links in the system with their set of neighbors. In the presented approach, link qualities represent the weights of the communication graph from which the adjacency matrix is estimated. The Laplacian matrix and its eigenvalues, including the algebraic connectivity, are then calculated from this local estimate of the global adjacency matrix. A method for network topology control is proposed, which creates and deletes communication links based on the Albert-Barabási probabilistic model, depending on the estimated and referenced connectivity level. The proposed algebraic connectivity estimation and connectivity maintenance strategy have been validated both in simulation and on a physical robot swarm, demonstrating the method performance under varying initial topology of the communication graph, different multi-agent system sizes, in various deployment scenarios, and in the case of agent failure.**

***Index Terms*—Coordination, distributed control, multi-agent systems, trust-based consensus.**

## I. Introduction

A LARGE research interest in the field of multi-agent systems, exploiting the scalability, robustness, and efficiency of such systems, is not surprising. One of the prominent applications is in modeling robotic teams as multi-agent systems, which enables solving a wide variety of real-world problems in different domains. Promising results are already obtained in the exploration of unknown environments [1], manipulation [2], [3], transportation [4], surveillance [5], search-and-rescue [6] and interaction with animal societies [7]–[9]. One of the core challenges is enabling such assemblies towards realizing a common objective despite their limited communication, sensing, computing, and actuating capabilities. Various decentralized control algorithms have been developed over years, such as protocol for rendezvous [10], formation control [11], [12] and consensus [13]–[15]. Most of these algorithms, however, assume ideal conditions, such as continuous availability of communication. Networking and topology control methods have emerged in an attempt to answer conditions more likely to happen in real applications that encompass various disturbances, such as deterioration of the communication quality, obstacles, agent failures, malicious agents, and similar.

Recognizing the similarities of the multi-robot systems with the network systems has provided useful insights for achieving more resilient robot coordination. For example, it has been shown that information exchange between robots has a direct impact on the system performance, including convergence speed and effectiveness of cooperation [16], [17]. However, a broad information exchange requires increasing the number of communication links, which can lead to deteriorated control, for example in case of delays due to token exchange. In other cases, increasing the number of communication links is not even possible, with a given communication network. Additionally, the effects that outside and inside disturbances and shocks have on the system performance should be considered if a particular method strives to be implemented in a real-world scenario.

The communication network of the multi-robot system can, in many cases, be described by an undirected graph, where each link in the network represents a communication channel between two robots. In that case, graph theory mathematical formalism can be used to provide insights into the global properties of the underlying network topology, an approach exploited in this work. Furthermore, this enables adaptation of the system communication topology to retain the desired functionality even under disturbances.

### A. Literature Overview

In the research of distributed connectivity control for robotic systems, most early control laws were incorporated

Manuscript received March 21, 2021; revised November 17, 2021; accepted December 19, 2021. Date of publication January 7, 2022; date of current version May 23, 2022. This work of doctoral students Marsela Polic and Marko Krizmancic were supported by the Young Researchers Career Development Project—training of doctoral students of the Croatian Science Foundation funded by the European Union from the European Social Fund. This work was supported in part by the EU-ICT Project ASSISIbf - Animal and robot Societies Self-organise and Integrate by Social Interaction bees and fish under Grant 601074, in part by the EU-ICT Project WatchPlant - Smart Biohybrid Phyto-Organisms for Environmental In Situ Monitoring under Grant 101017899, and in part by the EU-ICT Project AeRoTwin - Twinning coordination action for spreading excellence in Aerial Robotics under Grant 810321. Recommended for acceptance by Dr. Gang Yan. *(Corresponding author: Karlo Griparic.)*

Karlo Griparic is with the Department of Engineering, Juraj Dobrila University of Pula, 52100 Pula, Croatia (e-mail: karlo.griparic@unipu.hr).

Marsela Polic, Marko Krizmancic, and Stjepan Bogdan are with the LARICS Laboratory for Robotics and Intelligent Control Systems, Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia (e-mail: marsela.polic@fer.hr; marko.krizmancic@fer.hr; stjepan.bogdan@fer.hr).

Digital Object Identifier 10.1109/TNSE.2021.3139045

within the motion controllers [18]–[22], similar to the approach in the sensor networks, regarding sensor communication radius control [23]. Such Euclidean distance-based approaches are quite restrictive in applicability because of unpredictable variations in wireless networks performance. In these approaches, connectivity control focuses on preserving initially created topology, without considering real phenomena in wireless protocols, such as fading and shadowing. A comprehensive overview of real wireless networks for multi-agent systems is presented in [24]. In contrast to these approaches, the method proposed here does not rely on distance measures, but considers the communication channel quality instead.

By definition, resilience as a property of a technical system encompasses concepts related to the ability of the system to rebound from trauma and to return to an equilibrium state. The traumas considered here are mostly related to faults and/or malfunctions of parts of the system [25]. In the case of a multi-agent system network architecture, resilience can be seen as the sustained system's ability to adapt to failures caused by either intrinsic source (e.g. malfunction of a communication link), or outside source (e.g. adversarial attack on a communication link). The measures of network resilience, among others, include probabilistic measures of network fault tolerance (probability of a node failure) [26], disruption propagation time, and the time required to return to the equilibrium state (connectivity level prior to the disruption) [27]. The focus of this paper is not to quantitatively measure the network resilience - in our case resilience can be seen as a boolean value, i.e. a network that encompasses the method presented herein is resilient (able to return to the equilibrium state, defined through the Fiedler value, after trauma, represented by an agent failure). If the network cannot return to the equilibrium state after trauma, then it is not resilient.

A large body of literature exists that addresses resilience in terms of underlying communication networks in multi-agent systems. Generally, two questions arise: i) *how to determine*, and ii) *how to maintain* the structure/connectivity of the network. Answers to these questions should provide techniques applicable for real-time deployment, in a decentralized/distributed manner. The most commonly used methods are thus based on various variations of consensus algorithm [28]. For example, in [29], the authors designed a consensus protocol based on the redundancy of information exchanged within the local neighborhood, and used this redundancy as a property for analyzing the behavior of the algorithm. Other examples include discarding of faulty measurements, described in [30], or in the case of robotic systems, a dynamic topology adjustment through robot motion control, presented in [31]. These methods use a consensus algorithm in connectivity control, i.e. they answer to the question of *connectivity maintenance*, whereas in our work, the consensus is used for the estimation of the communication graph structure, i.e. as an answer to the question of how to *determine the connectivity*. Another novelty is in using the trust-based consensus algorithm variant. Among other methods for connectivity determination, a recent one can be found in [32], where authors combine a switching signal that determines the network topology with the consensus algorithm in order to exclude spoofed nodes and achieve agreement. Another interesting approach is suggested in [33]. Authors used frequency response of Wiener filters to pinpoint spurious links in the network, and assure algorithm convergence using only valid links in linear consensus. It is important to mention that the convergence speed is directly related to the number of links in the graph, [13], [34], usually measured with *algebraic connectivity*, that will be formally introduced in the next section.

The above-mentioned methods only consider keeping the underlying communication graph connected, without taking into account the cost of maintaining such connectivity. In the case of wireless networks, the cost comes in the form of energy required to communicate, and the computation time for algorithm execution. Clearly, the notions of cost and connectivity level introduce contradictory objectives in the network connectivity control. In order to increase the network resilience (decrease the time required to return to the connectivity level prior to disruption), the number of links should be increased. On the other hand, an increased number of links would in general increase the cost. Hence, the topology design, conciliating the algebraic connectivity maximization, and network cost minimization, can be considered as a convex optimization problem, [35], [36]. This work does not attempt to find a solution to this optimization problem, since the optimal criteria depend on the actual use case. Instead, a higher-level control module is assumed to define the desired graph structure and to reference it to the connectivity maintenance method in the form of desired Fiedler value. Our method then aims to realize a communication graph configuration with the desired connectivity. Some examples of these higher-level optimization solutions are given in the remainder of this section.

In [37] authors proposed Mixed Integer Semidefinite Program (MISDP) to find the optimal network design, where the specified performance is met with minimized communication cost, compared to a complete communication graph. The cost considerations lead to the optimization method presented in [38], where the power transmitted by each node of a network was adapted in order to maximize the network connectivity. The goal was to allow each node to estimate and track the algebraic connectivity of the underlying expected graph, which is in line with the objective of the method that we propose herein. The main drawback of the approach described in [38] is the assumption that transmitted power was the same for all nodes in the network. Some of the recently obtained results consider robot-to-robot interactions, as shown in [39]. The objective of the presented reconfiguration strategy is to compute an optimal k-connectivity graph that would minimally constrain pairwise robot motion (defined by the edges of the graph) required by the original task.

### B. Paper Contributions

Within this work, a novel communication graph connectivity control method is proposed for a multi-agent system, using the algebraic connectivity (Fiedler value) as the measure of graph connectivity. Fiedler value of the underlying communication graph is obtained from the adjacency matrix that is estimated in a decentralized manner, through a trust-based consensus algorithm. In contrast to existing work, we propose a method that takes into account the actual quality of communication links and is able to re-establish graph connectivity in case of an agent failure. Furthermore, based on the estimation of the algebraic connectivity, a distributed connectivity feedback control method is proposed. The method does not require for the initial underlying communication graph to be known to the agents in the group - local information about the neighborhood only is sufficient. We propose a modification of the Albert-Barabási probabilistic model [40] for communication links creation and deletion. Since we assume that each communication link introduces additional energy and computation cost, the method aims to ensure desired performance (in terms of algebraic connectivity) by balancing the number of realized links among those physically feasible. Finally, as we show in the results section, the proposed method is successfully validated in the simulation and through real-world experiments with a swarm of underwater robots. To summarize, the contributions of this paper are:

1) A trust-based consensus algorithm for estimation of the Fiedler value of the communication graph in multi-agent systems.
2) A distributed connectivity control law for maintaining the desired algebraic connectivity.
3) A modified Albert-Barabási probabilistic model that determines the procedure of managing the number of agent links.

### C. Paper Structure

The remainder of this paper is structured as follows. First, in Section II we give basic notations and preliminaries on graph theory, followed by Section III dedicated to details of a multi-agent system and formulation of the distributed connectivity control problem. In Section IV, we present the distributed connectivity control method for a multi-agent system. Section V is dedicated to a description of the decentralized trust-based algorithm for algebraic connectivity estimation, along with a nonlinear observation function representing the quality of a communication link. Verification of the proposed method, in different scenarios in simulation and in the real world, is given in Sections VI and VII. Final remarks and conclusions are discussed in concluding Section VIII.

## II. Preliminaries on Graph Theory

Herein we introduce graph theory notations and basic concepts used in this paper. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices $\mathcal{V} = \{1, 2, \dots, n\}$, where $n$ is the total number of vertices in the graph, and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents connections between vertices. In general, a graph can be directed or undirected, depending on the assigned direction on its edges. The vertices that are adjacent to vertex $i$ are called the neighbors of $i$ and this set is denoted as $\mathcal{N}_i = \{v_j \in \mathcal{V} : e_{ij} \in \mathcal{E}\}$.

TABLE I
Bounds to Algebraic Connectivity Related to Basic Graph's Parameters

| Graph's parameter | Algebraic connectivity $\lambda_2$ |
|---|---|
| Number of vertices $n$ ($\mathcal{G} \neq \mathcal{K}_n$) | $\lambda_2 \leq n-2$ |
| Minimal degree $d_{min}$ | $2d_{min} - n + 2 \leq \lambda_2 \leq \frac{n}{n-1} d_{min}$ |
| Number of edges $m$ ($\mathcal{G} \neq \mathcal{K}_n$) | $\lambda_2 \leq (-1 + \sqrt{1+2m})$ |

The adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ defines the connections of the vertices in the graph. If vertex $i$ is adjacent to vertex $j$, then adjacency matrix element $a_{ij}$ is equal to 1, otherwise $a_{ij} = 0$. The adjacency matrix is symmetric $a_{ij} = a_{ji}$ for an undirected graph. If a graph $\mathcal{G}$ is weighted (which is the case investigated in this paper), then adjacency elements $a_{ij} \geq 0, \in \mathbb{R}$ if $e_{ij} \in \mathcal{E}$. The degree matrix $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$ is a diagonal matrix, where $d_i$ denotes the degree of vertex $i$, and is calculated as $d_i = \sum_{j=1}^{n} a_{ij}$. The Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{1}$$

with $l_{ii} = d_i$ and $l_{ij} = l_{ji} = -a_{ij}$.

Some of the properties of the Laplacian matrix are particularly interesting from the graph connectivity perspective. For an undirected weighted graph $\mathcal{G}$ with non-negative weights,

1) all eigenvalues of $\mathbf{L}$ have real non-negative values,
2) $\mathbf{L}$ is a positive semi-definite matrix.

