# Improving robustness in multi-robot networks ⋆

**Cinara Ghedini** * **Cristian Secchi** ** **Carlos H. C. Ribeiro** *
**Lorenzo Sabattini** **

* *Computer Science Division, Aeronautics Institute of Technology, São José dos Campos, SP, Brazil (e-mail: cinara,carlos@ita.br).*
** *Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy (e-mail:lorenzo.sabattini,cristian.secchi@unimore.it).*

**Abstract:** This paper addresses the topological robustness of robot networks under failures; a subject often neglected in the literature. Robots are likely to fail due to several causes, which may lead to a poorly connected or a fragmented network. Our purpose is to discuss how to design resilient robot networks. For that, we first demonstrate the problem analyzing the results from a protocol to simulate failures of both central and random (w.r.t. topology) robots. Then, we propose mechanisms for detecting the probability of a robot being in a fragile local configuration and for improving its local robustness. The procedures rely solely on local information: each robot estimates its probability of being in a harmful configuration based on the positions of its neighbors. Such probability is given by the number of paths connecting a robot to its 2-hop neighbors by the number of paths existing in the subgraph encompassing its 1-hop and 2-hop neighborhood. For reversing an adverse configuration, robots change their position to an average position towards their 2-hop neighbors with fewer alternative paths. The results showed that the proposed mechanisms were efficient for detecting fragile topological configurations and for improving the overall network robustness.

*Keywords:* Networked robots; Multi cooperative robot control; Adaptive robot control.

## 1. INTRODUCTION

Network services are nowadays becoming pervasive, and network infrastructure is widely available: this makes it possible to access Internet, cloud technologies, and effective communication services in many circumstances. Nevertheless, there are wide areas of the world where network infrastructure is not available, either because it is not possible from an economic and a technological point of view, or because it was damaged during a disaster event (Tomic et al., 2012).

Mobile robots, if equipped with appropriate communications devices, can be exploited to create an infrastructure network to provide communication services in such environments. For instance, the interconnected mobile robots can provide rescuers, devices in an exploration task, or service clients (e.g. mobile phones, laptops, tablets, etc.) with communication services. Figure 1 illustrates a scenario that fits into several of such contexts of applications. In this example, *mobile robots* are controllable entities, that can communicate among each other, and whose position can be changed in order to provide a sufficiently good network service to a group of mobile clients. With the term *mobile client* we refer to generic uncontrolled mobile entities, that access the network service provided by the mobile robots.

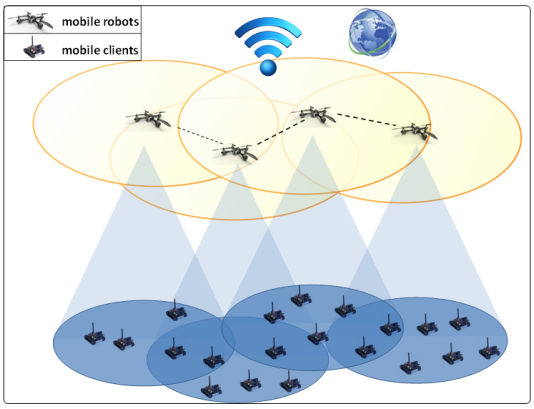


Fig. 1. Application scenario

On these lines, some pioneering projects [1] have been recently started for using autonomous flying systems (in particular, balloons) to bring internet services to areas where they are not available. These areas are typically sparsely populated and/or inhospitable, such as the Amazon region or Central Africa.

Providing network services for unstructured environments by means of interconnected mobile robotic systems involves several issues, from deployment and communication efficiency to topology control. In particular, it is

⋆ The authors would like to thank FAPESP for the financial support to carry out this research (proc. no. 2012/25058-9 and 2014/13800-8)

[1] See for instance: http://www.google.com/loon/ (Loon project), http://altave.com.br (Conectar project)

Preprint submitted to 11th IFAC Symposium on Robot Control.
Received March 13, 2015.

