Decentralized local-global connectivity maintenance for networked robotic teams

Koresh Khateri a, Mahdi Pourgholi a,*, Mohsen Montazeri a, Lorenzo Sabattini b

a Department of Electrical Engineering, Shahid Beheshti University, Evin, Tehran 19839-4716, Iran
b Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, via Amendola 2, Reggio Emilia 42122, Italy

Article history:
Received 24 April 2018
Revised 17 April 2019
Accepted 21 June 2019
Available online xxx

Keywords:
Communication delay
Connectivity maintenance
Distributed robotic systems
k-hop routing

Abstract
A prerequisite for a team of robots to cooperate is to maintain connectivity among robots. Connectivity maintenance has been extensively studied recently and several local and global connectivity maintenance algorithms have been proposed for the distance dependent communication topology. Local methods are known to be very restrictive and it will be shown that global methods, based on power iteration estimation, could be sluggish in the presence of communication delay and non-converging in large and sparse robot teams. Therefore, a method based on k-hop routing is proposed, where k is a design parameter to determine the locality level of the proposed method. Before any link disconnection, a test for an alternative k-hop path is executed and, based on its result, the disconnection is allowed or blocked. Blocking displacement for maintaining connectivity will eventually immobilize (freeze) the network. Therefore, a procedure for unfreezing the network is also proposed. Simulation results are provided to further investigate the proposed method.

© 2019 European Control Association. Published by Elsevier Ltd. All rights reserved.

1. Introduction
Connectivity maintenance has been an active research field in the last decades and it has been widely investigated. One of the requirements for decentralized coordination and control methods, in multi-robot systems, is the connectivity of the communication graph [3,38], which is relevant for accomplishing many cooperative multi-agent missions as collaborative transportation, search and rescue tasks [9,13] and decentralized control frameworks [16,21]. Therefore, connectivity preservation is a critical design consideration in mobile multi-robot systems and almost essential for any kind of team working in multi-robot systems.

In a network of mobile robots with limited range communication links, the movement of robots can cause link disconnection. Therefore, a connectivity maintenance procedure should provide a supervisory or an integrated control method to preserve connectivity of the network (i.e., maintenance of at least one communication path between each two robots). However, preserving connectivity is not the primary goal of the multi-robot system. Therefore, the main controller or the path planner should have enough freedom to reach the primary goal, which is different from connectivity maintenance, like reaching a place in the aggregate movement of a group, in an immigration problem. In other words, connectivity maintenance procedure should take action only when it is essential (i.e., any further movement may cause a disconnection).

Several approaches have been proposed to maintain connectivity of the graph related to the interconnections among robots. Mainly two different approaches can be cited, with regard to the amount of information exchange required to accomplish this goal. Authors in [1,17] have proposed a local approach, where the controller design is based on preservation of all initially existing links. In local methods, each robot only needs to access current states or outputs of its immediate neighbors and the time requirement to gather such an information is equal to a 1-hop communication delay, which can be safely neglected in many applications. On the other hand, local connectivity maintenance approach is restrictive with respect to the freedom of movement of the robots: in fact, keeping every initial connection leads to constraining the mobility of the robots, when disconnection of a certain link may not result in two disconnected sub-graphs (i.e., a disconnected network). Global connectivity maintenance, on the other side, has been investigate recently [11,29,31,39] and the goal is to preserve connectivity of the network in a global sense (i.e., preserving the possibility of communication for each two robots with a multi-hop path, while some initially existing links might disconnect), which is sufficient for cited multi-robot tasks and many others to be accomplished. Global connectivity algorithms are mainly based on

* Corresponding author.
E-mail addresses: k_khateri@sbu.ac.ir (K. Khateri), m_pourgholi@sbu.ac.ir (M. Pourgholi), m_montazeri@sbu.ac.ir (M. Montazeri), lorenzo.sabattini@unimore.it (L. Sabattini).

https://doi.org/10.1016/j.ejcon.2019.06.008
0947-3580/© 2019 European Control Association. Published by Elsevier Ltd. All rights reserved.

Please cite this article as: K. Khateri, M. Pourgholi and M. Montazeri et al., Decentralized local-global connectivity maintenance for networked robotic teams, European Journal of Control, https://doi.org/10.1016/j.ejcon.2019.06.008
an ascending gradient method to increase the algebraic connectivity of the associated communication graph. To find the up hill direction in this problem, a power iteration based decentralized estimation procedure had been proposed to estimate the eigenvector associated with the algebraic connectivity, which is needed for finding the ascending direction of the algebraic connectivity itself [2,10,28,39]. However, in this paper, sluggishness of the global connectivity maintenance in certain circumstances will be shown and a method based on existence of an alternative k-hop path between robots with disconnecting links is proposed, where k is a design parameter for tuning the desired locality of the algorithm. Our method is a link monitoring algorithm which allows or blocks the disconnection of links that are going to be disconnected by robots movements. Actually, the proposed algorithm cannot be classified as the traditional connectivity preserving algorithms. First, a simple procedure for connectivity maintenance of a robotic network without communication delay is developed, and then the effect of communication delay and dynamical constrains is considered. Considering delay in connectivity maintenance is a very new topic. In [40] the delay is considered to be time-varying while the delay is always less than a maximum upper bound for each edge. Also, authors in [3,32] had considered an upper bound of delay time for each one hop communication.

A summary of our contributions are as follows:

1. Power iteration based algebraic connectivity estimation tool is analyzed and it is shown that, in certain circumstances, it might be sluggish and the assumptions considered for its proper operation might not be practical.
2. A new connectivity maintenance method is proposed, that allows to tune the level of locality, from a completely local algorithm with \( k = 0 \) to a global algorithm with \( k = N - 1 \).
3. The effect of communication delay and discrete communication is then considered in the proposed algorithm.
4. Connectivity maintenance problem for an N-robot multi-robot system is decomposed into N output constrained control problems to be solved by each robot based on local information.

The rest of the paper is organized as follows: in the next section, necessity for a new approach with tune-able level of locality is discussed. Section 3 provides needed background on graph theory and illustrates a modified algorithm for finding a path with certain properties in ad-hoc networks. Section 4 is a statement of the connectivity maintenance problem and, in Section 5, our algorithm to maintain connectivity is presented. Simulation results for 3 different scenarios are shown in Section 6, and the paper is concluded in Section 7.

2. Related works

Recent works in the area of connectivity maintenance could be categorized based on several features as locality, boundedness of the control input, dynamic behavior of the robots and time delay considerations.