Furthermore, for the column vector $\mathbf{1}$, with all elements equal to 1, and null vector $\mathbf{0}$, the following equality holds: $\mathbf{L1} = \mathbf{0}$. Thus, one eigenvalue of the Laplacian matrix is equal to zero and can be denoted by $\lambda_1 = 0$. For an undirected graph, the Laplacian matrix $\mathbf{L}$ is symmetric with real eigenvalues $\lambda_i$ that can be ordered in a non-descending order:

$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n.$$

The second smallest eigenvalue of the Laplacian matrix, $\lambda_2$, is called the Fiedler value or algebraic connectivity and represents a measure of connectivity of the associated graph. So, the graph is connected if and only if $\lambda_2 > 0$ [41].

For some well know graph families, such as star, cycle or complete graphs, the algebraic connectivity can be determined by exact expressions. For example, the algebraic connectivity of the complete graph $\mathcal{K}_n$ is equal to the number of vertices in the graph, $\lambda_2 = n$. Since, in general, connectivity increases with an increase in $\lambda_2$, it is useful to know the maximum Fiedler value $\lambda_{2max}$. For the complete graph $\lambda_{2max} = n$, but in the cases $\mathcal{G} \neq \mathcal{K}_n$, the upper bound of algebraic connectivity $\lambda_2$ depends not only on the number of vertices, but also on other parameters of the graph, including the number of edges and minimal degree of the graph, as shown in Table I (there are other graph parameters that impact the algebraic connectivity, but those are beyond the scope of this paper). For instance, Table II shows an insight into the correlation of the number of vertices with the algebraic connectivity for a star, cycle, and complete graphs. Additionally, a variation between

TABLE II
THE ALGEBRAIC CONNECTIVITY OF STAR, CYCLE AND COMPLETE GRAPHS RELATED TO NUMBER OF VERTICES

| Number of vertices $n$ | Star | | Cycle | | Complete graph | |
|---|---|---|---|---|---|---|
| | $\lambda_2$ | $m$ | $\lambda_2$ | $m$ | $\lambda_2$ | $m$ |
| 5 | 1 | 4 | 1.382 | 5 | 5 | 10 |
| 10 | 1 | 9 | 0.382 | 10 | 10 | 45 |
| 15 | 1 | 14 | 0.173 | 15 | 15 | 105 |
| 25 | 1 | 24 | 0.063 | 25 | 25 | 300 |
| 50 | 1 | 49 | 0.016 | 50 | 50 | 1225 |

the algebraic connectivity and the number of edges for different topologies can be observed. A comprehensive overview of the graph's spectral properties can be found in [41].

## III. Problem Description

Let $n$ denote the number of agents in the system, whose communication network topology can be described with a time-varying weighted undirected graph $\mathcal{G}$. We consider the system where the weights of the edges represent the quality of the communication links between agents. It is assumed that each agent can exchange information only with agents that belong to its set of neighbors. When the agents are mobile, which is the case for example in many cooperative multi-robot systems, the communication network topology is typically influenced by robots' positions. Additionally, the quality of the communication links can be affected by failures, attacks, noise, and obstacles, which can vary in time as well. In extreme cases, the quality can drop to values so low, that a particular communication link can be considered lost.

As mentioned, in a multi-agent system where a consensus algorithm is used to reach an agreement, the speed of convergence of the consensus protocol, $\beta$, in the worst case, is determined with algebraic connectivity $\lambda_2$ of the communication graph. More specifically [42]:

$$\beta = \frac{\lambda_n + \lambda_2}{\lambda_n - \lambda_2}. \tag{2}$$

We will return to notion of the consensus convergence time later in Section V-A.

When the spectral property of a graph is applied to a multi-agent system, then the algebraic connectivity plays a fundamental performance measure of the system. Therefore, in many networked multi-agent systems, a strategy to keep the algebraic connectivity at a certain level is necessary in order to achieve the desired global objective of the system. However, maximizing the algebraic connectivity in weighted undirected graphs, where weights represent relations between neighboring agents, is an optimization problem that is challenging to solve due to the limited computational and power capabilities of a single agent participating in decentralized multi-agent applications.

On the other hand, keeping the algebraic connectivity in the multi-agent system at a maximal level does not necessarily contribute to the quality of control, due to delays and redundancy of messages that such policies can cause. Hence, the goal of graph connectivity control is to keep $\lambda_2$ at a certain level.

In distributed and decentralized multi-agent systems, connectivity control is a non-trivial problem, since each agent is only aware of the connections with its neighbors and thus unable to directly determine the overall topology of the communication graph. In other words, information on the communication links should somehow be propagated through the network, so that each agent can estimate the underlying communication graph topology, and determine its Fiedler value. Formally, each agent $l$, at the discrete communication interval $k$, should calculate its local estimate $\mathbf{A}^l(k)$ of the adjacency matrix. Since each agent is able to communicate with its neighbors only, local estimates $\mathbf{A}^l(k)$ should be shared with all the agents in the group, so that eventually all the group members determine a common adjacency matrix of the graph,

$$\mathbf{A}^l(k) = \mathbf{A}, \forall l, k \to \infty. \tag{3}$$

How to ensure the convergence of (3) is the first problem that we address in this paper.

Once $\mathbf{A}$ is determined, each agent can calculate $\lambda_2$, and, depending on the $\lambda_2$ tracking error to the reference value, act accordingly. Here, we consider two actions - establishing and deleting communication links. These actions effectively change the underlying communication graph topology, and hence the value of $\lambda_2$ accordingly. It should be noted that within this work, we assume that the process of creation and deletion of arcs in the underlying graph is much slower than the proposed consensus protocol that is initiated every time a new set of arcs is added (deleted). Which of these actions (addition or deletion) should be taken, and when, is the second problem that we discuss herein.

In the remainder of the paper, we will first discuss the second presented problem, namely the connectivity control problem. This problem assumes that $\mathbf{A}$ is known (through whichever method), and that reference $\lambda_{2ref}$ is given. Then, we explain the proposed consensus-based algorithm for obtaining the $\mathbf{A}$.

## IV. Connectivity Maintenance

The main objective of connectivity maintenance is to control the algebraic connectivity of the underlying communication graph of a multi-agent system during operation. This could be done by changing the number of vertices and/or edges. Since we are considering the system where vertices represent agents, and edges represent communication links, the changes in the graph connectivity will be done by adding or removing edges, i.e. communication links.

Following the well-known conservative property of the algebraic connectivity with respect to the change of the number of links in the graph, we can assume that if a new graph is obtained by adding a link, then the new value of the algebraic connectivity will be equal to or greater than the $\lambda_2$ value for the original graph. The same logic can be drawn for a new graph obtained by removing a link - the new value of the algebraic connectivity will be equal to or less than the value of the

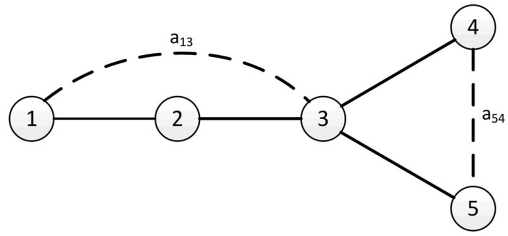


Fig. 1. An example of the graph topology with two added links causing different change of the algebraic connectivity value.

original graph. Even though our approach to connectivity maintenance - increasing and decreasing the algebraic connectivity - is based on these assumptions, it should be clear that the exact value of algebraic connectivity depends on other parameters as well, most significantly on the graph topology. Similar dependency holds for the rate of change of the algebraic connectivity when adding or removing a link.

In order to demonstrate this dependency, we return to the lower and upper bounds of the algebraic connectivity defined with respect to the minimal degree of the graph, $d_{min}$, (shown in Table I):

$$2d_{min} - n + 2 \leq \lambda_2 \leq \frac{n}{n-1} d_{min}. \quad (4)$$

Boundaries defined in (4) imply that in order to achieve the desired value of algebraic connectivity, $\lambda_{2ref}$, the minimal degree of the graph must be

$$d^*_{min} \geq \frac{n-1}{n} \lambda_{2ref}. \quad (5)$$

In other words, if a node, representing agent $l$, has a degree that is less than the minimum degree $d^*_{min}$ required by the desired connectivity $\lambda_{2ref}$, then, establishing a new communication link, i.e. adding a link to the graph, has no effect on the algebraic connectivity, if this new link is added to any other node but the node with the minimum degree.

In an example of the graph with 5 nodes, as shown in Fig. 1, we have chosen two different links, represented with dashed lines, that have been added to the original graph. The values of algebraic connectivity for the original graph, and for the graphs obtained after links $a_{13}$ and $a_{54}$ have been added, are given in Table III. Results clearly demonstrate that adding a single link can have a drastically different effect on the algebraic connectivity - while in the case of new link $a_{54}$ the value of $\lambda_2$ did not change, adding link $a_{13}$ almost doubled the algebraic connectivity.

### A. Connectivity Controller

Now we proceed with the description of the proposed method for adding and removing communication links. Firstly, it relies on an estimate of the adjacency matrix. As stated, we propose an approach where each agent $l$ calculates its local estimate of the adjacency matrix at time step $k$, $\mathbf{A}^l(k)$, using the algorithm that we describe later in Section V. Then, each agent determines $\lambda_2^l(k)$ and calculates the difference between the desired and the current (estimated) value of

TABLE III
THE ALGEBRAIC CONNECTIVITY OF DIFFERENT GRAPH'S TOPOLOGY WITH 5 NODES

| | **Original graph** | **Adding link $a_{54}$** | **Adding link $a_{13}$** |
|---|---|---|---|
| $\lambda_2$ | 0.5188 | 0.5188 | 1 |

**Procedure 1.** The Pseudo-Code of the Proposed Connectivity Maintenance Procedure Running on Agent $l$.

Initialize adjacency matrix estimate $\mathbf{A}^l$ with known local connections.
Initialize $K_{\lambda_2}$ according to Lemma IV.1.
**while** True **do**
  Exchange communication link qualities $a^l_{ij}$ with neighbors.
  Update the adjacency matrix estimate $\mathbf{A}^l$. (See Section V.)
  Calculate $\lambda^l_2$.
  Calculate $e^l_{\lambda_2}$ using (6).
  **if** $e^l_{\lambda_2} > K_{\lambda_2}$ **then**
    Add a link.
  **end**
  **if** $e^l_{\lambda_2} < -K_{\lambda_2}$ **then**
    Remove a link.
  **end**
**end**

the algebraic connectivity

$$e^l_{\lambda_2}(k) = \lambda_{2ref} - \lambda^l_2(k). \quad (6)$$

Due to the discontinuous nature of the graph connectivity ($\lambda_2$ does not change continuously with the number of edges in the graph), as well as due to unknown relation between $\lambda_2$ and the number of edges $m$ (the example shown in Fig. 1 demonstrated that graphs with the same number of edges have different $\lambda_2$), implementation of formal design methods for a linear controller for the tracking error $e^l_{\lambda_2}(k)$ is not possible (usually heuristic approaches are applied). Instead, we propose a classical approach based on the nonlinear feedback control theory. Firstly, since $d\lambda_2/dm \geq 0$, by increasing the number of edges $m$, the Fiedler value $\lambda_2$ will increase or remain the same. Then, if the tracking error $e^l_{\lambda_2}(k)$ is greater than user-defined value $K_{\lambda 2}$, an edge should be added to the graph, while in case the difference is below $-K_{\lambda 2}$, an edge should be removed from the graph. This approach corresponds with the typical three-level relay controller. The pseudo-code of the proposed feedback connectivity control strategy is given in the Procedure 1.

The proposed relay controller has a single parameter $K_{\lambda 2} > 0$ that should be determined so that

$$|e^l_{\lambda_2}(k)| < K_{\lambda 2}, \quad k \to \infty. \quad (7)$$

Again, due to the discontinuous nature of the graph connectivity, setting $K_{\lambda 2}$ too small could cause cyclic addition and removal of edges, that will, in turn, produce oscillations of $\lambda_2$.

*Lemma 4.1 (Lower bound of $K_{\lambda 2}$):* For (7) to hold, relay parameter should satisfy

$$K_{\lambda 2} > \min_{i,j}(f_i - f_j)^2; (f_i - f_j) \neq 0, \tag{8}$$

where $f_i$ and $f_j$ are $i$-th and $j$-th elements of normalized Fiedler vector $\mathbf{f}$ corresponding to $\lambda_2$.