necessary to guarantee that the interconnection topology defines a connected graph, thus ensuring the possibility of exchanging data among all the nodes in the network. In fact, connectivity maintenance is a very well studied topic in the field of decentralized multi-robot systems, and several strategies can be found in the literature for solving this problem in different manners. The main approaches provide solutions for ensuring that, if the communication graph is initially connected, then it will remain connected, that is, if a link among two robots is active at time $t = 0$, than it will remain active as the system evolves, for time $t > 0$. Examples can be found in (Ji and Egerstedt, 2007; Notarstefano et al., 2006; Cao and Ren, 2010; Hsieh et al., 2008; Ajorlou et al., 2010; Dimarogonas and Johansson, 2010; Morbidi et al., 2010). More recently, a few strategies reported in the literature solve the connectivity maintenance problem from a global point of view: single links are allowed to be added or removed, as long as the overall communication graph remains connected. Examples can be found in (Sabattini et al., 2013a,b; Secchi et al., 2013; Sabattini et al., 2014).

To the best of the authors' knowledge, the literature on connectivity maintenance does not generally consider robot failures, and hence does not provide robust solutions in this respect. Robots are prone to failures due to hardware or communication issues, and — as it is well known from the literature on Complex Networks — successive failures, particularly of robots playing a central role in the network topology, may lead to an inoperative or reduced service, see for instance (Albert et al., 2000; Dall'Asta et al., 2006; Ghedini and Ribeiro, 2011; He et al., 2013; Manzano et al., 2013).

On these lines, in this paper we address the robustness problem in multi-robot systems. In particular, in our context a robust network implies that, despite robot failures, most of its elements are still connected, being able to maintain a certain level of service. Thus, the first concern is how to detect and mitigate harmful topological configurations, w.r.t network connectivity. The infrastructure is required to be adaptive, which means that the position of the mobile robots has to dynamically change to avoid such harmful network configurations. In addition, as in practice it is not possible to assume that the mobile robots have a global knowledge of the network, they need to control their position based only on locally available information.

The approach presented here considers that robots use their 1 and 2-hop neighbor positions to estimate the probability of being vulnerable, which means to be in a harmful neighborhood. The results showed good prospects for detecting such states, and based on this local estimation procedure, a control strategy is then proposed, leading to an adaptive network topology for enhancing robustness.

The model comprises mechanisms for environment and network simulations, for network configuration evaluation, and also a protocol to simulate perturbations (*i.e.*, failures) in the network, presented in Section 2. The problem statement is discussed in Section 2.3. A detailed description of the proposed mechanisms is presented in Section 3, and the results obtained are presented and discussed in Section 4.

## 2. PRELIMINARIES

### *2.1 Model of the system*

Consider a multi-robot system composed of $N$ robots, and let $x_i \in \mathbb{R}^m$ be the state of the $i$-th robot. We will hereafter assume that each robot's state is the position of the robot itself, and that the velocity of the robots can be controlled. Hence, we model each robot according to the following discrete time kinematics:

$$x_i(t+T) = x_i(t) + T\,\mu_i(t) \tag{1}$$

where $T > 0$ is the sampling time, and $\mu_i(t) \in \mathbb{R}^m$ is the control input computed at time $t$. Assume also that robots are able to communicate with other robots within the same communication radius $R$. The communication topology can then be represented by means of an undirected graph $\mathcal{G}$ where each robot is a node of the graph, and each communication link between two robots is an edge of the graph.

Let $\mathcal{L} \in \mathbb{R}^{N\times N}$ be the Laplacian matrix of graph $\mathcal{G}$. As is well known from algebraic graph theory, this matrix defines relevant properties of $\mathcal{G}$ (Godsil and Royle, 2001), as detailed in what follows.

### *2.2 Network properties*

We will hereafter define some quantities that can be exploited for evaluating the network topology robustness.

*Definition 1.* (Algebraic connectivity, (Fiedler, 1973)). The algebraic connectivity $\lambda$ of a graph is defined as the second smallest eigenvalue of the Laplacian matrix $\mathcal{L}$. $\diamond$

For a connected graph, the algebraic connectivity is greater than zero ($\lambda > 0$), and it defines a lower bound for node connectivity and link connectivity (Bigdeli et al., 2009).