With regard to the locality of the proposed methods, local connectivity maintenance based on preserving initial links of a connected network, to preserve its connectedness property, has been investigated in multiple works [14,17,37]. Artificial potential functions to provide attraction between robots when their distance increases are proposed in [35–37] for preserving local connectivity while reaching a flocking goal. In these works, gradient of the artificial potential function is added to the control input of the robots to artificially impose the existence of the potential function. Recently authors in [27–30,39] have discussed about achieving connectivity from a global point of view, to attain more flexibility. The idea is to allow disconnections of initially existing links, while these disconnections could be tolerated and connectivity of the network is preserved (i.e., a criteria for network connectivity like positiveness of the second smallest eigenvalue of the Laplacian matrix is satisfied). However, such a global solution is not always achievable, without restricting the speed of robots in the presence of communication delay. To show this, simply consider robots with maximum speed \( S_{\text{max}} \) and 1-hop communication delay \( \tau \), with a limited range of communication which we call \( R \) hereafter. In such a network, an 1-hop communication delay is \( \tau R \) and the maximum, non looped, length of a communication path in a network with \( N \) robots is \( N - 1 \). Also, consider the condition where there is only an \((N - 1)\)-length communication path between two robots in the network, which is the worst case. The time needed to exchange information between these robots is equal to \((N - 1)\tau \).

Therefore, even a centralized estimation of connectivity (i.e., sending all information to one of the robots) at least requires a time larger than \((N - 1)\tau \) to retrieve all the information needed for calculating the adjacency matrix and provide a reliable output about connectivity. Consider two robots in opposite direction and velocity magnitude \( S_{\text{max}} \), with only an \((N - 1)\)-length communication path connecting each other: their relative displacement before a reliable output from the connectivity estimator can be up to \( 2S_{\text{max}}(N - 1)\tau \). Therefore, if \( R < 2S_{\text{max}}(N - 1)\tau \), the network could be disconnected, just before obtaining any reliable output from the connectivity estimator. Thus, for any multi-robot system with \( R < 2(N - 1)\tau S_{\text{max}} \), global connectivity maintenance is not achievable for \( N \) robots. In other words, the upper bound on the maximum achievable robot speed is less than \( \frac{\tau}{2R + \tau} \), for any method guaranteeing global connectivity in the presence of communication delay as in [32], even with the assumption of unbounded control input. On the other hand, with the same line of argument, this upper bound on the speed for a \( k \)-hop Connectivity maintenance algorithm is \( \frac{\tau}{2kR + \tau} \) and consequently \( \frac{\tau}{2kR + \tau} \) for local connectivity maintenance. Therefore, global connectivity preservation may be even more restrictive than the local approach of maintaining initial connections, in certain circumstances, where the displacement of robots in a time delay is comparable to the limited range of communication or with a highly populated robotic system on a sparse associated graph. It is worth noting that according to these calculated upper bounds, speed restriction for global connectivity is of \( O(\frac{1}{R}) \) (i.e., reciprocally more restriction with the scale of the network). While, speed restriction for \( k \)-hop connectivity maintenance is of fixed order \( O(\frac{1}{kR}) \) with respect to the size of multi-robot system. A more detailed comparison between local and global connectivity maintenance is given in [22].

The aforementioned works are considering the possibility of using unbounded control inputs to solve the connectivity maintenance problem, which is not practical in real world robotic systems. To solve this issue, boundedness of the control input has been investigated in [5,11,12,23]. Connectivity maintenance control input, in these works, is added to another bounded input needed for achieving any specific goal of the robotic system. Also, boundedness of the control input puts another restriction on the connectivity maintenance procedures, which is unavoidable from a practical point of view: in many robotic systems the control input is the acceleration of the robots. Therefore, a robot is not able to stop instantly. Hence, there will be a displacement after a stop command from a path planner, which puts another upper bound on the speed of the limited communication range multi-robot system. To show this, consider the maximum initial inter distance among the robots as \( R_{\text{max}}(0) \) and let \( \Delta R = R - R_{\text{max}}(0) \). Consider two robots with this maximum distance link to be moving in opposite directions with the maximum speed magnitude: the robots should be able to stop, if their movement is going to result in a disconnection. Therefore, each robot should be able to stop before having a
displacement more than $\Delta R/2$. With a fully actuated robot, in the best case (i.e., possibility of determining direction of the acceleration to be in opposite direction of speed), the maximum speed must be less than $\sqrt{2} \Delta R$, for the robot to be able to stop with a displacement less than $\Delta R/2$, where $\beta$ is the maximum magnitude of the bounded acceleration and $R$ is the communication range.

Different types of robotic mechanical systems and various classes of dynamic models have been investigated in the recent works on connectivity maintenance. Authors in [20] proposed a formation with connectivity preservation algorithm for robots with single integrator dynamics. Flocking with connectivity maintenance has been investigated for double integrator with damping in [6]. Robots with Lagrangian model have been considered in [29,30,32] and connectivity maintenance in rendezvous problem with unknown additive dynamic and disturbance is investigated in [7]. Connectivity maintenance in the presence of communication delay has been examined in [32].

In this paper, locality of the proposed algorithm can be deliberately chosen. Therefore, it will be possible to tune the search distance for an alternative path and the designer will be able to match the upper bound on the speed limit, in delayed communication graphs, with the physical capability of the robots. The control input is considered to be bounded, and no restriction is considered with the dynamic model of the robots as long as there exists a controller capable of steering the robot to a desire point, while satisfying a (to be defined) output constraint, which we have considered as assumptions in this work.

The proposed algorithm is a supervisor on the reference signal provided by the path planner and the problem of connectivity maintenance is decomposed into designing a controller for the robots to track the reference input and a supervisor to change the reference input, if the planned path is leading to disconnection.

3. Preliminaries

A graph theoretic description of the interconnection topology of a multi-robot system is provided in the following subsection and a modified dynamic source routing method for finding an alternative path, disjoint from a set of edges, which is used as a function in our main algorithm, is described in Section 3.2. The following equation is used multiple times, for calculating the maximum traveling distance with a constant break acceleration to stop a robot from moving:

$$\Delta x = \frac{s^2(0)}{2a}$$

where $\Delta x$ is the traveling distance, $s(0)$ is the initial speed and $a$ is the break acceleration. Indexing variables in this work are defined locally and it might happen to redefine them in several places without making ambiguity. A spherical sector, in this paper, is a portion of a sphere defined by the intersection of a cone with apex at the center of the sphere. The axis of the sector is considered to be the axis of the defining cone.

3.1. Graph theoretic concepts in multi-robot systems

Connections between $N$ robots with limited connectivity range, moving in an $n$-dimensional space, is modeled using a distance dependent graph which, based on movement of the robots, could be time-varying. Let $G(t) = (V,E(t))$ be a graph and the set of nodes or vertices be $V = \{v_1, \ldots, v_N\}$, where $E = \{e_{ij} \in V \times V \mid A_{ij} > 0\}$ is the set of edges and $A$ is the adjacency matrix associated with the graph, such that $A_{ij} = 0$ and $A_{ij} > 0$ if there is a link from node $i$ to node $j$. The neighbors of node $v_j$ are $N_j = \{v_j \in V \mid A_{ij} > 0\}$. The graph is called undirected if $A = A^T$, otherwise it is called directed.

A path is a sequence of edges that connects a sequence of vertices. A sub string of a path is a sub sequence of the vertices in the path. Two edge-disjoint (or edge-independent) paths have no edge in common. A path is disjoint from a set $S_e$ of edges if none of the edges in $S_e$ are contained in the path. The length of a path is the number of edges included in that path and a $k$-hop path is a $k$-length path with $k-1$ nodes as hops between the source and the destination.