*Proof:* To determine lower bound of $K_{\lambda 2}$ (to avoid oscillations of $\lambda_2$) we use normalized Fiedler vector $\mathbf{f}$, i.e. an eigenvector of Laplacian matrix that corresponds to $\lambda_2$,

$$\mathbf{Lf} = \lambda_2 \mathbf{f}. \tag{9}$$

From perturbation theory of symmetric matrices we have

$$\frac{\partial \lambda_2}{\partial l_{ij}} = \mathbf{f}^\mathbf{T} \frac{\partial \mathbf{L}}{\partial l_{ij}} \mathbf{f}. \tag{10}$$

Since

$$\mathbf{f}^\mathbf{T}\mathbf{Lf} = \sum_{i,j} l_{ij}(f_i - f_j)^2, \tag{11}$$

we get

$$\frac{\partial \lambda_2}{\partial l_{ij}} = (f_i - f_j)^2, \tag{12}$$

which states that change of algebraic connectivity due to change in $l_{ij}$ (that is actually same to change in $a_{ij}$ - see (1)) is equal to difference of corresponding Fiedler vector components. Hence, by finding the minimal difference (not equal to zero) between pairs of all entries of $\mathbf{f}$, we will determine the minimal change in $\lambda_2$ caused by addition/removal of a single link,

$$\Delta\lambda_{2\,min} = \min_{i,j}(f_i - f_j)^2, \tag{13}$$

So, to avoid oscillations in algebraic connectivity caused by addition/removal of links, i.e. to guarantee that (7) is satisfied, the relay controller parameter $K_{\lambda 2}$ should be greater than $\Delta\lambda_{2\,min}$. ■

In Lemma IV.1 only the lower bound of the controller parameter is considered since an upper bound of $K_{\lambda 2}$ does not influence the system dynamics but only the static error of the algebraic connectivity (larger $K_{\lambda 2}$, larger the gap from the reference value).

Further reference to Table I gives an interesting insight on $K_{\lambda 2}$. Namely, the third row in the table relates $\lambda_2$ and the number of edges $m$. Since this dependency is represented by a square root function, $d\lambda_2/dm$ will have a hyperbolic form, which means that when the graph has a relatively small number of edges, adding a single edge could significantly change the connectivity, while adding an edge when the graph has a relatively large number of edges would not significantly change $\lambda_2$.

### B. Adding/Removing Communication Links

Once the agent $l$ decides to remove or add a link, the question is which link should be removed or established. From the energy point of view, the most appropriate network topology for multi-agent systems would be the one that achieves the desired algebraic connectivity with a minimal number of links. This argument holds for attacks as well, since the lower number of communication links enables less opportunity of the attack. However, a minimal number of links makes the system sensitive to communication failures, since the loss of a single link might cause the loss of the connectivity property of the underlying graph. Hence, the appropriate value of $\lambda_{2ref}$ is a trade-off between a robust, secure, and energy effective underlying communication graph, which depends on the number of agents, the number of links, the network topology, and finally, on the intended application, i.e. required speed of convergence of the consensus protocol (2).

In the remainder of this section, we introduce a novel distributed connectivity maintenance method, based on the Albert-Barabási preferential model [40]. This simple model determines the probability that a new node will connect to an existing node in the network. In the Albert-Barabási model, similar to the real-world social networks, nodes with more existing connections are more likely to receive new links. We modified this model for integration into the connectivity control for decentralized multi-agent systems. It should be noted that aside from the Albert-Barabási preferential model, our approach can support other connectivity maintenance methods as well, including the most widespread network topologies such as Erdős-Rényi and Watts-Strogatz graphs. A set of randomly connected nodes, where the probability of connecting two nodes is equal in each step, constructs a random network, and can be described using Erdős-Rényi model introduced in [43]. On the other hand, using a model given by Watts-Strogatz [44] a network with highly clustered nodes can be described, where some nodes in the cluster are connected with other clusters in the network. Having in mind real multi-agent applications, the Watts-Strogatz small-world graph is not suitable for implementation due to the limited range of communication. Erdős-Rényi probabilistic model can be implemented in a similar manner to the Albert-Barabási preferential model. However, in experiments carried in the paper [45], authors have shown that the Albert-Barabási free-scale graph is more robust to node and link failures compared to other graph topologies.

For an agent $j$ that has a set of active neighbors $\mathcal{N}_l$ and is within the connection neighborhood $\mathcal{R}^l$ of an agent $l$, we define a probability that agent $l$ will connect to agent $j$ as

$$p_j^l(\kappa) = \frac{d_j(\kappa)}{\sum_{\mathcal{P}_l} d_j(\kappa)}, \tag{14}$$

where $d_j(\kappa)$ is the degree of agent $j$ and

$$\mathcal{P}_l = \{j: \quad j \in \mathcal{R}^l, \quad j \notin \mathcal{N}_l\}.$$

It should be mentioned that discrete time intervals for adding/removing communication links are longer than discrete communication intervals $T_d$. That is why in (14) we use $\kappa$ instead of $k$.

By applying the given method to the system of 5 agents, with graph topology shown in Fig. 2, one gets

$$\mathcal{N}_5 = \{3\}, \mathcal{R}^5 = \{2,3,4\}, \mathcal{P}_5 = \{2,4\},$$

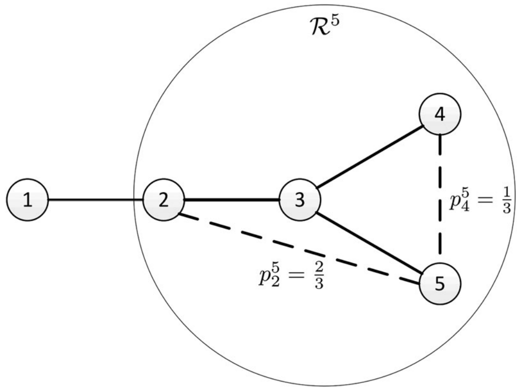


Fig. 2. A graph topology with preferential probability distribution when agent 5 should add a link.

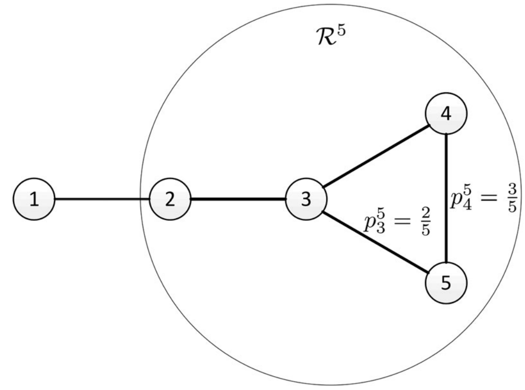


Fig. 3. A graph topology with preferential probability distribution when agent 5 should remove one of its links.

which gives

$$p_2^5 = \frac{2}{2+1} = \frac{2}{3}, \quad p_4^5 = \frac{1}{2+1} = \frac{1}{3},$$

hence the probability for agent 5 to connect to agent 2 is two times higher than to connect to agent 4. Following the same logic when the algebraic connectivity of a graph should be decreased, the agent would probably disconnect from the neighbor with the largest number of communication links. That is in direct contradiction with the fact that, in general, nodes with a higher degree contribute more to the graph connectivity. Hence, in the case of link removal for agent $l$, the higher probability for disconnection is given to the agent with the lowest degree. However, as the algebraic connectivity of a graph is lower bounded with the minimal degree $d^*_{min}$, that is determined from (5), agent $l$ is allowed to initiate removal of the link only in case $d_l > d^*_{min}$. The same holds for the neighboring nodes, so the probability of removal is calculated as

$$p_j^l(\kappa) = 1 - \frac{d_j(\kappa)}{\sum_{\mathcal{N}_l} d_j(\kappa)}, \quad d_j(\kappa) > d^*_{min}. \tag{15}$$

Using the same example of the multi-agent system of 5 agents, with graph topology shown in Fig. 3, assuming $\lambda_{ref}$ such that $d^*_{min} = 1$, let us consider situation when agent 5 initiates removal of one of its links. Having $\mathcal{N}_5 = \{3, 4\}$, according to (15)

$$p_3^5 = 1 - \frac{3}{3+2} = \frac{2}{5}, \quad p_4^5 = 1 - \frac{2}{3+2} = \frac{3}{5}.$$

As the direct consequence of (14), agents with more links are more likely to get new connections since nodes with higher degrees are contributing to higher network connectivity. However, as it was mentioned earlier, due to other criteria, such as energy constraints, we introduce the upper boundary $d^*_{max}$ for the number of connections each agent can have.

Finally, a complete failure of an agent $j$ should be discussed prior to describing the adjacency matrix estimation method. In that case, $d_j = 0$, hence $p_j^l = 0, \forall l$, i.e. none of the agents within $\mathcal{R}^j$ will try to link with the failed agent. As shown in the following section, this situation is perceived locally and is propagated through the graph. As a consequence, the failed agent is excluded from the group, meaning that the number of nodes in the underlying communication graph becomes $n - 1$, consequently changing the dimension of adjacency matrices $A^l$.

## V. Local Adjacency Matrix Estimation

As mentioned in the problem statement section II, when a global approach for algebraic connectivity estimation of a cooperating multi-agent system is used, each agent must estimate the topology of the complete communication graph, and the group must eventually reach a common value of the adjacency matrix. Hence, the question is how to update the elements of the local estimate $\mathbf{A}^l(k)$, in order to fulfill (3)? The simplest discrete update law that could be used is of the form

$$a_{ij}^l(k+1) = a_{ij}^l(k) + \Delta a_{ij}^l(k), \tag{16}$$

where $a_{ij}^l \in [0, 1]$ represents the weight of the communication link between the agent $i$ and $j$, perceived by agent $l$ (hence, $a_{ij}^l$ is an element of adjacency matrix $\mathbf{A}^l$). This discrete update law is executed once during a communication (sampling) interval (every $T_d > 0$ time units). It is assumed that the underlying graph does not contain self-loops, i.e. $a_{ii}^l = 0$.

### A. Trust-Based Consensus

The main challenge in this update law is how to determine $\Delta a_{ij}^l(k)$ in order to ensure convergence of the method. The approach proposed in this paper is built upon our previous work on a trust-based consensus algorithm presented in [46]. In that work, we assigned a trust value to each link in the network and introduced an algorithm that ensured the convergence of the agents' states, as well as agents' trusts, to a common value. Building upon other research conducted in the field of mission planning and scheduling in multi-robotic systems [47]–[50], we propose a modification to this approach, so that the quality of the communication link is considered the same way as we treated the trust in our previous work. In other words, we exploit the mathematical formalism developed in our earlier research on trust, with adaptations accounting for communication link quality instead. Herein, the observation function (introduced later in this section) does not depend on the state vector $\mathbf{x}$ - it is formed based on the number of data packets exchanged between agents, i.e. the quality of the communication link. However, from the mathematical formalism point of view, the quality of the communication links can be introduced into the system description through $a_{ij}$, i.e. in the same way as trust.

Furthermore, in the approach proposed in this paper, we allow for the communication links to fail, and new links to establish.

As described in Section II, the agents should have the ability to determine (and to reconfigure) the group topology in a decentralized fashion, based on agents' local views and estimates of weights (quality) of the communication graph. This implies that there is no central agent that has access to all the values of the adjacency matrix (representing the quality of communication links). Therefore, each agent should keep and update its own matrix with communication weights regarding every other agent in the group, including those that are not its neighbors. Under the assumption that each agent is capable of observing the weights of communication links with its neighbors, the update law should include this information as well.

In general, the consensus is written as

$$x_i(k+1) = \sum_{j\in\mathcal{N}_i} a_{ij}(k)[x_j(k) - x_i(k)], \tag{17}$$

or in matrix form,

$$\mathbf{x}(k+1) = -\mathbf{L}\mathbf{x}(k), \tag{18}$$

where $x_i$ describes the state of agent $i$, $x_j$ is the state of agent $j$, $a_{ij}$ represents level of relation between these two agents, and $\mathcal{N}_i$ defines a set of agents neighboring with agent $i$. The state could, for example, be the position or the orientation of a robot, environment temperature measured by the agent, or some other physical property, while the level of relation could be the trust between the agents, the attractive force between the agents, or any other relational property. In this paper, we assign the quality of the communication link to the relational property and, as stated in (16), we investigate how to update this property so that (3) is satisfied, i.e. to guaranty convergence of the update algorithm.