A *connected component* of a graph is a set of nodes such that a path exists between any pair of nodes in this set. In most real-world complex networks, it has been observed that there is a large connected component together with a number of small components containing no more than a few percent of the nodes (Chen and Chen, 2010). For very large networks, this component is generally referred to as *giant component*. With a slight abuse of notation, we will hereafter use the following definition even for small and medium size networks:

*Definition 2.* (Giant Component, (Newman, 2003)). The Giant Component of a graph is defined as its largest connected component. $\diamond$

The connectivity of a network $\mathcal{G}$ can be estimated by the relative size $S(\mathcal{G})$ of the giant component, given by the fraction of nodes in the network taking part in the largest connected component:

$$S(\mathcal{G}) = \frac{n_{Giant}}{N}, \tag{2}$$

where $n_{Giant}$ is the number of nodes in the giant component and $N$ is the number of nodes in the network. In fact, for a connected graph, all the nodes belong to the same component, implying $n_{Giant} = N$, and then $S(\mathcal{G}) = 1$.

Even though algebraic connectivity is most commonly used in the multi-robot systems literature for assessing

Preprint submitted to 11th IFAC Symposium on Robot Control.
Received March 13, 2015.

the connectedness of a graph, using the giant component exhibits a few remarkable advantages. In particular, as soon as the graph becomes disconnected, the algebraic connectivity goes to zero: this happens even if one single node loses connectivity with the rest of the nodes. On the other hand, the size of the giant component gives a better insight regarding network fragmentation as a process: if a small number of nodes lose connectivity with the other ones, then the size of the giant component decreases slightly. Conversely, we can observe a huge decrease in the size of the giant component when a large fraction of the nodes lose connectivity with the remaining ones.

Generally speaking, different nodes have different roles in maintaining connectivity of the overall network. In particular, the concept of *centrality* is generally exploited for identifying the most important nodes within a graph (Koschützki et al., 2005). Several indicators can be found in the literature for defining centrality. In particular, referring to connectivity maintenance, we will exploit the concept of *Betweenness Centrality* ($BC$) (Wasserman et al., 1994), which establishes higher scores for nodes that are contained in most of the shortest paths between every pair of nodes in the network. In fact, nodes with this feature are likely to be crucial for maintaining the network functionality.

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

$$BC(i) = \sum_{j<l} \frac{g_{jl}(i)}{g_{jl}} \tag{3}$$

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

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

*Definition 3.* (Robustness level). Consider a graph $\mathcal{G}$ with $N$ nodes. Let $[v_1, \ldots, v_N]$ be the list of nodes ordered by descending value of $BC$. Let $\varphi < N$ be the minimum index $i \in [1, \ldots, N]$ such that, removing nodes $[v_1, \ldots, v_i]$ leads to disconnecting the graph, that is the graph including only nodes $[v_{\varphi+1}, \ldots, v_N]$ is disconnected.

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

$$\Theta(\mathcal{G}) = \frac{\varphi}{N}, \tag{4}$$

$\diamond$

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

As a final remark, notice that $\Theta(\mathcal{G})$ is only an estimate of how far the network is from getting disconnected w.r.t. fraction of nodes removed. In fact, it might be the case that different orderings of nodes with the same $BC$ produce different values of $\Theta(\mathcal{G})$.

### *2.3 Problem statement*

This section emphasizes the importance of mechanisms for detecting and mitigating harmful topological configurations. For that, from a centralized point of view, we introduce perturbation in randomly generated networks (parametrization details are described in Section 4), and assess their connectivity sensitivity. In particular, two perspectives are considered to produce such perturbations in the network:

(1) Any robot can fail, and this can be simulated removing nodes from the network at random, with a uniform probability distribution.
(2) Robots with higher centrality fail.

The effect of removing the most central nodes can be quantitatively evaluated, for a given network, using the following procedure:

(1) **network evaluation**: topological properties are computed, namely the algebraic connectivity (1) and the size of the giant component (2),
(2) **nodes ranking**: the network nodes are ranked based on their $BC$ values,
(3) **network perturbation**: the highest centrality node is removed from the network based on its ranking position.

As an example, consider a 20 robots network (Figure 2). Looking at the illustration on the left, is it possible to identify some key elements for maintaining the network connectivity. Focusing on the first two positions regarding $BC$ ranking (red circles), nodes 7 and 19, respectively, if any of them fail the resulting network will be fragmented.