Let $x_i \in \mathbb{R}^n$ denote the position of $v_i$; then the distance dependent adjacency matrix, for an undirected communication (i.e., duplex links), is defined as follows:

$$A_{ij}(t) = \begin{cases} 1 & \|x_i(t) - x_j(t)\| \leq R \\ 0 & \text{otherwise} \end{cases}$$

Adjacency matrix for a weighted graph is defined as $A_{ij}(t) = w_{ij}(t)$, where $w_{ij}(t)$ is the weight associated to the edge from vertex $i$ to vertex $j$.

The degree matrix associated with a graph is defined as:

$$D_{ij}(t) = \begin{cases} \sum_{k \neq i} A_{ik}(t) & i = j \\ 0 & \text{otherwise} \end{cases}$$

Let $L(t) = D(t) - A(t)$ be the Laplacian matrix. The second smallest eigenvalue $\lambda_2$ of the Laplacian matrix is known to be an index of the connectivity of the graph.

A communication topology with equal range of communication for all the robots will be called homogeneous limited range hereafter.

Theorem 1 [26]. In an undirected graph, the following statements are equal:

(a) The graph is connected.
(b) There exists at least one path between every pair of the nodes in the graph.
(c) There is at least a spanning tree.
(d) $\lambda_2 > 0$.

Definition 1. A $k$-local connectivity maintenance algorithm is a method that allows an edge disconnection only if there is an alternative $k$-length path for that edge, disjoint from the set of other disconnecting edges.

Definition 2. A disconnecting edge is an edge with an inter distance larger than some specific value, defined in Eq. (9), while the robots are still able to prevent disconnection of that edge.

0-local connectivity is trivially identical with the local connectivity definition, as defined in [23,29–31]. It will be shown that $k > (N-1)$, for $k$-local connectivity maintenance, is meaningless and $(N-1)$-local connectivity is equivalent to global connectivity, defined in other works.

A complete (or all to all) graph has an edge between each pair of its nodes and a circular graph has exactly two edges for any node, and it is connected.

The algebraic connectivity is bounded and its amount is dependent on the definition of weights associated with the edges in a weighted graph (i.e., $A_{ij} = w_{ij}$), which consequently impacts the definition of the adjacency matrix and the Laplacian matrix. For example, a scalar multiplication of weights has the same effect on the algebraic connectivity (i.e., $A_{ij}(\alpha L) = \alpha A_{ij}(L)$).

Lemma 1 [4,8]. Assume an undirected graph with $N$ nodes, and with $\max_{ij} D_{ij} \leq N - 1$. For such a graph, the maximum achievable algebraic connectivity is $N$, and it can be achieved by a complete graph.

3.2. Routing in ad-hoc networks

An ad-hoc routing protocol is a way of finding a communication path between two nodes in a network, with a decentralized
procedure, where the nodes are not connected to any centralized manager. Two main types of routing protocols in ad-hoc networks are proactive (or table driven) and reactive (or on-demand) algorithms. The proactive algorithms, like the optimized link state routing protocol [15], keep fresh available route lists based on periodic messages among nodes. On the other hand, for on demand routing protocols, like dynamic source routing (DSR) [19] and mobility and direction aware ad-hoc on demand distance vector [33], a route discovery function is activated, whenever there is a need to send a new information packet through an unknown route. Also, there are hybrid and hierarchical ad-hoc routing procedures, that have features of both basic procedures [34].

In this paper, ad-hoc routing is used to assess if a link can be disconnected. In particular, given a link to be disconnected, ad-hoc routing is used to find the availability of an alternative k-hop path that does not contain such links.

The disconnecting links should not be used by the routing procedure and a universal clock is not considered in our work. Therefore, we will use a maximum k-hop propagation of the discovery phase (i.e., robots ignore a discovery message if it is exchanged more than k times) instead of using a dead time. However, the robot which had requested the search for an alternative path uses a local timer for indicating a dead time and waits only \((k + 1)\tau_{\text{max}}\) for the return of the discovery message, which also does not require a universal clock. However, the robot which had requested the search for an alternative path uses a local timer for indicating a dead time and waits only \((k + 1)\tau_{\text{max}}\) for the return of the discovery message, which also does not require a universal clock. We have added a set of links that are not allowed to forward discovery messages, to avoid using those disconnecting links in the routing process. Also, in this work, we are searching for an alternative path. Therefore it is not needed to announce the found path using the found path itself, and it is possible to use the direct disconnecting edge itself to announce the existence of the found alternative path. This modified discovery function works as follows:

1. On demand, a discovery message containing \(k + 1\) id blocks and the set of my disconnecting edges (i.e., disconnecting edges of the robot running the algorithm) will be created by the source, with the first and the last blocks filled by the source and the destination ids, respectively. This message will be broadcast on all of the source edges excluding those that are in the set of disconnecting edges.

2. When receiving a discovery message, each node investigates if it is itself the destination: if so, the found path will be announced to the source. Otherwise, the receiver of the discovery will search for its own id in the id blocks: if its id is not in the message and there exist a non filled id block, the receiver will register its id at the first empty block and the message will be broadcast on all of its edges excluding those that are in the set of disconnecting edges. Otherwise, if blocks are all filled or the node has already registered its id in the message, the receiver will not propagate the message.

Algorithm 1 illustrates our modified version of the path discovery method. Each robot is considered to have a unique id and the algorithm is event-driven (i.e., the algorithm does nothing until an event occurs). The Demand event is triggered whenever there is a need to find a path between the robot and another robot with id = Destination Id. My_Id is the id of the robot itself, which is running the code. My Disconnecting Edges is the set of disconnecting edges of the robot. The Broadcast(., .) function sends its first argument on all the edges connected to the broadcasting node but those in the second argument. The Receive event occurs when a discovery Message is received from a neighbor. The symbol \(\cdot^*\) as an operator is used to extract a part or property of an object (e.g., Message.ID is the ID list in the Message). The Register(., .) function inserts the first argument in the first empty block of the second argument. The Announce(Message, Source_Id) function sends the discovered path to the source robot which has initiated the discovery message.

4. Problem statement

The concern of this paper is to design a decentralized supervisor, such that movement in the robotic network with distance dependent graph defined in (2) will not result in a disconnected network (i.e., non-satisfaction of one of the items in Theorem 1), while a 1-hop communication is delayed at most \(\tau_{\text{max}}\) seconds.

Each robot is only able to determine its own input and only the information about its neighbors are accessible. There is no assumption directly on the dynamic model of the robots. The assumptions are a maximum velocity magnitude and a controller existence, that will be subsequently defined in (8) and (7). Also, there is no assumption about homogeneity of robots dynamic model.

5. Main results

In this section, we will introduce our approach toward connectivity maintenance. First, the sluggishness of the decentralized power iteration based method, in certain conditions, will be shown. A basic version of our proposed algorithm is proposed in Section 5.2, and realistic considerations as communication delay and mechanical constraints are investigated in Section 5.3.