Closely related to the speed of convergence $\beta$, given in (2), the convergence time $T_n(\rho)$ of consensus (18) is defined as [51]:

$$T_n(\rho) = min\left\{\theta : \frac{\| x(k) - x^* \|_\infty}{\| x(0) - x^* \|_\infty} \leq \rho, k \geq \theta, \forall x(0) \notin c\mathbf{1}\right\} \tag{19}$$

where $\rho$ is a positive constant, $x^*$ stands for $lim_{k\to\infty}x(k)$, and $c\mathbf{1}$ is the set of vectors with equal components. In general, positive constant $\rho$ can take arbitrary small value, however, due to the fact that $T_n(\rho) \propto log(1/\rho)$, [51], the most common range is $0.01 \leq \rho \leq 0.1$, i.e. the time to consensus is equal to the time when the average absolute distance to $x^*$ drops below value in range 1% - 10% of the initial value, [52].

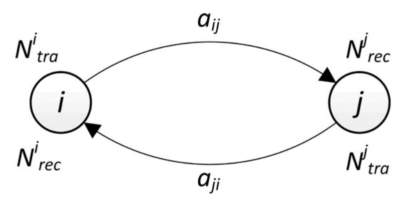


Fig. 4. An exchange of data used to determine the quality of the link between two nodes in a multi-agent system.

### *B. Communication Links Quality*

The main goal of the proposed method is to relate $\Delta a^l_{ij}(k)$ with the communication link quality. However, since the agent measurements might be erroneous, we also relate $\Delta a^l_{ij}(k)$ with the information on $a^l_{ij}(k)$ as perceived by other agents. Hence, the estimate of local adjacency matrix $\mathbf{A}^l$ is a function of both the neighboring environment perceived by the agent and the "opinions" of others in the group.

First, the part related to the measurements is derived. For this purpose, the basic model, shown in Fig. 4, is used to establish a measure of the communication link quality with packet loss computation (other measures can be used as well - the adjacency matrix update law (16) does not depend on the way the quality is measured). It is assumed that in a discrete-time step $k$ agent $i$ transmits a certain number of data packets, denoted $N^i_{tra}$, to agent $j$. At the same time agent $j$ receives the number of data packets $N^j_{rec}$, such that $N^j_{rec} \leq N^i_{tra}$. If $N^j_{rec} = N^i_{tra}$, than the communication link between agents $i$ and $j$ has the highest quality. The measure can be written as

$$\delta_{ij}(k) = N^i_{tra}(k) - N^j_{rec}(k). \tag{20}$$

Since in the same discrete time step, there is data exchange from agent $j$ to agent $i$ ($N^i_{rec} \leq N^j_{tra}$), we apply the same logic. Therefore,

$$\delta_{ji}(k) = N^j_{tra}(k) - N^i_{rec}(k). \tag{21}$$

This approach implies that in the general case, the quality of the links between two agents might differ, $\delta_{ij}(k) \neq \delta_{ji}(k)$, which leads to the directional communication graph ($a_{ij}(k) \neq a_{ji}(k)$) and the spectral properties of undirected graphs do not apply. To keep the underlying graph undirected we calculate a common quality of the link as the maximum of two values,

$$\delta^{ij}(k) = max[\delta_{ij}(k), \delta_{ji}(k)]. \tag{22}$$

Because in general, the measurement $\delta^{ij}(k)$ can take any positive value, while $a_{ij} \in [0, 1]$ as in [46] where we introduced the so-called observation function, in order to adjust the versatility of measured values to the range of the adjacency matrix weights. The form of the observation function is based on an assumption that measurements follow Gaussian law, thus the following exponential function is used:

$$\tau^l_{ij}(k) = e^{-\frac{(\delta^{ij}(k))^2}{\sigma_l^2}}, \tag{23}$$

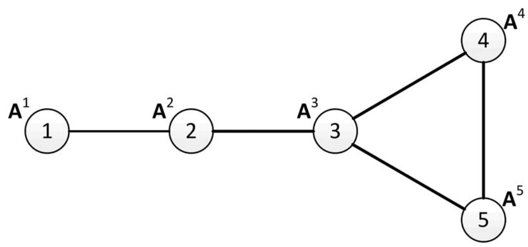


Fig. 5. The initial graph topology for a multi-agent system with 5 robots where $\mathbf{A}^l$ is local estimate of the global adjacency matrix.

where $\sigma_l$ is a constant that defines the rate of change of the exponential function. This constant should be determined based on the parameters that can influence the quality of communication links, such as the maximal number of data packets that can be exchanged between two agents in a discrete-time step, noise level, and probability of errors. It should be noted that the general form of the observation function (23) takes values in the interval [0,1] and allows negative measurements ($\delta^{ij}(k) < 0$), which is not the case in our proposed approach. When agent $l$ initiates a disconnection from its neighbor $j$, we set $\tau_{lj}^l(k) = 0$, as there will not be any new messages exchanged between them.

Now, the difference between the current observation of agent $l$ and the current value of its local adjacency matrix can be calculated as:

$$\epsilon_{ij}^l(k) = \tau_{ij}^l(k) - a_{ij}^l(k). \tag{24}$$

In general, one can allow agent $l$ to observe (or have an access to) the measurements between agents $i$ and $j$. However, here we assume that only the agents involved in data exchange are able to observe that exchange, and according to (20)–(23) we have

$$\tau_{ij}^i = \tau_{ij}^j = \tau_{ji}^i = \tau_{ji}^i. \tag{25}$$

Refering to (24), if $i$ and $j$ do not communicate, the measurements of transmitted and received packets of data are not available, and we set $\epsilon_{ij}^l(k) = 0, \forall l$.

### C. Adjacency Matrix Updates

We proceed further with determination of influence that estimates of other agents in the group have on the update law (16). To grasp the concept, we refer to Fig. 5, and consider the link between agents 4 and 5 in the local adjacency matrix of agent 3, i.e. we consider element $a_{45}^3$. According to the proposed consensus algorithm, agent 3 compares its value $a_{45}^3$ with neighboring agents which gives, $(a_{45}^2 - a_{45}^3)$, $(a_{45}^4 - a_{45}^3)$ and $(a_{45}^5 - a_{45}^3)$. The exchange of this information goes through neighboring communication links with varying qualities, all of which are accounted for, so that the influence is calculated as a linear combination, $a_{23}^3 \cdot (a_{45}^2 - a_{45}^3) + a_{43}^3 \cdot (a_{45}^4 - a_{45}^3) + a_{53}^3 \cdot (a_{45}^5 - a_{45}^3)$. Since by the assumption agent 3 cannot observe communication between agents 4 and 5, $\epsilon_{45}^3 = 0$.

Now we can state the main result of the paper.

*Theorem 5.1 (Adjacency matrix updates):* Under the assumption that the initial underlying communication graph contains a spanning tree, if $1 \geq \tau_{ij}^l(k) \geq 0$ (which according to (23) is always true), then update equation

$$\Delta a_{ij}^l(k) = \sum_{p \in \mathcal{N}_l} a_{lp}^l(k)(a_{ij}^p(k) - a_{ij}^l(k)) + \epsilon_{ij}^l(k), i \neq j. \tag{26}$$

guarantees that (3) holds, i.e.

$$a_{ij}^1(k) = a_{ij}^2(k) = \ldots = a_{ij}^n(k) = \tau_{ij}^i(k), k \to \infty. \tag{27}$$

*Proof:* Proof of the theorem follows logic presented in [46]. Rewriting (26) for node 1 ($l = 1$) gives (step $k$ is kept away for abbreviation)

$$\begin{aligned} \Delta a_{ij}^1 &= \sum_{p \in \mathcal{N}_1} a_{1p}^1(a_{ij}^p - a_{ij}^1) + \epsilon_{ij}^1 \\ &= -a_{ij}^1(a_{12}^1 + a_{13}^1 + \cdots + a_{1n}^1) \\ &\quad + \begin{bmatrix} a_{12}^1 & a_{13}^1 & \ldots & a_{1n}^1 \end{bmatrix} \begin{bmatrix} a_{ij}^2 \\ a_{ij}^3 \\ \ldots \\ a_{ij}^n \end{bmatrix} + \epsilon_{ij}^1, \end{aligned} \tag{28}$$

while for node 2 ($l = 2$) one gets

$$\begin{aligned} \Delta a_{ij}^2 &= \sum_{p \in \mathcal{N}_2} a_{2p}^2(a_{ij}^p - a_{ij}^2) + \epsilon_{ij}^2 \\ &= -a_{ij}^2(a_{21}^2 + a_{23}^2 + \cdots + a_{2n}^2) \\ &\quad + \begin{bmatrix} a_{21}^2 & a_{23}^2 & \ldots & a_{2n}^2 \end{bmatrix} \begin{bmatrix} a_{ij}^1 \\ a_{ij}^3 \\ \ldots \\ a_{ij}^n \end{bmatrix} + \epsilon_{ij}^2. \end{aligned} \tag{29}$$

According to the definition of degree matrix $\mathbf{D}$,

$$\begin{aligned} (a_{12}^1 + a_{13}^1 + \cdots + a_{1n}^1) &= d_1^1, \\ (a_{21}^2 + a_{23}^2 + \cdots + a_{2n}^2) &= d_2^2, \end{aligned} \tag{30}$$

while

$$\begin{aligned} \begin{bmatrix} d_1^1 & -a_{12}^1 & -a_{13}^1 & \ldots & -a_{1n}^1 \end{bmatrix} &= \mathbf{l_1^1}, \\ \begin{bmatrix} -a_{21}^2 & d_2^2 & -a_{23}^2 & \ldots & -a_{2n}^2 \end{bmatrix} &= \mathbf{l_2^2}, \end{aligned} \tag{31}$$

represent 1st row, $\mathbf{l}_1^1$, of $\mathbf{L}^1$, and 2nd row, $\mathbf{l}_2^2$, of $\mathbf{L}^2$. Now, by defining three vectors,

$$\mathbf{a}_{ij} = \begin{bmatrix} a_{ij}^1 \\ a_{ij}^2 \\ \ldots \\ a_{ij}^n \end{bmatrix}, \mathbf{\Delta a}_{ij} = \begin{bmatrix} \Delta a_{ij}^1 \\ \Delta a_{ij}^2 \\ \ldots \\ \Delta a_{ij}^n \end{bmatrix}, \epsilon_{ij} = \begin{bmatrix} \epsilon_{ij}^1 \\ \epsilon_{ij}^2 \\ \ldots \\ \epsilon_{ij}^n \end{bmatrix}, \tag{32}$$

we can generalize for all nodes in the system and write (26) in matrix form,

$$\mathbf{\Delta a}_{ij} = -\hat{\mathbf{L}}\mathbf{a}_{ij} + \epsilon_{ij}, \tag{33}$$

**Algorithm 2.** The Pseudo-Code of the Proposed Adjacency Matrix Estimation on Agent $l$.

**Input**: $a_{ij}^l(k), i, j \in [1, n]$
$a_{ij}^p(k), i, j \in [1, n], p \in \mathcal{N}_l$
$\delta^{ij}(k), i, j \in [1, n]$
$T_d, \sigma_l$
**Output**: $a_{ij}^l(k+1)$
**foreach** $i \in [1, n]$ **do**
  **foreach** $j \in [1, n]$ **do**
    Calculate observation value:
    $\tau_{ij}^l(k) = \exp - \frac{(\delta^{ij}(k))^2}{\sigma_l^2}$
    **if** $l \neq i \neq j$ **then**
      $\epsilon_{ij}^l(k) = 0$
    **else**
      $\epsilon_{ij}^l(k) = \tau_{ij}^l(k) - a_{ij}^l(k)$
    **end**
    Calculate update term:
    $\Delta a_{ij}^l(k) = \sum_{p \in \mathcal{N}_l} a_{lp}^l(k)(a_{ij}^p(k) - a_{ij}^l(k)) + \epsilon_{ij}^l(k)$
    Calculate new estimates of adjacency matrix elements:
    $a_{ij}^l(k+1) = a_{ij}^l(k) + T_d \cdot \Delta a_{ij}^l(k)$
  **end**
**end**

with

$$\hat{\mathbf{L}} = \begin{bmatrix} \mathbf{l_1^1} \\ \mathbf{l_2^2} \\ \dots \\ \mathbf{l_n^n} \end{bmatrix}.$$

Finally, including (33) in (16) gives

$$\mathbf{a}_{ij}(k+1) = \mathbf{a}_{ij}(k) - \hat{\mathbf{L}}\mathbf{a}_{ij}(k) + \boldsymbol{\epsilon}_{ij}(k). \quad (34)$$

As the last step of the proof we have to show that

$$-\hat{\mathbf{L}}\mathbf{a}_{ij}(k) + \boldsymbol{\epsilon}_{ij}(k) = 0$$

if (27) is true. From (31) it is clear that in case $a_{ij}^1(k) = a_{ij}^2(k) = \ldots = a_{ij}^n(k)$ one has

$$\begin{aligned} \mathbf{l}_1^1 &= \mathbf{l}_1^2 = \ldots = \mathbf{l}_1^n = \mathbf{l}_1, \\ \mathbf{l}_2^1 &= \mathbf{l}_2^2 = \ldots = \mathbf{l}_2^n = \mathbf{l}_2, \\ &\ldots, \\ \mathbf{l}_n^1 &= \mathbf{l}_n^2 = \ldots = \mathbf{l}_n^n = \mathbf{l}_n, \end{aligned}$$

which gives $\hat{\mathbf{L}} = \mathbf{L}$. Due to Laplacian matrix properties, $\mathbf{L}\mathbf{a}_{ij}(k) = 0$ if (27) is true. The same holds for vector $\epsilon_{ij}$, i.e. $\epsilon_{ij} = \mathbf{0}$ since, according to definition (24), all its components are equal to zero. ∎

To summarize, the pseudo-code of the proposed decentralized adjacency matrix estimation method is given in algorithm 2. It can be noticed that the computational complexity of the proposed algorithm is equal to $O(n^2)$. Compared to power iteration methods for distributed estimation of the algebraic connectivity, that have been intensively investigated in the decentralized multi-agent systems [18]–[22], as described in Section I-A, our method provides the same complexity level with the benefit of estimating the entire graph's topology in a decentralized manner.