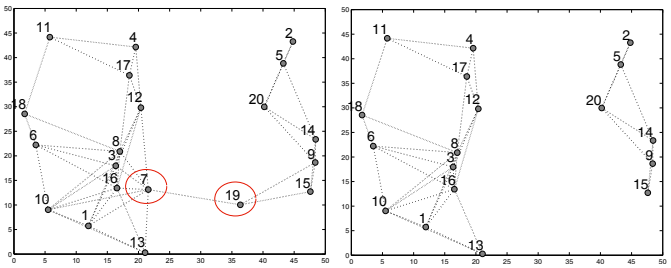

Fig. 2. Initial network configuration (left) and network configuration after the two first highest BC nodes removal (right).

Similar characteristics can be found in larger networks as well. Consider the example depicted in Figure 3, which represents a 100 nodes network. Once again, this network exhibits some poor connectivity spots, as highlighted. Failures of nodes 41, 74 and 11 disconnect the network, which prevents the multi-robot system from achieving cooperative objectives. In this example, these nodes are

Preprint submitted to 11th IFAC Symposium on Robot Control.
Received March 13, 2015.

also the first ones in the $BC$ ranking, demonstrating the centrality measure performance for assessing the impact of robot failures.

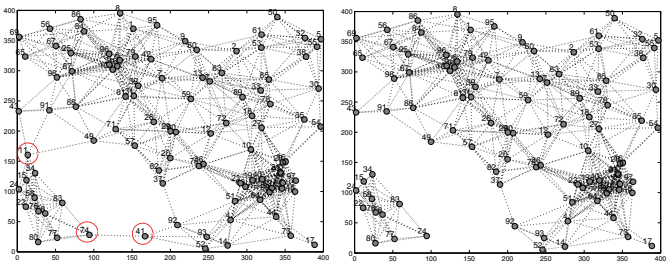

Fig. 3. Initial network configuration (left) and network configuration after the three first highest BC nodes removal

Figure 4 shows the algebraic connectivity (on the left) and the giant component (on the right) evolution as a result of the network perturbations (node failures). The horizontal axis represents the fraction of nodes removed from networks, and the vertical axis the property value. Green lines represent the $N = 20$ case, while blue lines represent the $N = 100$ case. The same experiments were repeated removing the same number of nodes, but randomly chosen (dotted lines). The results demonstrate that the connectivity of both networks was considerably affected by successive losses of the highest $BC$ nodes.

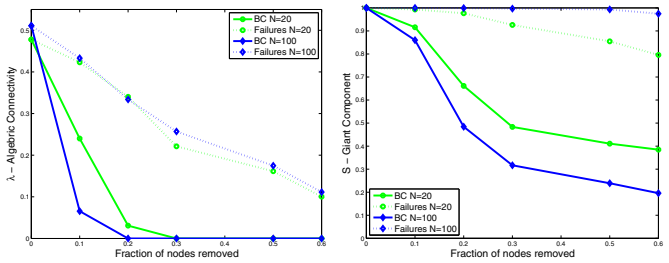


Fig. 4. The algebraic connectivity (on the left) and the size of the giant component (on the right).

To summarize, connectivity is a crucial requirement in decentralized multi-robot systems: in order to achieve a common objective, robots may need to exchange information. On the other hand, robots are likely to fail. The analytical results and examples discussed in this Section demonstrate the importance of a robust connectivity maintenance. Given this evidence, in the next Section we aim at solving the following **problem**: *Given a multi-robot system, design a local estimation procedure that allows each robot to assess its probability of being in a vulnerable configuration, based on locally available information, and subsequently exploit this estimate for controlling the motion of the robots in such a way that the overall robustness of the network is increased.*

## 3. ADAPTIVE MECHANISM

### *3.1 Local evaluation of robustness*

In this section we introduce a local heuristic for estimating the probability of a node being in a vulnerable configuration. It is assumed that nodes can acquire information from their 1-hop and 2-hops neighbors. Thus, define $d(v, u)$ as the shortest path between nodes $v$ and $u$. For the sake of simplicity, we consider an unweighted graph: namely, each edge has weight equal to one. Under this assumption, the shortest path between nodes $v$ and $u$ is the minimum number of edges that connect nodes $v$ and $u$. Subsequently, define $\Pi(v)$ as the neighborhood of node $v$, that is the set of nodes from which $v$ can acquire information, namely

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

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

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

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

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

Moreover, let $|Path_\beta(v)|$ be the number of elements of $Path_\beta(v)$.