5.1. Power iteration based connectivity maintenance

The approach of the power iteration method is to determine the eigenvector of a matrix that is corresponding to the largest eigenvalue, using iterated multiplication of the matrix by a random vector. Assume \(F \in \mathbb{R}^{m \times m}\), with \(m\) as the size of matrix \(F\), as a diagonalizable matrix, with \(v_1, \ldots, v_n\) as its eigenvectors and its ordered eigenvalues as \(\lambda_{v_1}, \ldots, \lambda_{v_n}\), where \(\lambda_{v_i}\) is the largest eigenvalue in magnitude. Then, \(F^2 = v_1 \Lambda F^{-1} v_1^T\), where \(\Lambda = \text{diag}(\lambda_{v_1}, \ldots, \lambda_{v_n})\) and \(v_F\) is the matrix of eigenvectors of matrix \(F\). Assume a random vector \(\Omega\) and let \(\Omega = v_F^{-1} \Omega\). Then, \(F^2 \Omega = F v_F \Omega = \sum_{i=1}^n v_i \lambda_{v_i}^2 \Omega_i\), where \(\Omega_i\) is the jth element of \(\Omega\). Therefore:

\[
F^2 \Omega = \lambda_{\Omega_j}^2 \left( \sum_{j=1}^n v_j \left( \frac{\lambda_{v_j}}{\lambda_{\Omega_j}} \right)^2 \Omega_j \right)
\]  

(4)
For all \( j \neq \omega, (\lambda_{\omega j}^L)^2 \) converges to zero, when \( z \to \infty \). Therefore, 
\[
\lim_{z \to \infty} F\Omega = \lambda_{\omega \omega}^L \Omega V_{\omega \omega},
\]
that is the eigenvector corresponding to the largest eigenvalue of matrix \( F \), up to a multiplier. In each iteration, the result is normalized after each multiplication to avoid the result to grow unboundedly or diminish to the zero vector.

Authors in [28,39] have proposed a decentralized form of the power iteration method to find the eigenvector in the subspace \( P = \{ r \in \mathbb{R}^m \mid r \perp \mathbf{1}^T \} \). By using a spectral shift transformation (i.e., \( Q = I - \alpha \mathbf{1}\mathbf{1}^T \)), to reverse the eigenvalues order, where \( I \) is the identity matrix and \( \mathbf{1}^T \) is an \( \omega \)-dimensional vector with all elements equal to one. Their estimator resulted in an estimation of the eigenvector corresponding to the second biggest eigenvalue of \( Q \), that is the second smallest eigenvalue of the Laplacian matrix.

The error of estimation of the centralized form of power iteration method is \( e = \frac{\lambda_{\omega 0}^L}{\lambda_{\omega \omega}^L} \left( \frac{\lambda_{\omega j}^L}{\lambda_{\omega 0}^L} \right)^2 \), where the biggest term has a coefficient equal to \( \left( \frac{\lambda_{\omega 0}^L}{\lambda_{\omega 0}^L} \right)^2 \). Therefore, the error is exponentially converging with the rate \( \frac{\lambda_{\omega 0}^L}{\lambda_{\omega \omega}^L} \). However, it is quite common in structural graphs (e.g., complete or circular graphs) to have a Laplacian matrix with similar second and third smallest eigenvalues. In this case, the spectral transformed problem will not converge (i.e., \( \frac{\lambda_{\omega 0}^L}{\lambda_{\omega \omega}^L} \left( \frac{\lambda_{\omega j}^L}{\lambda_{\omega 0}^L} \right)^2 = \frac{\lambda_{\omega 0}^L}{\lambda_{\omega \omega}^L} \left( \frac{\lambda_{\omega j}^L}{\lambda_{\omega 0}^L} \right)^2 \)). Even though, in the derivations of the decentralized power iteration method [39], there exists an equilibrium for the case of similar eigenvalues, the dynamic estimator will converge to a vector in a (\( \omega - 1 \))-dimensional ball perpendicular to the eigenvectors associated with \( \lambda_1 \neq \lambda_2 \), where \( \lambda_1 = 0, \lambda_2 = \ldots = \lambda_\omega < \lambda_{\omega+1}, \ldots, \lambda_\omega \) are eigenvalues of the Laplacian matrix. However, the convergence of the estimator only guarantees a bounded error and nothing more than a simple normalized guess about associated eigenvector with \( \lambda_2 \) in the described ball.

Results for connectivity maintenance based on decentralized power iteration [28,29,31] are based on an initial excess connectivity assumption (i.e., \( \lambda_2(0) > \eta \), where \( \eta \) is a bounded value greater than the sum of estimation error bound of \( \lambda_2 \) and a desired minimum algebraic connectivity). However, as described in Lemma 1, \( \lambda_2(0) \) is a property of the initial communication graph and it is also bounded (it might be even smaller than the initial estimation error). In other words, for the initial estimate, it is possible to choose a number in the interval \([0, \omega]\) as the initial guess about \( \lambda_2(0) \), where \( \lambda_2 \) is the output of the estimator and \( \max(D_{ij}) < \omega - 1 \). Also, \( 0 < \lambda_2(0) < \omega \). Therefore, it could happen that \( \lambda_2(0) < |\lambda_2(0) - \lambda(0)| \), which means \( \lambda_2(0) < \eta \), that is in contrary with the assumption made in the aforementioned decentralized power iteration based methods. Also, in the condition of repeated eigenvalues equal to \( \lambda_2 \), the estimation error is not converging and the assumption of excess connectivity may not be satisfied even running the estimator for a long time before execution of the system, without any movement of the robots.

The decentralized power iteration algebraic connectivity estimate error converges with a slower rate than the centralized form. Therefore, the decentralized \( \lambda_2 \) estimation error rate of convergence is slower than \( g(\lambda_2)^2 \) with \( g \) as some coefficient. Thus, for the error to be slower than some specific value \( c \), the required number of iterations is \( z_c > \frac{\log(1/c)}{\log(\lambda_2)} \). Length of the longest path in a network with \( N \) nodes is \( N - 1 \) and the delay for communication over this path is \((N - 1)\tau_{\max}\). Therefore, the time needed for gathering information from all nodes to execute one iteration is at most \((N - 1)\tau_{\max}\) seconds. Thus \( z_c(N - 1)\tau_{\max} \) seconds are needed for the estimation error of \( \lambda_2 \) to be smaller than some specific value \( c \). In this period of time, the distance between two robots with opposite directions can grow to \( 2z_c(N - 1)\tau_{\max} \).

5.2. Basic local-global connectivity maintenance method

With increasing number of agents and the sparsity of the communication graph, the complexity of a global solution to the connectivity maintenance problem also increases. Therefore, in this subsection a method based on k-hop connectivity is introduced in which the level of locality can be tuned based on networks requirements and specifications.

Based on the definition of connectivity, which is availability of a communication path between any two different nodes of the network, our proposed method investigates the existence of a k-hop path between robots whose movement may disconnect some existing communication links.