## VI. Simulation Results

The presented simulation experiments are inspired by our work on animal-robot societies within ASSISIbf project [53], where miniature robots used a communication network to exchange information related to the properties of the society they formed with honeybees [9]. To demonstrate the effectiveness of the proposed consensus-based method for calculation of $\lambda_2$ and the described control law for connectivity maintenance, we conducted extensive simulations for three different group sizes, i.e. we have chosen a group of 5, 9, and 25 robots. For each of these examples, we generated an initial communication graph topology in order to set the initial distribution of robots and their connectivity radius. Based on the initial topology, as we stated earlier, we assume that each agent $l$ can establish a communication channel with all agents that are within $\mathcal{R}^l$, whose value is based on technical properties of the chosen type of communication protocol, corresponding hardware layer, and environment in which the group executes their missions. At the beginning of the simulation, each agent only has the knowledge about the local network topology, which, for the 5-robots example as in Fig. 5, gives

$$\mathbf{A}^1(0) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{A}^2(0) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A}^3(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \mathbf{A}^4(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{A}^5(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

In all examples presented in this section, the sample time was $T_d = 0.2$ [s], $\kappa = 25 \cdot k$ (adding/removing links is initiated once for every 25 communication intervals), the algebraic connectivity threshold was $K_{\lambda 2} = 0.1$ (for example with 25 nodes $K_{\lambda 2} = 0.05$) and the maximum number of links that a robot can have was $d_{max}^* = 5$ (for the example with 5 robots $d_{max}^* = 4$).

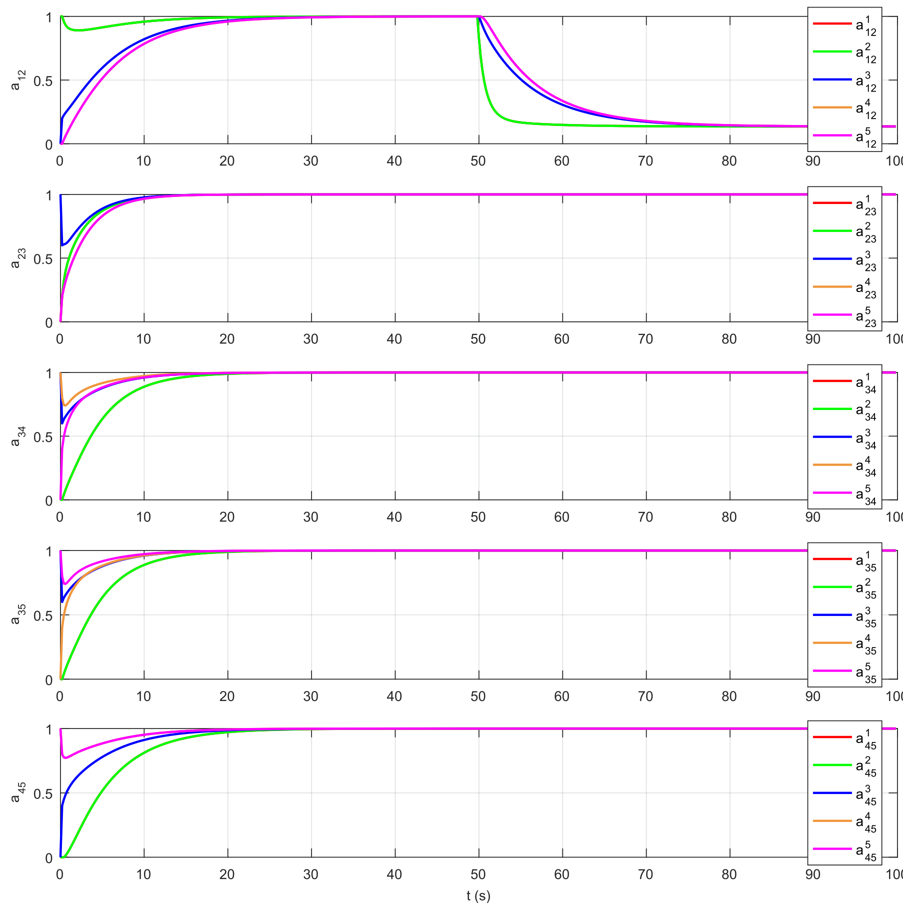


Fig. 6. Time response of estimations of local adjacency matrices elements.

### A. Estimation of Global Adjacency Matrix

In this example, we demonstrate the convergence of the local matrices towards the global adjacency matrix for the 5-robots group with the initial graph topology shown in Fig. 5.

As the simulation proceeded, the elements of local adjacency matrices, calculated from (16), asymptotically converged to the real values so that (3) is satisfied, as Fig. 6 illustrates.

At the time instant $t = 50$ [s] a step change in $N_{rec}^2$ is initiated, so that the number of received data packets significantly differs from the number of sent data packets, thus increasing $\delta^{12}$. This increase has been noticed by agents 1 and 2 through the observation function, which in turn changed $d_{12}^1$ and $d_{12}^2$ causing a new cycle of estimation. In the presented time responses (Fig. 6), it can be noticed that the estimated values for $a_{12}$ converge slightly faster to the new value in robot 3 that is closer to robot 2, than in robots 4 and 5 that are further away. Naturally, the fastest convergence is achieved in robots 1 and 2 that are communicating through $a_{12}$.

Fig. 7 shows time responses of local algebraic connectivity values and their convergence to a common value before and after the step change in the quality of the communication link $a_{12}$. It can be noticed that initial values for $\lambda_2$ are all equal to 0 since agents are aware only of local connections, hence the estimated global graph is not connected. As information from other agents in the group arrives, local estimates become more and more accurate, finally approaching the true value of algebraic connectivity. The decrease in $a_{12}$ quality at $t = 50$ [s] caused $\lambda_2$ to decrease, but this has been closely followed by local estimates.

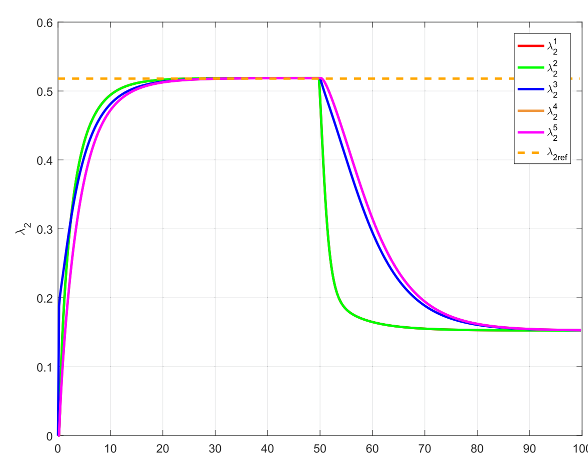


Fig. 7. Time response of local algebraic connectivity values with the algebraic connectivity of initial graph given by dashed line.

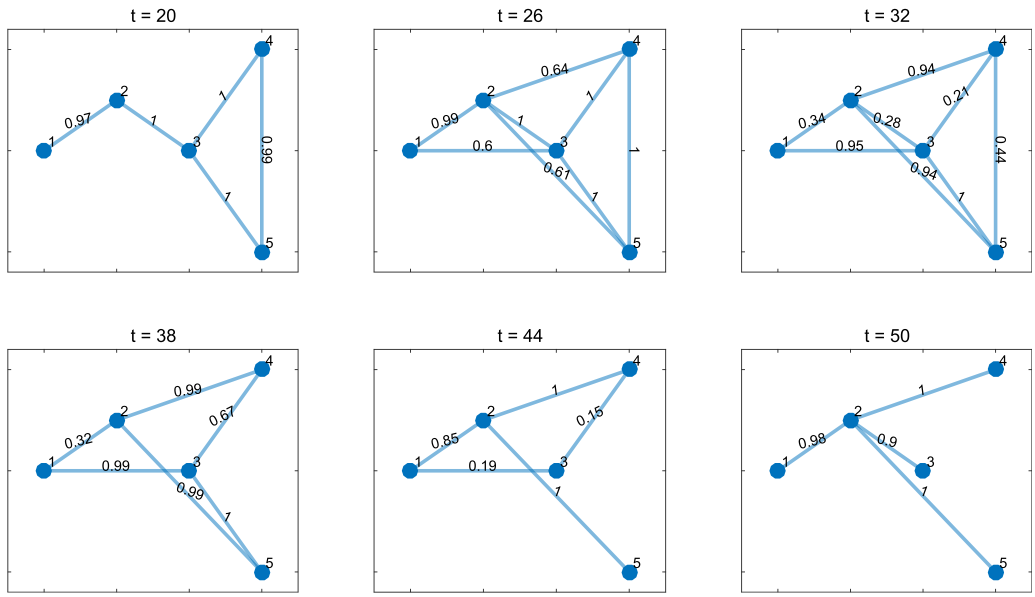


Fig. 8. Snapshots of the network topology of the system with 5 robots in different time instants.

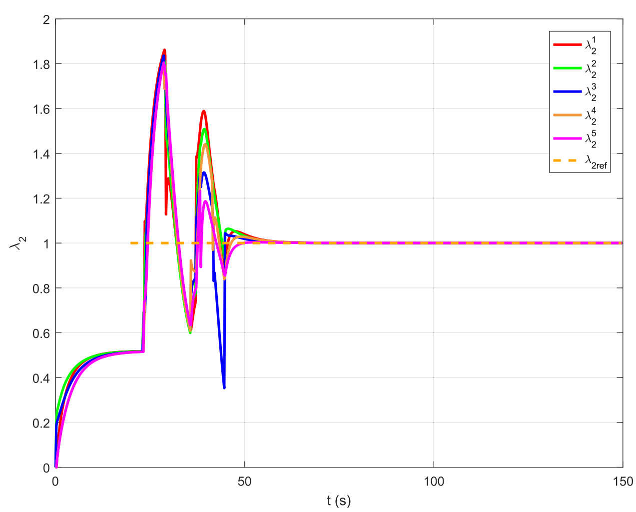


Fig. 9. Time responses of robots' local algebraic connectivity for the system with 5 robots and reference value $\lambda_{2ref} = 1$. The connectivity feedback control loop is active from $t = 20$ s.

### B. Connectivity Maintenance

The effectiveness of the connectivity maintenance method is demonstrated on the same topology as in the previous experiment. As shown in Fig. 9, time responses of local estimates of $\lambda_2^l$ attain values corresponding to the initial graph layout (Fig. 5) after 15 seconds. Then, at $t = 20$ [s] the reference algebraic connectivity value changes to $\lambda_{2ref} = 1$ (dashed line in Fig. 9). The effect of the connectivity maintenance method is visible as the agents start adding/removing communication links (snapshots of the network topology in different time instants are presented in Fig. 8), which causes changes in local estimates of adjacency matrices, and consequently changes in $\lambda_2^l$. Finally, at $t = 50$ [s], agents reach consensus on the topology so that $|e^l_{\lambda_2}(k)| < K_{\lambda 2}$.