As an example, setting $\beta = 3$ implies that $Path_\beta(v)$ contains all the 2-hop neighbors $u$ of node $v$ for which no more than 3 different paths exist that connect $v$ to $u$. Therefore, using a low value for $\beta$, is it possible to identify the most weakly connected 2-hop neighbors. Hence, the value of $|Path_\beta(v)|$ is an indicator of the magnitude of node fragility w.r.t. connectivity.

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

On these lines, we introduce the following *probability of a node being vulnerable to failures* $P_\theta(v) \in (0, 1)$:

$$P_\theta(v) = \frac{|Path_\beta(v)|}{|\Pi(v)|} \tag{5}$$

It is worth noting that this quantity can be locally computed by each node, relying on information regarding 1-hop and 2-hop neighbors.

As an example, consider the network in Figure 5 with the instantiation showed in Table 1. It is possible to notice that node 1 is clearly vulnerable because it is relying only on node 9 to communicate with the entire network. The probability assigned to node 1 is $P_\theta(1) = 0.800$. In contrast, node 8 has a single path to node 9, however $|\Pi(8)| = 4$, indicating that alternative paths, with more than 2-hops, can exist, as it is the case. So, its probability of being vulnerable is 0.2500.

### *3.2 Control strategy for robustness improvement*

In this section we introduce a control strategy that, based on the local robustness evaluation procedure introduced in Section 3.1, aims at improving the robustness of the network.

Assume the $i$-th robot identifies itself as vulnerable. In this case, the aim of the control strategy is to increase the number of links towards its 2-hop neighbors that are in

Preprint submitted to 11th IFAC Symposium on Robot Control.
Received March 13, 2015.

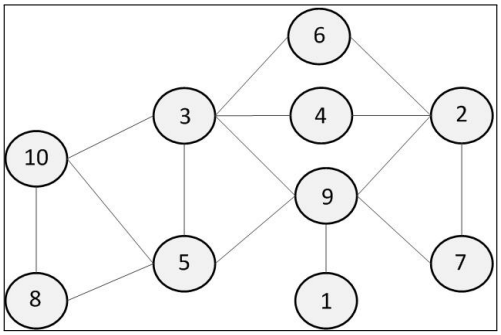


Fig. 5. Example of a network with 10 nodes

$Path_\beta(i)$, for a given value of $\beta$. Hence, define $x^i_\beta \in \mathbb{R}^m$ as the barycenter of the positions of the robots in $Path_\beta(i)$, namely

$$x^i_\beta = \frac{1}{|Path_\beta(i)|} \sum_{j \in Path_\beta(i)} x_j \tag{6}$$

The control law $\mu_i$ in (1) is then defined as follows:

$$\mu_i = \frac{x^i_\beta - x_i}{\left\| x^i_\beta - x_i \right\|} \alpha \tag{7}$$

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

This control law drives vulnerable robots towards the barycenter of the positions of robots in $Path_\beta(i)$, thus decreasing the distance to those robots and eventually creating new edges in the communication graph. It is worth noting that (5) provides a decentralized methodology for each robot to evaluate the probability of being vulnerable.

The control law in (7) is applied in a probabilistic manner, with higher probability for those robots $i$ whose value $P_\theta(i)$ is high. This is obtained comparing $P_\theta(i)$ with a random number $r \in (0, 1)$: if $P_\theta(i) > r$, then the $i$-th robot considers itself as vulnerable, and applies the control law in (7).

## 4. SIMULATION RESULTS

The proposed control strategy was validated using a simulation environment developed in Matlab by the authors. This simulation environment allows to randomly positioning a variable number $N$ of robots in $\mathbb{R}^2$, over a bounded area of size $\mathcal{A}$. Given a communication radius $R$, the communication network is then generated, based on the relative positions among the robots.