Lemma 2. Disconnecting a set of edges \( D_E = \{ (v_i, v_j), \ldots, (v_i, v_k) \} \) from the node \( v_l \) will not cause loss of connectivity in a connected undirected graph \( G \), if there is a k-hop alternative edge-disjoint path from the set \( D_E \) for each disconnecting edge.

Proof. Because of connectivity, there is a path between each pair of nodes, initially. For all disconnecting edges, with the assumption made, there is an alternative edge-disjoint path from \( D_E \). Therefore, pair of nodes on sides of edges in \( D_E \) will remain connected.

Consider \( P((v_i, v_j)) \) as the alternative edge-disjoint path from \( D_E \) for the edge \((v_i, v_j)\), where \( v_j \) is one of the nodes on one side of a disconnecting edge of \( v_i \). Then, for each pair of nodes \((v_i, v_j)\) that are connected by a path involving an edge in \( D_E \), there exists another path disjoint from \( D_E \). To show this, consider the non disjoint path involving the edge \((v_i, v_j)\) and the initial connecting paths \( P((v_i, v_j)) = \{ (v_i, v_p), \ldots, (v_i, v_k), \ldots, (v_i, v_j) \} \). Then, an alternative path disjoint from \( D_E \) for \( P((v_i, v_j)) \) is \( P((v_i, v_j)) = \{ (v_i, v_p), \ldots, P_1((v_i, v_k)), \ldots, (v_i, v_j) \} \). Therefore, there will be an alternative disjoint path from \( D_E \) for each pair of nodes and the network remains connected after disconnecting \( D_E \).

Fig. 1 further illustrates Lemma 2. In Fig. 1a the disconnecting edge will be allowed to disconnect, because there is an alternative path to connect the vertices on sides of the disconnecting edge. While, in Fig. 1b, the network will be divided into two isolated networks so the disconnection will not be allowed. Fig. 1c shows a condition where the alternative path has a disconnecting edge and there is no alternative path excluding the set of all disconnecting edges \( (D_E) \). Therefore, disconnection in this case will not be allowed.

With the movement of a robot, more than one of its incident edges could be disconnected: hence, the existence of an alternative path should be checked for all of them in the general case. However, in the homogeneous limited range problem, neighbor nodes...
could be clustered in connected sub-graphs to decrease the number of investigations, while multiple edges are disconnecting.

**Lemma 3.** For connectivity preservation, while multiple edges are going to be disconnected, in the homogeneous limited range problem with radius $R$, it is enough to investigate an alternative, disjoint from disconnecting edges path, only for one of the nodes in each connected sub-group of neighbors.

**Proof.** In the homogeneous limited range problem, each node based on its neighbors’ location, is able to compute which ones of its neighbors are neighbors themselves. Therefore, a node is able to partition its neighbors in connected disjoint sub-graphs, which are defined by the connected disjoint sub-graphs of the neighbors. Consider the case where the movement of node $v_i$ will disconnect the set of edges $D_{E_i} = \{(v_i, v_j), \ldots, (v_i, v_k)\}$. Let $V_2 = \{v_j, \ldots, v_k\}$ (i.e., the set of all vertices adjacent to the edges in the set $D_{E_i}$ except for $v_i$) and consider a partition of $V_2$ into $k$ sub-sets $S_{V_i} = V_2$, such that each set of vertices $S_{V_i}$ with its associated edges constructs a connected sub-graph. If there is a path between $v_i$ and $v_j$, $v_k \in S_{V_i}$, with $1 < y < k$, then by connectivity of $S_{V_i}$ there also exist a path between $v_i$ and $v_j$. Therefore, connectivity will be preserved for disconnection of multiple edges, if there is an alternative path for one representative in each connected partition of neighbors. □

Lemma 3 helps reducing the number of needed searches for alternative paths. For example, in $\mathbb{R}^2$, the circular coverage of a node can be partitioned into 6 equal sectors. The maximum distance between any two points in these sextants is less than $R$: then, any pair of robots in these sectors are already connected. Therefore, the neighbors of each node can be partitioned in at most 6 connected disjoint subgraphs, in this case.

Here we provide a supervisory algorithm for a simple case of connectivity maintenance problem, that will be modified for a realistic case in the following subsection. Based on Lemmas 2 and 3, our local-global connectivity preservation algorithm is shown in Algorithm 2, for robots with dynamics described as follows:

$$\dot{x}_i(t) = u_i(t)$$

(5)

where $x_i(t) \in \mathbb{R}^n$ is the position of the robot $i$, and $u_i(t) \in \mathbb{R}^n$ is the actuator input.

Algorithm 2 has to be implemented on all robots. It could be considered as a decentralized supervisory controller, which cancels any attempt toward disconnection of the network.

**Definition of the functions and variables used in this algorithm are as follows:**

- $x_i(t)$ is the position of the robots in the neighborhood of robot $i$.
- $\text{Edge\_Disconnection}(x(t), R)$ output is a set containing all the edges associated with $v_i$, which will be disconnected with any further movement (i.e., $v_i \in N_i \cup \{x_j \mid \|x_i - x_j\| = R\}$).
- $\text{Select\_Sector}(v_i)$ output is a set containing only one node in each connected sub-graph of the input argument, as described in Lemma 3.
- $\text{Vert}(v_i)$ output is the set of vertices on sides of the set of edges given as the input.
- $\text{Discover}([v_i, v_j], D_{E_i}, k, R)$ initiates a $k$-hop routing search to find a path, disjoint from $D_{E_i}$, between $v_i$ and $v_j$, by means of Algorithm 1.
- $\text{Response}(\cdot)$ is returning the result of its input argument, which is the result of the route discovery, which is done by Algorithm 1.
- $\text{Freeze}(v_i)$ is to stop robot $i$ (i.e., $u_i(t) = 0$).

**Remark 1.** In this algorithm, communication and computation delays are not considered, and the dynamic model is not realistic, because the input works on the velocity of the robot, which requires unbounded acceleration.

**Theorem 2.** A robotic network composed of $N$ robots whose dynamics are given as in (5) and whose distance dependent adjacency matrix is defined as in (2), will preserve connectivity under the supervisory controller given in Algorithm 2.

**Proof.** Considering the dynamics of the robots and the communication model, the motion of robot $i$ can cause disconnection of an edge connecting with robot $j$ only if $\|x_j - x_i\| = R$. Edges with this characteristic can then be stored in set $D_{E_i}$ for each robot $i$. Then, the existence of a $k$-hop alternative path disjoint from $D_{E_i}$ is tested (considering the results in Lemma 3 regarding the reduction of the number of necessary searches). If $k$-hop alternative routes are found for each edge, then $u_i(t)$ is allowed, otherwise $v_i$ will be frozen and the edges will not be disconnected. Therefore, an edge is disconnected only if there are other disjoint (from disconnecting edges) paths connecting the vertices with disconnecting edges and, according to Lemma 2, this is sufficient for connectivity preservation. □

**Corollary 2.1.** The results of Theorem 2 imply global connectivity preservation if $k = N - 1$.

**Proof.** Consider the minimum spanning tree of the graph excluded from the disconnecting edges. This tree will have $N - 1$ edges, if it exists. Therefore, if the graph stays connected after an edge disconnection, then there will be a path with length less than $N - 1$ between each vertices. Therefore, searching for a $(N - 1)$-hop alternative path is sufficient for global connectivity. □

**5.3. Local-global connectivity maintenance method with realistic considerations**

The previous subsection presented a basic version of the proposed method. However, it is not preserving connectivity in the presence of communication delay, which does exist in real world communication links. Nevertheless, it will be shown, that in the expense of shortening the communication range, achieving connectivity maintenance in the presence of communication delay is possible.