TABLE IV
AVERAGE VALUES OF GRAPH'S PARAMETERS FOR A SYSTEM WITH 5 ROBOTS IN 10 EXPERIMENTS

| $\bar{d}$ | $\bar{m}_{add}$ | $\bar{m}_{rem}$ | $\bar{\lambda}_2$ |
|---|---|---|---|
| 2.24 | 15.1 | 17.5 | 1.0 |

Table IV summarizes the results of 10 conducted simulation experiments for the examined multi-robot system, with variations in the time instant when the reference value of the algebraic connectivity was set to $\lambda_{2ref} = 1$. We calculated the average values of the following variables:

- $\bar{d}$ - average node degree in stationary state,
- $\bar{m}_{add}$ - average number of added links,
- $\bar{m}_{rem}$ - average number of removed link,
- $\bar{\lambda}_2$ - average algebraic connectivity in stationary state.

### C. Scalability of the Proposed Method

To demonstrate the scalability of the proposed connectivity maintenance method, we conducted numerous simulation experiments using a group of 9 and 25 robots. For the system with 9 robots, an initial configuration is shown in Fig. 10, where each agent is connected with its closest neighbors. Using a similar approach, an initial topology for the system with 25 robots is constructed.

First, we presented the ability of the proposed connectivity maintenance method to follow the reference value as demonstrated in Fig. 11. The connectivity control starts at $t = 50$ [s], when the algebraic connectivity reference value is set to $\lambda_{2ref} = 0.5$. At $t = 150$ [s], the reference value changes to $\lambda_{2ref} = 1.5$. Due to the decentralized nature of the proposed method, the time required to reach consensus on the topology remained similar to the case with the 5-robot group.

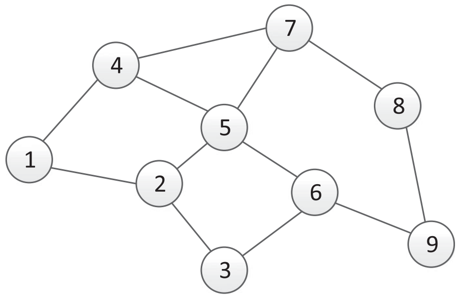


Fig. 10. The initial graph topology for a multi-robot system with 9 robots where $\mathbf{A}^l$ defines robot's $l$ estimate of the adjacency matrix.

In the following experiment, demonstrating the method resilience, we simulate a failure of robot 5, first with the connectivity maintenance controller turned off, and then with the controller turned on. The time responses of estimated algebraic connectivity values for the group of 9 robots with the controller turned off, are presented in Fig. 12. Obviously, the estimated values approach zero, since the robots are not able to establish new links, and the communication graph is not connected anymore. With the controller turned on, once the robot with a poor connectivity performance is detected, it is excluded from the group and the algebraic connectivity estimation process. Time responses of local algebraic connectivity estimates for that scenario are shown in Fig. 13. All robots, except for robot 5 (that becomes a single node graph), reached a consensus.

The final set of simulations relates to a group of 25 robots. Results are shown in Figs. 14 and 15. In the first scenario, the algebraic connectivity reference value changes from $\lambda_{2ref} = 1$ to $\lambda_{2ref} = 0.5$ at $t = 150$ [s], while in the second scenario the algebraic connectivity reference value is $\lambda_{2ref} = 0.2$, connectivity maintenance controller is active and two robots (robot 5 and 13) fail at $t = 150$ [s]. It can be seen that the remaining robots detect the failure and form a new topology in order to maintain the algebraic connectivity. Even though the local estimates are kept close to the desired value, slight oscillations appear in the time responses. This is due to the small communication radius for a group with a relatively large number of nodes, hence agents have only a few options for establishing a new connection. This situation can be resolved for example by increasing $K_{\lambda 2}$ (we already mentioned future work on an adaptation of this parameter), or by allowing agents to extend their communication radius (which would increase energy demand).

## VII. Experimental Results

The proposed communication network connectivity maintenance method has been experimentally validated in a very demanding underwater environment. A group of 5 aMussel robots, developed in subCULTron project [54] with a mission to monitor sea conditions, was deployed as a multi-agent system at the seabed near the city of Biograd na Moru in Croatia, as shown in Fig. 16. The experiments that we have conducted are a continuation of our work presented in [55], where we used median consensus to track the outliers in group measurements under two communication topologies - a complete

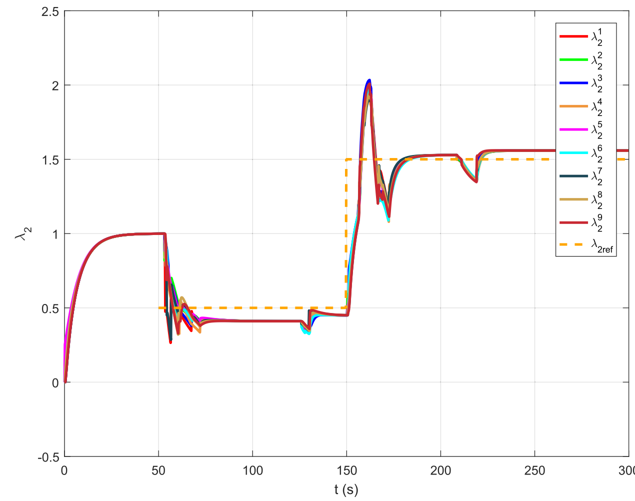


Fig. 11. Time responses of local algebraic connectivity estimates for the system with 9 robots.

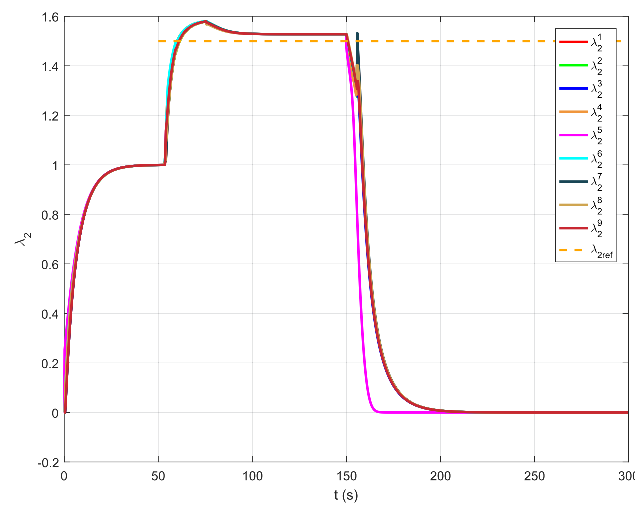


Fig. 12. Time responses of local algebraic connectivity estimates for the system with 9 robots without detection of failed node. The robot 5 fails at $t = 150$ s.

graph, and a chain graph. Since underwater acoustic communication is energy demanding, changing the topology to keep the connectivity at a certain (low) level is of utmost importance, particularly in the case when the mission scenario assumes changes in distances between underwater robots. Furthermore, the underwater environment imposes three key limitations - low bandwidth, sequential communication, and broadcasting - dictating some implementation details of the proposed methods, as we describe in the following subsection.

### A. Implementation Details for the Proposed Method

*Limited bandwidth.* Acoustic modems used on aMussel robots operate in 5 seconds wide intervals, during which they transmit 7 bytes of data. We reserved a total of 50 bits for the values of the transmitted adjacency matrix. The upper triangle part of the 5x5 square matrix consists of 10 values. Each $a_{ij} \in [0, 1]$ was therefore quantized using 5 bits, or 32 discrete values. Since the adjacency matrix is symmetric and has zeros on its diagonal, it can be easily reconstructed upon receiving the message.

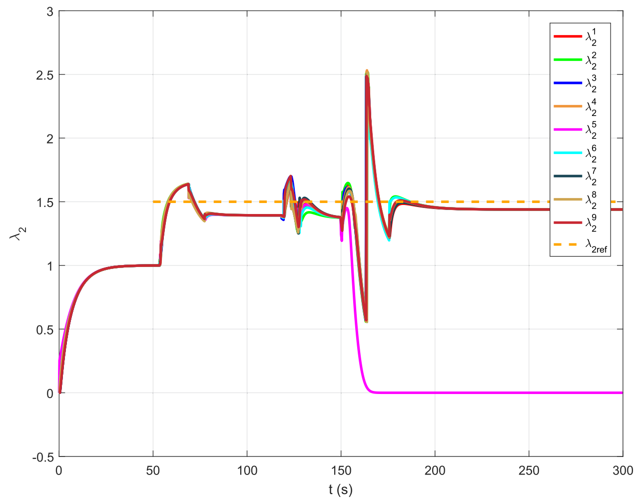


Fig. 13. Time responses of local algebraic connectivity estimates for the system with 9 robots with detection and exclusion of failed robot. The robot 5 fails at $t = 150$ s.

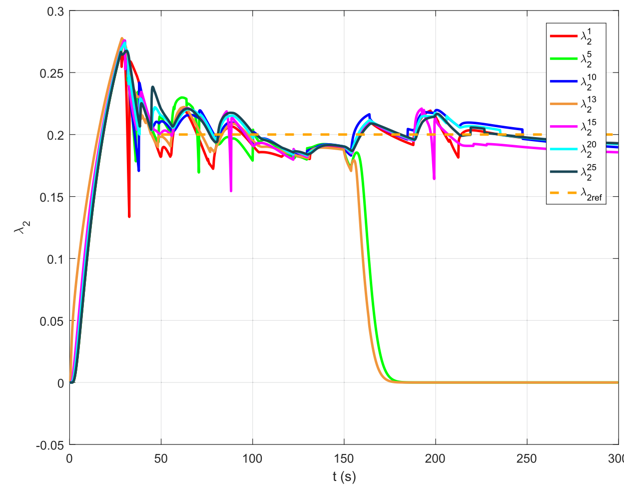


Fig. 15. Time responses of local algebraic connectivity estimates of the system with 25 robots, in case robots 5 and 13 fail at t=150 s.

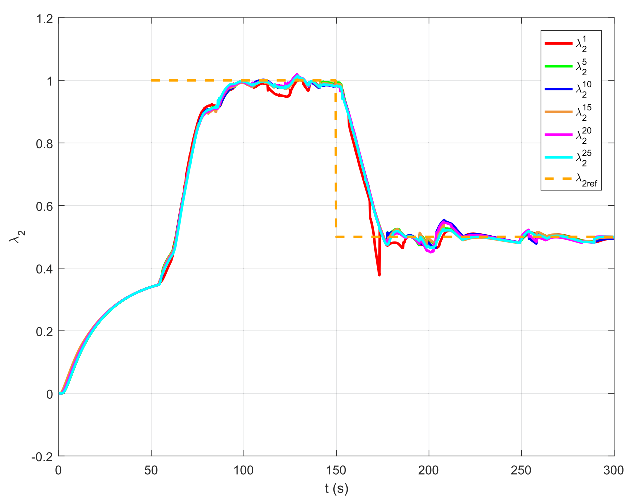


Fig. 14. Time responses of local algebraic connectivity estimates of every fifth robot for the system with 25 robots.

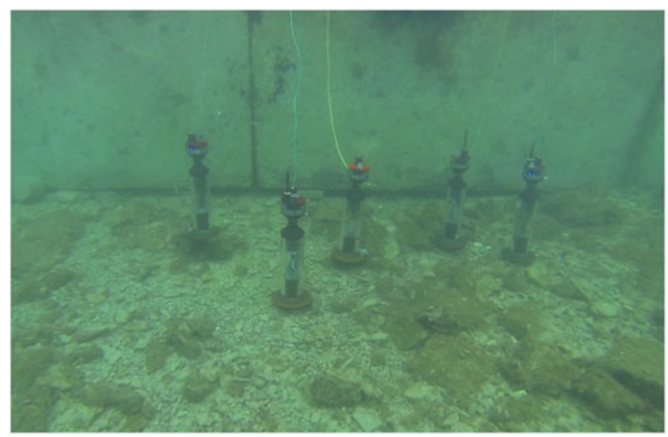

Fig. 16. Setting up aMussel robots prior to the experiments.

*Broadcasting.* Acoustic communication is undirected and spreads in all directions under water. Therefore, all agents in the communication range of the modem have access to the message, regardless of the communication topology. However, in order to save CPU time and energy, each aMussel has implemented a filter, so that incoming messages originated by the agents that are not in its neighborhood set, are ignored.

*Sequential communication.* In underwater communication, only a single acoustic modem is allowed to transmit data at a time. If multiple acoustic signals are sent simultaneously, they can interfere, and data can be corrupted. Because of that, we implemented a widely-used round-robin sequence of communication windows in which each of the agents may transmit data [48], i.e. communication time slots are assigned equally among all agents in a circular order. We consider that a single cycle of all 5 agents sending data (lasting 25 seconds) is equivalent to a single step $k$ of the proposed algorithm. In order to shorten the time to convergence, agents run consensus at a frequency of 5 Hz. That means that each agent is continuously reapplying the consensus update rule using values received in the last step $k$, which does not change the overall behavior of the system, but greatly increases the convergence speed.