Table 1. Neighborhood parameters for the network shown in Fig. 5

| $v$ | $\Pi(v)$ | $\Pi_2(v)$ | $Path_\beta(v)$ | $P_\theta(v)$ |
|---|---|---|---|---|
| 1 | {2,3,5,7,9} | {2,3,5,7} | {2,3,5,7} | 0.800 |
| 2 | {1,3,4,5,6,7,9} | {1,3,5} | {1,5} | 0.285 |
| 3 | {1,2,4,5,6,7,8,9,10} | {1,2,7,8} | {1,7} | 0.222 |
| 4 | {2,3,5,6,7,9,10} | {5,6,7,9,10} | {5,7,10} | 0.428 |
| 5 | {1,2,3,4,6,7,10,8,9} | {1,2,4,6,7} | {1,2,4,6,7} | 0.555 |
| 6 | {2,3,4,5,7,9,10} | {5,7,9,10} | {5,7,10} | 0.428 |
| 7 | {1,2,3,4,5,6,9} | {1,3,4,5,6} | {1,3,4,5,6} | 0.714 |
| 8 | {3,5,9,10} | {3,9} | {9} | 0.250 |
| 9 | {1,2,3,4,6,5,7,8,10} | {4,6,8,10} | {8} | 0.125 |
| 10 | {3,4,6,5,8,9} | {4,6,9} | {4,6,9} | 0.333 |

Repeated experiments were carried out in order to assess the performance of the proposed methodology in a statistically sound manner (see $\#Experiments$ in Table 2). Starting from random positioning, discrete time iterations were performed. At each iteration, we implemented the following procedure:

(1) **network adaptation**: based on the local evaluation of robustness defined in Section 3.1, control law (7) is applied to the fraction $\zeta \in (0, 1)$ of robots that identified themselves as vulnerable,
(2) **robustness evaluation**: computes the number of central nodes required to be removed from the network to achieve a disconnected topology.

Simulations were performed using the parameter set defined in Table 2, with $\beta = 1$, $\alpha = 0.25R$.

Table 2. Model settings for simulation

| $N$ | Area $\mathcal{A}$ | Range $R$ | # Iterations | # Experiments |
|---|---|---|---|---|
| 20 | 50 | 18 | 10 | 200 |
| 100 | 400 | 90 | 50 | 100 |

Results of simulations for $N = 20$ and $N = 100$ robots are in Figures 6 and 7, respectively. Each color depicts one iteration result.

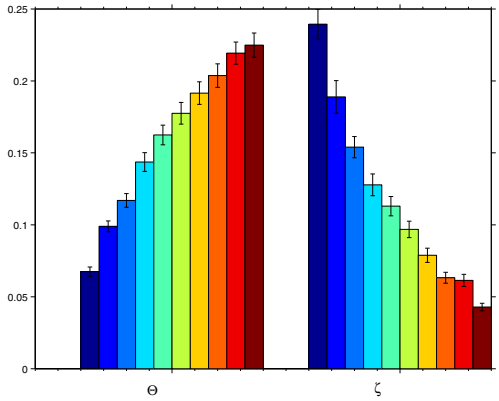


Fig. 6. Adaptive mechanism performance, N=20

For both network sizes, the adaptive mechanism enhanced the initial robustness level more than three times. Also, in general, at each iteration, as the robustness level increases, the number of adaptations performed decreases, demonstrating the effectiveness of the process. The trend is however not monotonic, due to fact that each robot estimates its vulnerability state in a probabilistic manner.

Figures 8 to 10 present the evolution of the network illustrated in Figure 2 as a result of the adaptive mechanism. The vulnerable nodes are identified as blue, the red nodes are those that connect vulnerable nodes to their 2-hops neighbors. As vulnerable nodes rely on these nodes, their loss can be potentially harmful to the network connectivity. Notice that such harmful nodes can also hold a high probability of being vulnerable themselves. Notably, in seven iteration (the sixth is not shown), the topology has evolved to a more robust one.

Figure 11 presents the evolution for the network introduced in Section 2.3 (see Figure 3). Despite the number of nodes in the network, it is possible to notice how the critical configuration could be reverted after 22 iterations.

Preprint submitted to 11th IFAC Symposium on Robot Control.
Received March 13, 2015.