We will hereafter consider the following realistic robotic dynamic, which is more realistic than (5), with $M_i(x_i)$ as a positive
definite matrix representing the inertia of robot $i$ and $f(x_i, x_j, d_i, u_i)$ as a function of states of the robot where $d_i$, $u_i$ are exogenous inputs:

$$M_i(x_i)\dot{x}_i = f_i(x_i, x_j, d_i, u_i)$$

(6)

We have made the following assumptions:

$$\exists u_i(t) \parallel u_i(t) < \beta \parallel x_i(t) - Line(x_i(t_0), x_{i_0})$$

$$< \delta \implies \lim_{t \to \infty} x_i(t) = x_{i_0}$$

(7)

$$\parallel \dot{x}_i \parallel < S_{max}$$

(8)

which are a feasibility condition for controller design and a maximum speed consideration for robots (i.e., a maximum attainable speed by bounded inputs). With $x_{i_0}$ as the desired location for robot $i$ and $\delta$ as the maximum distance from the line connecting the initial location and the desired location, while controlling the robot to move on this line. It is worth noting that the condition $\exists u_i(t) \parallel u_i(t) < \beta \parallel x_i(t) - Line(x_i(t_0), x_{i_0})$$ \parallel < \delta$ is necessary for any kind of controller which maintains connectivity. Otherwise, a robot with only one link to the network could be detached regardless of the controller’s bounded action. The set $Line(x_1, x_2)$ is $\{x \in \mathbb{R}^2 \mid x = (1-\alpha)x_1 + \alpha x_2, \forall 0 \geq \alpha \leq 1\}$ and the distance of a point with a set is defined as the minimum distance of that point with a set in the plane.

The controllability notion $\lim_{t \to +\infty} x_i(t) = x_{i_0}$ represents the ability to asymptotically drive the robot to a desired position.

With this defined dynamic model for the robots and the existence of communication delay. It is not possible to stop robots from moving at the exact time a disconnection of the network is going to happen. Therefore, the connectivity maintenance algorithm has to evaluate the safety of an edge disconnection and command the robots, when the inter-distance between robots is a certain amount less than $R$. Consider an asymmetric communication topology in which request for discovering a $k$-hop connectivity can only propagate through the links with limited range less than $R_e$ defined in (9), with adjacency matrix for discovery propagation as (10), and the adjacency matrix for response as defined in (11).

$$R_i = R - (2k + 4)S_{max} + 2\delta$$

(9)

where $\tau_{max}$ is the maximum time delay with a 1-hop communication. Note that with this definition of $R_i$, implicitly we have assumed that $2(k + 4)S_{max} + 2\delta < R$. Where $\delta$ is the communication delay from the closest robot to the robot which has to be communicated with. $R_i$ represents as a maximum range where communication is possible, and $\delta$ represents the time delay. In this case, a procedure like Algorithm 2 with $R_i$ as the maximum communication range can maintain connectivity. However, the asymmetric definition of the adjacency matrix might freeze the whole system with an inter-distance $R_i$. Where $R_i < R - 2\delta$, any $k$-hop routing discovery will not propagate, while the network is still connected. This phenomenon also occurs for Algorithm 2 with inter-distance $R_i$. Therefore, another procedure should be added to unfreeze the network simultaneously. Consider a robot with at least one link with length more than $R_i$ that did not get announcement of a found alternative path after a specific time. Let $X_i(t)$ be the set $\{v_i \in \text{Vert}(D_k(t)) \mid \text{Disjoint path}(v_i, v_j) \}$, where $D_k(t) = \{v_i \mid v_i \in \text{Vert}(N_i) \}$ and $\text{Disjoint path}(v_i, v_j)$ is a path between $v_i$ and $v_j$ disjointed from the set of disconnecting edges $D_k(t)$. Any robot should move in a direction which will result in decreasing the distance between it and its neighbors in $N_i(t)$. In general this problem could be solved as the following constraint problem, where $t_{f_j}$ is the moments when robot $i$ get a Freeze2 command. All solution to this problem $x_{u_i}$ are formulated in Equation (12). The Freeze2 command changes the reference set point of robot $i$ to the position it had occupied at the moment $t_f$. The time $t_{u_i}$ is the first moment robot $i$ is unfrozen after some interval of being frozen.

$$x_{u_i}(t) = \{p \mid \max_{j \in N_i} \parallel p - x_j(t_{f_j}) \parallel > \parallel x_i(t_{f_i}) - x_j(t_{f_j}) \parallel < 0\}$$

(12)

The set $x_{u_i}(t)$ includes all the points $p$, such that: if robot $i$ goes to point $p$ its distance with each of its neighbors will be reduced. Note that $x_i(t_{f_i})$ is not accessible at the moment $t_{f_i}$ because of the communication delay. Also, $t_{f_i} \geq t_{f_j}$ if $j \in N_i$, because at the moment $t_{f_i}$ robot $j$ might be already in a freeze state, otherwise it will get a Freeze2 command at the moment $t_{f_i}$ because robot $i$ has an edge with it, which is disconnecting without a disjointed alternative path from other disconnecting edges. Therefore, $x_{u_i}(t)$ is defined from $t_{f_i} + \tau_{max}$. Consider each robot pursue the following target:

$$x_{u_i}(t) = \begin{cases} x_i(t_{f_i}) & x_u(t) = \emptyset \\ \emptyset & t \in [t_{f_i} + \tau_{max}, t_{u_i}] \end{cases}$$

(13)

Theorem 3. If each robot follows the reference in (13), eventually $X_i$ will be empty.

Proof. Consider node $v_i$ with $X_i \neq \emptyset$. The constraint problem in (12) will always have a solution if all the nodes $v_j \in N_i$ connected to $v_i$ are inside a sphere sector with opening angle less than $\pi$ in a sphere with radius more than $R_i$ and center $v_i$ as shown in Fig. 2. Consider the right circular base of the sector sphere, which is closer to the vertex of the sector sphere than any robot in $N_i$. The center of this circular base is one of the solutions to the problem (12): to show this, consider each plane consisting $v_j \in N_i$, $v_i$ itself and the center of the circular base as shown in Fig. 2, then:

$$\parallel x_i(t_{f_i}) - x_j(t_{f_i}) \parallel^2 = \parallel p - x_j(t_{f_j}) \parallel^2 + \gamma^2 - 2 \parallel p - x_j(t_{f_j}) \parallel \gamma \cos \theta$$

(14)

the base circle is perpendicular to the sphere sector, $\pi/2 \leq \theta \leq \pi$ and $\gamma > 0$. Therefore, for any $j$:

$$\parallel x_i(t_{f_i}) - x_j(t_{f_i}) \parallel^2 > \parallel p - x_j(t_{f_j}) \parallel^2$$

(15)

Now consider the smallest convex surface $C(t)$ enveloping all the nodes. This convex surface will be a polyhedron and, because of convexity, any vertex $v_i$ of this polyhedron and its
neighbors in \( N_t \) are inside a sphere sector with opening angle strictly less than \( \pi \). Consider:

\[
\sum_{i \in (t) \cap j \in N_t(t)} \|x_i(t) - x_j(t)\| \quad \tag{16}
\]

as the sum of all inter distances between each pair of vertices on the surface \( C(t) \), with their neighbors in \( N_t(t) \). Let the vertices on the polygonal track the set points driven by equation (13), then the energy like function (16) will eventually decrease. To show the decreasing behavior of (16), while tracking (13), let \( t_f = \max_t t_f \).