After $\kappa = 5$ cycles, a connectivity maintenance cycle is inserted, in which agents use the same round-robin sequence for communication, but instead of the values of the adjacency matrix, they now send the number of messages received from their neighbors since the last connectivity maintenance cycle, and any changes in the topology (adding or removing links). The number of received messages can depend e.g. on the distance between the robots, or other factors, and is used to calculate the $\tau$ (23). The message filters are updated with newly added or removed links. The consensus protocol does not run during this 25-second cycle.

During the connectivity maintenance cycle, a situation may occur, in which two or more agents decide to remove their links with a third agent, possibly disconnecting it entirely from the group. The most effective solution to prevent that situation is to require both participating agents to agree on the removal of the link. Because of the limited bandwidth and sequenced communication, we allow only one agent to modify links in each cycle $\kappa$. This agent is also selected according to a round-robin

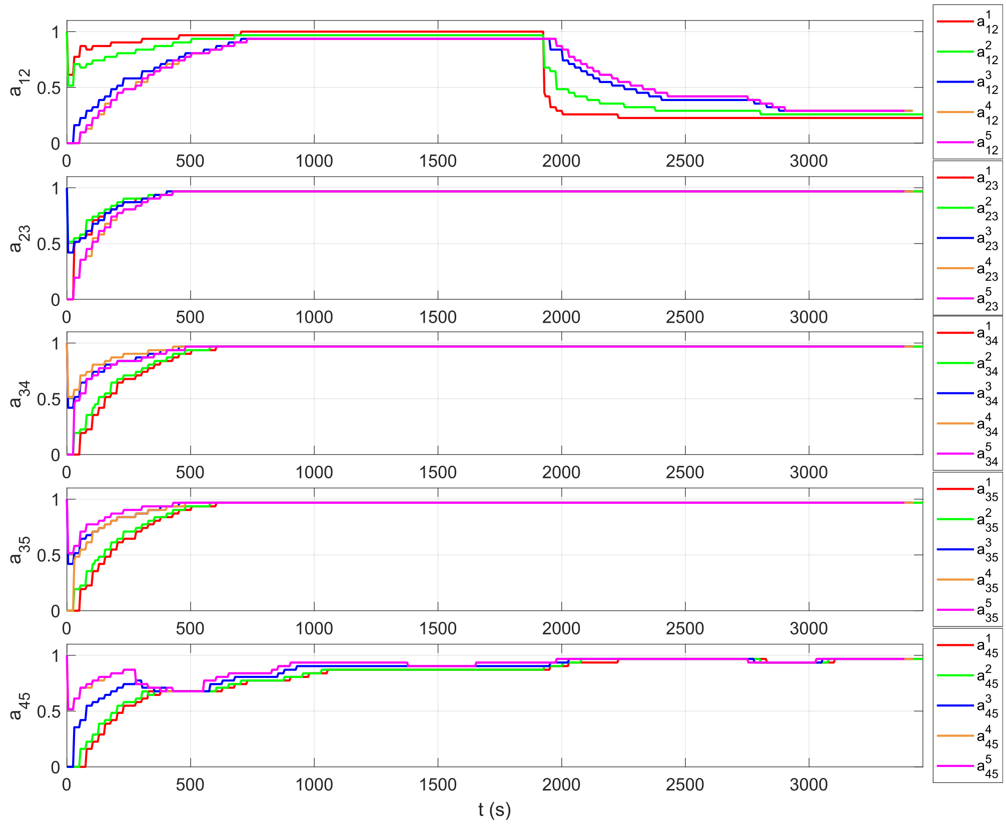


Fig. 17. Time response of estimations of local adjacency matrices elements.

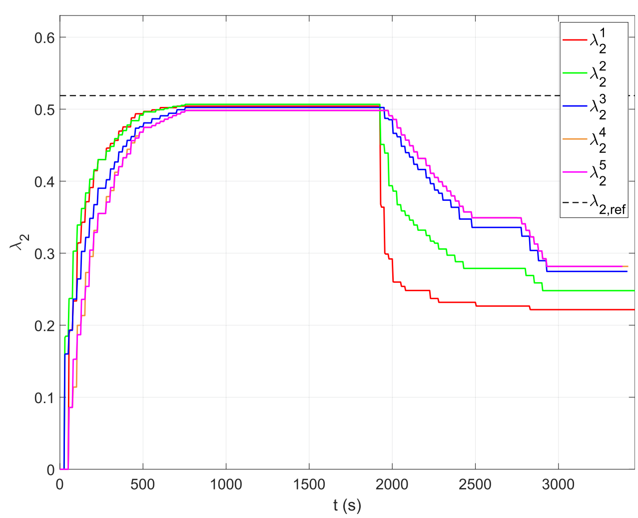


Fig. 18. Time response of local algebraic connectivity values with the algebraic connectivity of initial graph given by dashed line.

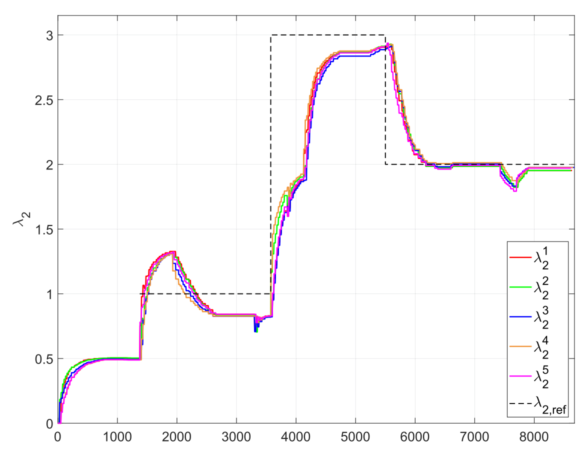


Fig. 19. Time responses of algebraic connectivity estimates for the system with 5 robots and changing $\lambda_{ref}$.

sequence. Since this introduces additional delay into the connectivity control method, we will investigate the possibility of predicting neighbors' actions, and acting accordingly, to prevent accidental disconnections, as a part of our future work.

## B. Estimation of Global Adjacency Matrix

The first set of experiments that we have conducted was related to the estimation of the adjacency matrix. The initial topology was set as in the simulation experiments (Fig. 5). The time response of estimations of local adjacency matrices elements is shown in Fig. 17. It can be seen that all aMussel robots converge to the same values of the communication links, i.e. they are able to correctly estimate the adjacency matrix. It is interesting to notice that estimation of $a_{45}$ takes longer due to the lower quality (noise) of this communication link. In order to check the response of the system to a sudden change in the quality of a particular link, the acoustic modem on aMussle 1 was intentionally occluded at $t = 1920$ [s] after

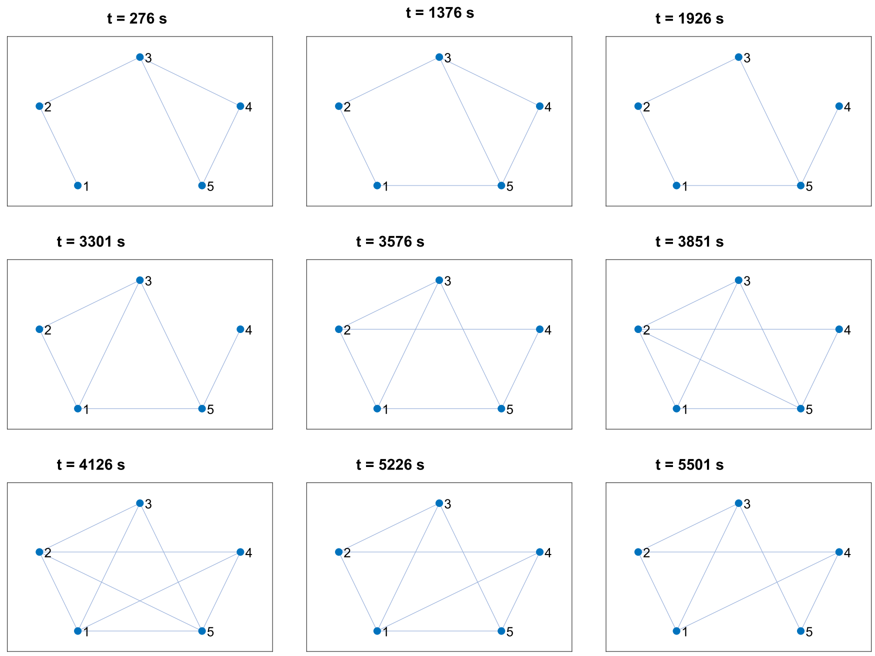


Fig. 20. Snapshots of the network topology of the system with 5 robots in different time instants during reference tracking experiment.

the beginning of the experiment. The graph shows that the response of the system to this change is fast, and all agents converge to a new value of link $a_{12}$.

The time response of the local estimates of the algebraic connectivity value is shown in Fig. 18. Clearly, all agents converged to the common value of $\lambda_2$, depicted by the dashed line. Offsets of estimated values with respect to the algebraic connectivity of the initial graph, as well as the differences in the final values, are due to the quantization of $a_{ij}$ (as we described in the previous section). At $t = 1920$ [s], the algebraic connectivity values start to decrease as a result of significantly deteriorated quality of link $a_{12}$.

### *C. Connectivity Maintenance*

The next set of experiments was dedicated to the investigation of connectivity maintenance. Firstly, we tested the group's ability to follow the reference point, $\lambda_{2ref}$. Secondly, we tested the group's ability to identify a failure of an agent and adjust the communication graph topology so that connectivity remains unchanged.

*Set-point following.* As demonstrated in Fig. 19, $\lambda_2$ set-point dynamics are introduced, so that $\lambda_{2ref}(t = 1375 \text{ s}) = 1$, $\lambda_{2ref}(t = 3576 \text{ s}) = 3$, and $\lambda_{2ref}(t = 5500 \text{ s}) = 2$. The tracking error threshold is set to $K_{\lambda_2} = 0.2$, which is the upper bound according to the recommendations from Section IV. The upper bound was chosen because the values of $a_{ij}$ are quantized due to the limited bandwidth, hence requiring a wider threshold. The connectivity feedback control loop is active from $t = 1375$ [s]. It can be seen that all agents reached consensus on the initial topology, $\lambda_{2ref} = 1$, and as soon as the new reference value was set, $\lambda_{2ref} = 3$, the control algorithm initiated the addition of new communication links. Decrease of $\lambda_{2ref}$ at $t = 5500$ [s] is closely followed by agents' actions as well. Snapshots of the network topology of the system with 5 robots at different time instants during the set-point following experiment are shown in Fig. 20.

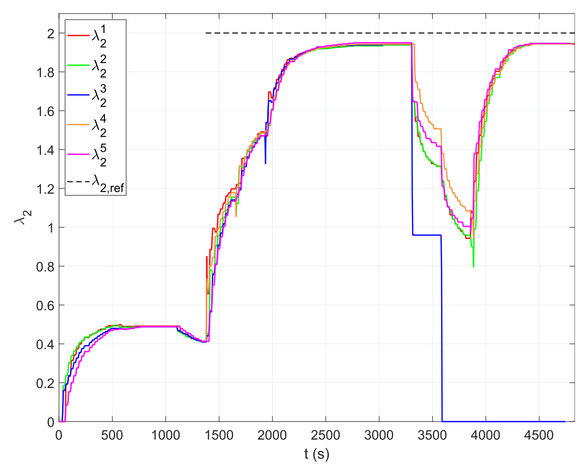


Fig. 21. Time responses of algebraic connectivity estimates for the system with 5 robots with detection and exclusion of failed robot.

*Agent failure.* In this set of experiments, we validated the ability of the proposed method to detect a failed agent and to reconfigure the communication topology so that the healthy agents remain connected. As in the previous experiments, the connectivity feedback control loop is active from $t = 1375$ [s] to allow agents to reach consensus on the initial topology, with the reference value of the algebraic connectivity set to $\lambda_{2ref} = 2$. At $t = 3300$ [s] robot 3 is pulled out from the sea to simulate a total failure of an agent. As it can be seen from Fig. 21, the neighbors of robot 3 detected that there have not been any messages from it, and the quality of their links significantly decreased. This information propagated through the group by the implemented consensus protocol, resulting in all the other agents excluding agent 3 from the communication graph, and rearranging the links to maintain the desired connectivity. It should be noted that robot 3 kept running connectivity maintenance algorithm outside the sea as well, and soon detected that it has been disconnected from the group, as can be seen with $\lambda_2^3 = 0$ (blue line in Fig. 21).