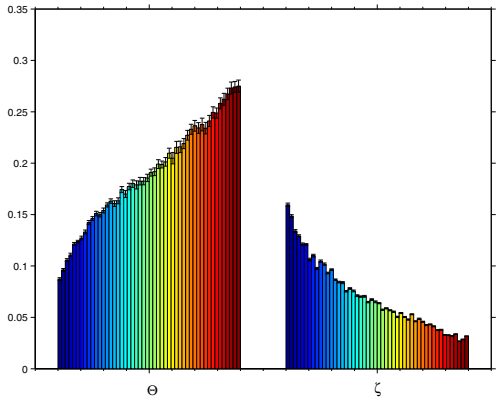


Fig. 7. Adaptive mechanism performance, N=100

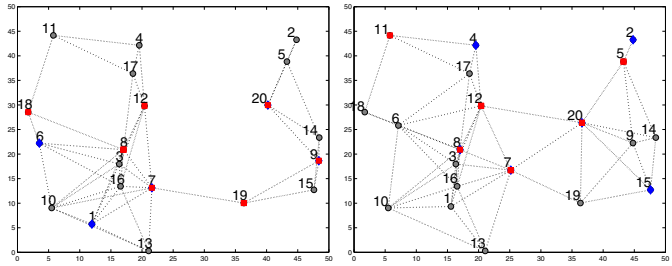

Fig. 8. Example of the adaptive mechanism performance - Iterations 1 and 2, N=20

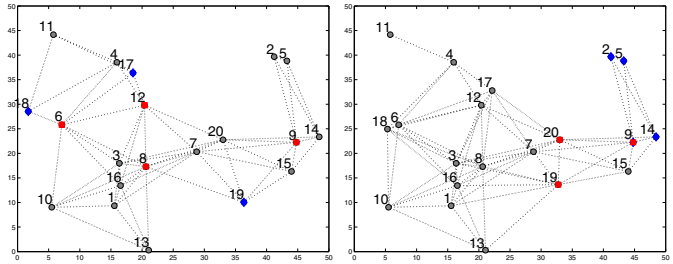

Fig. 9. Example of the adaptive mechanism performance - Iterations 3 and 4, N=20

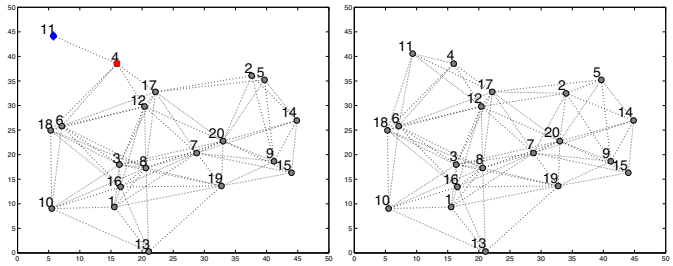

Fig. 10. Example of the adaptive mechanism performance - Iterations 5 and 7, N=20

Some additional examples can be freely viewed online [2].

## 5. CONCLUSION

This paper discusses the robustness of robot networks regarding failures, an issue that is underestimated in the design of multi-robot applications. The relevancy of the subject was introduced through analysis and some

[2] *http://www.arscontrol.unimore.it/syroco15_networks/*

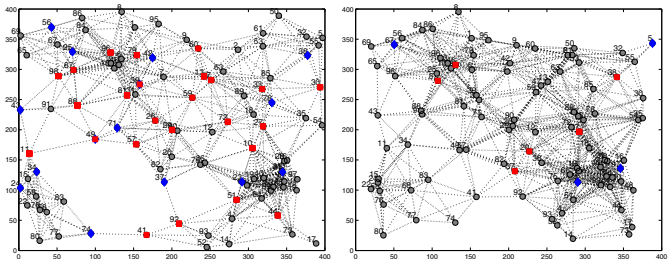

Fig. 11. Example of the adaptive mechanism performance - Iterations 1 and 22, N=100

demonstrative examples. We then proposed local procedures for addressing the problem that consider robots as autonomous agents relying only on their direct and 2-hop neighboring information to evaluate their state, and — when it is necessary — their new position. A protocol and a measure for evaluating the level of network robustness was set, and the results demonstrate the feasibility of the proposed approach. In fact, the mechanism for detecting vulnerable states has proved to be efficient, and despite the simplicity of the adaptation strategy, it was able to improve the network robustness in simulated environments with 20 and 100 robots. For future work, this strategy will incorporate new features such as collision avoidance, efficiency measures and cost functions in a fully mobile multi-robotic domain.