If the set of vertices which are on the largest enveloping surface remains unchanged, then:

\[
\lim_{t \to \infty} \sum_{i \in (t) \cap j \in N_t(t)} \|x_i(t_f + T) - x_j(t_f + T)\|
\]

\[
- \sum_{i \in (t) \cap j \in N_t(t)} \|x_i(t_f) - x_j(t_f)\| 
\]

With the assumption made in (7):

\[
\lim_{t \to \infty} x(t_f + T) = \rho \in X_0(t).
\]

So:

\[
\lim_{t \to \infty} \sum_{i \in (t) \cap j \in N_t(t)} \|x_i(t_f + T) - x_j(t_f + T)\|
\]

\[
= \sum_{i \in (t) \cap j \in N_t(t)} \|\rho - x_j(t_f)\| 
\]

Also, \( \lim_{t \to \infty} x(t_f + T) \subseteq x(t_0) \) for \( v_c \in C(t) \), because:

\[
\lim_{t \to \infty} \|x(t_f + T) - x(t_f)\| = \|\rho - x(t_f)\| < \|x(t_f) - x(t_f)\|
\]

Therefore:

\[
\sum_{i \in (t) \cap j \in N_t(t)} \|\rho - x_j(t_f)\| - \sum_{i \in (t) \cap j \in N_t(t)} \|x_i(t_f) - x_j(t_f)\|
\]

\[
\leq \sum_{i \in (t) \cap j \in N_t(t)} \|\rho - x_j(t_f)\| - \|x_i(t_f) - x_j(t_f)\| < 0
\]

Otherwise, if a new robot appears to be on the enveloping polyhe-
dral in a finite time, then we can apply the above argument for this new envelope. Therefore, the smallest enveloping surface is eventu-
ally shrinking. Consequently, there will be no frozen edge with
distance larger than \( R_0 \).

Remark 2. The proof of Theorem 3 has not considered free move-
ment for unfreeze robots. Therefore, our proof is correct only if
there is not any movement except for that caused by (13). How-
ever, in the case where there is free movement and a new robot
might get a freeze command, the number of frozen robots will oscil-
late. Hence, movements ruled by (13) will unfreeze some of frozen robots and free movements might add some new frozen robots.

Algorithm 3 is our proposed method to maintain connectiv-
ity in the presence of communication delay and dynamics (6). The
algorithm is event driven and each event subroutine is executed when the event occurs. The \(< unfreeze >\) event will occur at time \( t_{f_1} \), if \( N_t \) is inside a sphere sector with opening angle less than \( \pi \), \( X_t(t_{f_1} - \epsilon) \neq \emptyset \) and \( X_t(t_{f_1}) = \emptyset \), with \( \epsilon > 0 \& \epsilon \to 0 \). On \(< unfreeze >\) events, the reference input is switched back to the Path_Planner.

On \(< freeze >\) event, the axis of the sphere sector is calcu-
lated, and a point on this axis is selected as a reference input for the controller, such that all the robots in \( X_t \) are farther from the circu-
lar base crossing this point. \( Wait(t_{\text{max}}) \) is to wait \( t_{\text{max}} \) seconds, so that the robot can access its neighbors information at their freeze

time, because the freeze event occurs at \( t_{f_1} \) but information from neighbors are accessible after a 1-hop delay. The \( \cdot \) operator is the dot product of two vectors. Any time a discovery message is sent, a new instance of a \( \text{Timer}_{\text{event}}(\ldots) \) will be created, which is trig-
gered by an acknowledgment of discovery or after the dead time (specified in its second argument) is elapsed. \text{Ack} is to specify ac-
knowledgement of a discovery message for an alternative path by
the destination (i.e., the alternative path has been found). Here, the \( \text{Freeze}_2(v) \) command changes the reference set point of the robot to
the position it occupied at the moment \( t_{f_1} \) (i.e., the moment the \( \text{Freeze}_2 \) command is executed). Functions used in this algorithm are
similar to Algorithm 2, with the difference of passing \( R_0 \) as the lim-
ited range of communication distance and \( x(t - \tau)N_t \) is the delayed
location of neighbor nodes.

An overall view of our approach and its position in the control
loop is shown in Fig. 3.

Theorem 4. For an initially connected (based on adjacency matrix
defined in (10)) graph, the multi-robot system with communication
graph defined in (2), with link delay less than \( t_{\text{max}} \) and with the
maximum speed and controller existence assumptions (7) and (8), will
stay connected by execution of Algorithm 3 on all robots, for any \( k \) such that \((2k + 4)S_{\text{max}}t_{\text{max}} < 2\epsilon < R_0\).

Proof. After \((k + 1)t_{\text{max}} \) seconds the discovery of the alternative
k-length path will be evaluated and, in this time interval, in the worst case (i.e., robots on each side of a link moving in opposite

Please cite this article as: K. Khateri, M. Pourgholi and M. Montazeri et al., Decentralized local-global connectivity maintenance
for networked robotic teams, European Journal of Control, https://doi.org/10.1016/j.ejcon.2019.06.008
6. Discussion and simulation results

Based on robots' dynamic model involved in the networked multi-robot system, the probability of a controller design to satisfy the assumptions made in (7) can be solved with different control laws, where the probability of an alternative path is assumed to be (22) for all robots' radius. However, developing such a procedure is beyond the scope of this work and it is suggested for a future work.

In this section, we introduce a sliding mode controller for double integrator robots subject to uncertainty, where the needed assumptions are fulfilled.

Fig. 3. Overall view of local global algorithm in the control loop.