## VIII. Conclusion

The full potential of cooperating agents with a common objective of a group can be fully exploited only if appropriate decentralized control algorithms are used. The challenge is to guarantee the stable performance of a system composed of simple and inexpensive agents with limited computational and communication capabilities. In this paper, we have extended a well-known consensus algorithm, used in many multi-agent applications, in order to determine the time-varying network topology and graph's algebraic connectivity. Furthermore, we designed a connectivity controller with the local estimates of the algebraic connectivity used as feedback. Finally, we introduced a probability algorithm based on Albert-Barabási preferential model for creation and deletion of communication links, taking into account the agent's local neighborhood.

In this paper, we have selected a collection of simulation scenarios that best illustrate the effectiveness of the described decentralized algorithms for estimation and control of the algebraic connectivity in multi-agent systems. The conducted experiments show that the networked multi-robot system can successfully control the second smallest eigenvalue of the graph Laplacian at a common reference value in a decentralized manner. Furthermore, the resilience of the proposed method has been demonstrated through a failure of a robot, where we showed that the system was able not only to pinpoint the failed agent(s), but also to reconfigure the underlying communication graph, and return the algebraic connectivity to the value prior to the failure.

Finally, the experimental validation of the proposed method has been done in the demanding underwater environment, where we implemented the proposed consensus-based distributed connectivity control on a system of underwater robots that used acoustic modems for communication. The presented results demonstrate that the group of underwater agents can successfully maintain and track the reference connectivity. Furthermore, the group is able to adjust to the agent failure by re-configuring the underlying communication graph, so that pre-failure algebraic connectivity is achieved.

The future work will consider adaptation of the hysteresis controller parameter $K_{\lambda 2}$, so that the multi-agent system accommodates to the group size and to the variable number of communication links. We will also investigate how other models for the creation/deletion of the links fit with the consensus-based algorithm and hysteresis controller.

## References

[1] N. Correll and A. Martinoli, "Multirobot inspection of industrial machinery," *IEEE Robot. Automat. Mag.*, vol. 16, no. 1, pp. 99–102, Mar. 2009.

[2] T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Trans. Robot. Automat.*, vol. 18, no. 1, pp. 94–103, Feb. 2002.

[3] M. Toussaint and M. Lopes, "Multi-bound tree search for logic-geometric programming in cooperative manipulation domains," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 4044–4051.

[4] A. Oliveira *et al.*, "Design and development of a multi rotor UAV for oil spill mitigation," in *Proc. IEEE Ocean. Eng. Soc. Marseille*, 2019, pp. 1–7.

[5] F. Bonnet and X. Defago, "Exploration and surveillance in multi-robots networks," in *Proc. 2nd Int. Conf. Netw. Comput.*, 2011, pp. 342–344.

[6] D. Scaramuzza *et al.*, "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments," *IEEE Robot. Automat. Mag.*, vol. 21, no. 3, pp. 26–40, Sep. 2014.

[7] A. Colot, G. Caprari, and R. Siegwart, "InsBot: Design of an autonomous mini mobile robot able to interact with cockroaches," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2004, pp. 2418–2423.

[8] K. Griparić, T. Haus, D. Miklić, M. Polić, and S. Bogdan, "A robotic system for researching social integration in honeybees," *PLoS One*, vol. 12, no. 8, 2017, Art. no. e0181977.

[9] F. Bonnet *et al.*, "Robots mediating interactions between animals for interspecies collective behaviors," *Sci. Robot.*, vol. 4, no. 28, 2019, Art. no. eaau7897.

[10] F. Ducatelle, G. A. Di Caro, C. Pinciroli, F. Mondada, and L. Gambardella, "Communication assisted navigation in robotic swarms: Self-organization and cooperation," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2011, pp. 4981–4988.

[11] V. Gazi, "Swarm aggregations using artificial potentials and sliding-mode control," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1208–1214, Dec. 2005.

[12] S. Kernbach *et al.*, "Adaptive collective decision-making in limited robot swarms without communication," *Int. J. Robot. Res.*, vol. 1, no. 32, pp. 35–55, 2013.

[13] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[14] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[15] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control: Collective group behavior through local interaction," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, May 2007.

[16] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[17] F. Bullo, J. Cort, and S. Mart, *Distributed Control of Robotic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[18] L. Sabattini, C. Secchi, and N. Chopra, "Decentralized estimation and control for preserving the strong connectivity of directed graphs," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2273–2286, Oct. 2015. [Online]. Available: https://doi.org/10.1109/tcyb.2014.2369572

[19] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proc. IEEE*, vol. 99, no. 9, pp. 1525–1540, Sep. 2011.

[20] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 3628–3633.

[21] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1326–1332, Oct. 2013.

[22] Y.Kantaros and M. M. Zavlanos, "Distributed intermittent connectivity control of mobile robot networks," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3109–3121, Jul. 2017. [Online]. Available: https://doi.org/10.1109/tac.2016.2626400

[23] F. Knorn, R. Stanojevic, M. Corless, and R. Shorten, "A framework for decentralised feedback connectivity control with application to sensor networks," *Int. J. Control*, vol. 82, no. 11, pp. 2095–2114, Sep. 2009. [Online]. Available: https://doi.org/10.1080/00207170902912056

[24] Y. Mostofi, A. Gonzalez-Ruiz, A. Gaffarkhah, and D. Li, "Characterization and modeling of wireless channels for networked robotic and control systems - a comprehensive overview," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 4849–4854.

[25] D. D. Woods, “Four concepts for resilience and the implications for the future of resilience engineering,” *Rel. Eng. Syst. Saf.*, vol. 141, pp. 5–9, 2015.
[26] R. Wehbe and R. K. Williams, “Probabilistically secure and resilient multi-robot coordination in the face of unreliable interactions,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6640–6646.
[27] D. G. Angeler and C. R. Allen, “Quantifying resilience,” *J. Appl. Ecol.*, no. 53, pp. 617–624, 2016.
[28] M. M. Gulzar, S. T. H. Rizvi, M. Y. Javed, U. Munir, and H. Asif, “Multi-agent cooperative control consensus: A comparative review,” *Electronics*, Multidisciplinary Digit. Publishing Inst., vol. 7, no. 2, p. 22, 2018.
[29] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.
[30] S. M. Dibaji and H. Ishii, “Resilient multi-agent consensus with asynchrony and delayed information,” *Int. Federation Autom. Control PapersOnLine*, vol. 48, no. 22, pp. 28–33, 2015. [Online]. Available: https://doi.org/10.1016/j.ifacol.2015.10.302
[31] K. Saulnier, D. Saldaña, A. Prorok, G. J. Pappas, and V. Kumar, “Resilient flocking for mobile robot teams,” *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 1039–1046, Apr. 2017.
[32] T. Wheeler, E. Bharathi, and S. Gil, “Switching topology for resilient consensus using Wi-Fi signals,” in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 2018–2024.
[33] S. Talukdar, D. Deka, S. Attree, D. Materassi, and M. Salapaka, “Learning the exact topology of undirected consensus networks,” in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 5784–5789.
[34] A. Olshevsky and J. N. Tsitsiklis, “Convergence speed in distributed consensus and averaging,” *Soc. Indian Automobile Manufacturers J. Control Optim.*, vol. 48, no. 1, pp. 35–55, 2009.
[35] Z. Zhang, X. Wang, and Q. Xin, “A new performance metric for construction of robust and efficient wireless backbone network,” *IEEE Trans. Comput.*, vol. 61, no. 10, pp. 1495–1506, Oct. 2012.
[36] S. Kar and J. M. F. Moura, “Sensor networks with random links: Topology design for distributed consensus,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, Jul. 2008.
[37] M. Rafiee and A. M. Bayen, “Optimal network topology design in multiagent systems for efficient average consensus,” in *Proc. 49th IEEE Conf. Decis. Control*, 2010, pp. 3877–3883.
[38] P. D. Lorenzo and S. Barbarossa, “Distributed estimation and control of algebraic connectivity over random graphs,” *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 1–10, Nov. 2014.
[39] W. Luo, N. Chakraborty, and K. Sycara, “Minimally disruptive connectivity enhancement for resilient multi-robot teams,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 11809–11816.
[40] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999. [Online]. Available: https://science.sciencemag.org/content/286/5439/509
[41] C. Godsil and G. Royle, *Algbraic Graph Theory*. Berlin, Germany: Springer, 2001.
[42] A. Bertrand and M. Moonen, “Seeing the bigger picture: How nodes can learn their place within a complex ad hoc network topology,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 71–82, May 2013.
[43] P. Erdos and A. Renyi, “On the evolution of random graphs,” *Publ. Math. Inst. Hungary. Acad. Sci.*, vol. 5, pp. 17–61, 1960.
[44] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.
[45] A. Jamakovic and S. Uhlig, “On the relationship between the algebraic connectivity and graph’s robustness to node and link failures,” in *Proc. Next Gener. Internet Netw.*, 2007, pp. 96–102.
[46] T. Haus, I. Palunko, D. Tolić, S. Bogdan, F. L. Lewis, and D. G. Mikulski, “Trust-based self-organising network control,” *Institution Eng. Technol. Control Theory Appl.*, vol. 8, no. 18, pp. 2126–2135, 2014.
[47] M. Krizmancic, B. Arbanas, T. Petrovic, F. Petric, and S. Bogdan, “Cooperative aerial-ground multi-robot system for automated construction tasks,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 798–805, Apr. 2020.
[48] I. Lončar *et al.*, “A heterogeneous robotic swarm for long-term monitoring of marine environments,” *Appl. Sci.*, vol. 9, no. 7, 2019, Art. no. 1388.
[49] D. Tolić, I. Palunko, A. Ivanović, M. Car, and S. Bogdan, “Decentralized cooperative control in degraded communication environments,” in *Control Complex Systems*. New York, NY, USA: Elsevier, pp. 373–395, 2016.
[50] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan, “Decentralized planning and control for uav-ugv cooperative teams,” *Auton. Robots*, vol. 42, no. 8, pp. 1601–1618, 2018.
[51] A. Olshevsky and J. Tsitsiklis, “Convergence speed in distributed consensus and averaging,” *Soc. Ind. Appl. Math. Review*, vol. 53, no. 4, pp. 747–772, 2011.
[52] S. Gavrilets, J. Auerbach, and M. van Vugt, “Convergence to consensus in heterogeneous groups and the emergence of informal leadership,” *Sci. Rep.*, vol. 6, 2016, Art. no. 29704.
[53] ASSISIbf, “EU FET project ASSISIbf - Bees and fish communicate via robots,” Mar. 20, 2019. [Online]. Available: https://www.youtube.com/watch?v=b8EQuUntENs
[54] R. Thenius *et al.*, “Subcultron - Cultural Development as a Tool in Underwater Robotics,” in *Artificial Life and Intelligent Agents*, P. R. Lewis, C. J. Headleand, S. Battle, and P. D. Ritsos, Eds. Berlin, Germany: Springer, 2018, pp. 27–41.
[55] G. Vasiljevic, T. Petrovic, B. Arbanas, and S. Bogdan, “Dynamic median consensus for marine multi-robot systems using acoustic communication,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5299–5306, Oct. 2020.

**Karlo Griparić** (Member, IEEE) received the bachelor’s and master’s degrees in electrical engineering from the Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Zagreb, Croatia, in 2009 and 2011, respectively, and the Ph.D. degree in control of decentralized multiagent systems from the same faculty, in 2018. From 2011 to 2018, he was a Research Assistant with the Laboratory for Robotics and Intelligent Control Systems, FER. He is currently a Postdoctoral Researcher with the Faculty of Engineering, University Juraj Dobrila of Pula, Pula, Croatia. His research interests include decentralized control, multiagent systems, robotics, and industrial automation.

**Marsela Polić** (Student Member, IEEE) received the M.S. degree in electrical engineering and information technology in 2016 from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, where she is currently working toward the Ph.D. degree as a Member of the Laboratory for Robotics and Intelligent Control Systems. Her research interests include soft and collaborative robotics and artificial intelligence.

**Marko Krizmancic** (Student Member, IEEE) received the M.S. degree in electrical engineering and information technology in 2019 from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, where he is currently working toward the Ph.D. degree as a Member of the Laboratory for Robotics and Intelligent Control Systems. His research interests include multi-agent systems, sensor networks, and high-level planning and scheduling.

**Stjepan Bogdan** (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Zagreb, Zagreb, Croatia, in 1999. He is currently a Full Professor with the Laboratory for Robotics and Intelligent Control Systems, Faculty of Electrical Engineering and Computing, University of Zagreb. He is a coauthor of four books and has authored or coauthored more than 200 conference and journal papers. His research interests include autonomous systems, aerial robotics, multiagent systems, intelligent control, and discrete event systems. He was an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and was a Program Committee and Organizing Committee Member of main control and robotics conferences.