## REFERENCES

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

Albert, R., Jeong, H., and Barabasi, A.L. (2000). Error and attack tolerance of complex networks. *Nature*, 406(6794), 378–382.

Bigdeli, A., Tizghadam, A., and Leon-Garcia, A. (2009). Comparison of network criticality, algebraic connectivity, and other graph metrics. In *Proceedings of the 1st Annual Workshop on Simplifying Complex Network for Practitioners*, SIMPLEX '09, 4:1–4:6. ACM, New York, NY, USA.

Cao, Y. and Ren, W. (2010). Distributed coordinated tracking via a variable structure approach – part I: consensus tracking. part II: swarm tracking. In *Proceedings of the American Control Conference*, 4744–4755.

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

Dall'Asta, L., Barrat, A., Barthelemy, M., and Vespignani, A. (2006). Vulnerability of weighted networks. *Theory and Experiment*, 2006, 04006.

Dimarogonas, D.V. and Johansson, K.H. (2010). Bounded control of network connectivity in multi–agent systems. *IET Control Theory & Applications*, 4, 1751–8644.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98), 298–305.

Ghedini, C. and Ribeiro, C.H.C. (2011). Rethinking failure and attack tolerance assessment in complex networks. *Physica A: Statistical Mechanics and its Applications*, 390(23–24), 4684–4691.

Preprint submitted to 11th IFAC Symposium on Robot Control.
Received March 13, 2015.

Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*. Springer.

He, Z., Liu, S., and Zhan, M. (2013). Dynamical robustness analysis of weighted complex networks. *Physica A: Statistical Mechanics and its Applications*, 392(18), 4181 – 4191. doi: http://dx.doi.org/10.1016/j.physa.2013.05.005.

Hsieh, M.A., Cowley, A., Kumar, V., and Talyor, C.J. (2008). Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*, 25(1), 111–131.

Ji, M. and Egerstedt, M. (2007). Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*.

Koschützki, D., Lehmann, K.A., Peeters, L., Richter, S., Tenfelde-Podehl, D., and Zlotowski, O. (2005). Centrality indices. In *Network analysis*, 16–61. Springer.

Manzano, M., Calle, E., Torres-Padrosa, V., Segovia, J., and Harle, D. (2013). Endurance: A new robustness measure for complex networks under multiple failure scenarios. *Computer Networks*, 57(17), 3641 – 3653. doi: http://dx.doi.org/10.1016/j.comnet.2013.08.011.

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

Newman, M.E.J. (2003). The structure and function of complex networks. *SIAM Review*, 5(2), 167–256. doi: 10.1137/S003614450342480.

Notarstefano, G., Savla, K., Bullo, F., and Jadbabaie, A. (2006). Maintaining limited–range connectivity among second–order agents. In *Proceedings of the American Control Conference*, 2134–2129.

Sabattini, L., Chopra, N., and Secchi, C. (2013a). Decentralized connectivity maintenance for cooperative control of mobile robotic systems. *The International Journal of Robotics Research (SAGE)*, 32(12), 1411–1423.

Sabattini, L., Secchi, C., and Chopra, N. (2014). Decentralized estimation and control for preserving the strong connectivity of directed graphs. *IEEE Transactions on Cybernetics*.

Sabattini, L., Secchi, C., Chopra, N., and Gasparri, A. (2013b). Distributed control of multi-robot systems with global connectivity maintenance. *IEEE Transactions on Robotics*, 29(5), 1326–1332.

Secchi, C., Sabattini, L., and Fantuzzi, C. (2013). Decentralized global connectivity maintenance for interconnected lagrangian systems in the presence of data corruption. *European Journal of Control*, 19(6), 461–468.

Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I., Ruess, F., Suppa, M., and Burschka, D. (2012). Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *Robotics Automation Magazine, IEEE*, 19(3), 46–56. doi:10.1109/MRA.2012.2206473.

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

Preprint submitted to 11th IFAC Symposium on Robot Control.
Received March 13, 2015.