Please cite this article as: K. Khayesi, M. Pourboli, and M. Montazeri et al., Decentralized local global connectivity maintenance for networked robotic systems, European Journal of Control, https://doi.org/10.1016/j.ejcon.2019.06.008
f(x(t), x̄(t), d(t)) for any ∥x∥ ≥ Smax.

\[ x_i(t) = -\xi x_i(t) + u_i(t) + f(x_i(t), x̄_i(t), d_i(t), t) \]  

(22)

\[ ̇x_i(t) = -\xi ̇x_i(t) - \beta \text{Sat}(\psi( ̇x_i(t) + x_i(t))) + f(x_i(t), x̄_i(t), d_i(t), t) \]  

(23)

Sat(F) = [sat(f_1) \ldots sat(f_n)]^T, \text{sat}(f_i) = \begin{cases} 1 & f_i > 1 \\ -1 & f_i < -1 \\ \frac{f_i}{|f_i|} & |f_i| \leq 1 \end{cases} \]  

(24)

Simulation results for 20 robots with dynamic model (22) on an initially complete graph are illustrated in Fig. 4. The goal is to organize the robots on the largest possible circular shape, while connectivity is maintained. Robots are initially in the same place and the path planner tries to unboundedly increase the diameter of the circle. In this experiment, the subroutine for “on event unfreeze” in Algorithm 3 is not executed, because we want the robots to freeze, when the largest possible circle happens. In this experiment, different values for the locality level of the algorithm are chosen. Parameter values used for this simulation are shown in column ex-1 of Table 1. It is shown that searching for a longer alternative k-length path will result in a larger coverage for the multi-robot system. However, this result is due to the small communication delay compared to the number of robots and their maximum speed. In conditions where the robots are very fast or the communication delay is very long, locality of the algorithm is not proportional to the maximum attainable coverage. Fig. 5 shows results for the same goal with Smax = 9 m/s and other parameters are equal to the experiment ex-1, as shown in ex-2 column of Table 1. Note that, in this experiment, maximum coverage for k = 5 is larger than the result for k = 19.

In another scenario, one of the robots is considered to be unable to move in the first 120 seconds, because of an obstacle or a fault, while the goal of all the robots is to reach to the point (200,200) in the Cartesian plane, from the origin. Due to the connectivity maintenance restrictions, the moving group should wait, until the unable to move robot’s fault is resolved. In Fig. 6a and b, the movement of robots in each Cartesian axis of R² is illustrated. The motion of the robots during this simulation is also shown in the attached video, which is 30 times faster than the real time. Dynamic models of the robots are considered to be as in (22), with f(x_i(t), x̄_i(t), d_i(t)) = r, where r is a random vector in R² and ∥r∥∞ < μ, considered as an additive disturbance. Also, number of
edges of the communication network and the algebraic connectivity are depicted in Fig. 6c and d, respectively. The oscillating movement of the robots, while waiting for the faulty robot, is due to the events freeze and unfreeze subroutines of Algorithm 3, such that the network is unfreezed by this subroutine (i.e., normal robots turn back toward the faulty robot to reduce inter-distances more than $R_c$) and in another try to reach to the point (200,200) the network freezes again and this oscillation continues until the faulty robot recovers from the fault. Parameters used for this simulation are shown in column ex-3 of Table 1.

6.1. Bandwidth requirement

To the best of the authors knowledge, all the existing connectivity maintenance procedures like [1,5,11,20,28,31] are assuming a continuous time availability of neighbors information, which needs infinite number of messages and infinite bandwidth. However, here we assume possibility of implementing a discrete versions of the existing methods. The proposed local-global connectivity maintenance needs information exchange for the alternative path search, when one of the edges are going to be disconnected. While, in normal working condition (i.e., there are no disconnecting edge), only distance between neighbors are checked. On the other side, traditional global connectivity maintenance methods are always information demanding. The number of message exchanges for different methods are as follows:

- With local connectivity maintenance method: each robot needs to send information to immediate neighbors. Therefore, $N$ messages are broadcast to neighbours.
- The needed assumption for power iteration based method to work correctly is to have a bounded estimation error [28]. To achieve a bounded error less than some bound $c$, $O(z_j) = \frac{\text{dist}(c)}{\lambda_j}$ number of iteration is needed, as shown in Section 5.1.

Each iteration needs data exchange between neighbors: hence, similar to the local method, each iteration needs $O(N)$ messages. Therefore, power iteration based method needs data exchange of $O(z_jN)$. It is worth noting, that $z_j$ is not a fixed value and is dependent on the ratio of $\lambda_2$ to $\lambda_3$ and when $\lambda_2 = \lambda_3$ it needs infinite data exchanges.

- In the proposed method, there are maximally $N^2$ edges for an $N$ robots network, and each robot propagates the alternative path discovery message, for a disconnecting edge, only once, and ignores the message if it had registered its id in the discovery message another time, according to Algorithm 1. Thus, for each disconnecting edge, at most $N$ messages will be broadcast to the neighbors. In the worst case of an all to all graph with all the edges being disconnecting, there will be $O(N^3)$ messages needed for the proposed method. Fig. 7 shows number of messages which are exchanged in ex-3.

7. Conclusion

In contrast with the belief of achieving more flexibility with a global solution for connectivity maintenance, this paper showed that, in the presence of communication delay, global approaches to connectivity maintenance are restricting the maximum achievable speed for the robots. Therefore, the paper proposed an algorithm that allows to freely tune the level of locality. The connectivity maintenance problem was divided into designing a controller to cope with the dynamics of the robots for satisfying some assumptions and a general algorithm to cope with the connectivity which, from a practical point of view, makes the system design modular (i.e., the designer of the multi-robot system only need to consider the controller part for different types of robots).

With connectivity maintenance as the highest level supervisor, a single permanent movement failure for one of the robots will limit the work-space of the whole robotic system permanently. Studying a higher level decentralized algorithm for disabling the connectivity maintenance algorithm in such situation is recommended for future study on this subject.

Fig. 6. Group displacement from origin to (200,200) with one of robots unable to move in the first 120 s.

Fig. 7. Number of messages.
Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejcon.2019.06.008.

References

[1] A. Ajourou, A. Momeni, A.G. Aghdam, A class of bounded distributed control strategies for connectivity preservation in multi-agent systems, IEEE Trans. Autom. Control 55 (12) (2010) 2828–2833.

[2] R. Aragues, G. Shi, D.V. Dimarogonas, C. Sagüés, K.H. Johansson, Y. Mezouar, Distributed algebraic connectivity estimation for undirected graphs with upper and lower bounds, Automatica 50 (12) (2014) 3253–3259.

[3] V.A. Battagel, C.H. Ribeiro, Analysis of the effects of communication delay in the distributed global connectivity maintenance of a multi-robot system, in: Complex Networks VI, Springer, 2015, pp. 149–157.

[4] N.M.M. De Abreu, Old and new results on algebraic connectivity of graphs, Linear Algebra Appl. 423 (1) (2007) 53–73.

[5] D.V. Dimarogonas, K.H. Johansson, Decentralized connectivity maintenance in mobile networks with bounded inputs, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2007, pp. 1507–1512.

[6] Y. Dong, J. Huang, Flocking with connectivity preservation of multiple double integrator systems subject to external disturbances by a distributed control law, Automatica 55 (2015) 197–203.

[7] Z. Feng, C. Sun, G. Hu, Robust connectivity preserving rendezvous of multirobot systems under unknown dynamics and disturbances, IEEE Trans. Control Netw. Syst. 4 (4) (2017) 725–735.

[8] M. Fiedler, Algebraic connectivity of graphs, Czechoslov. Math. J. 23 (2) (1973) 295–305.

[9] E.F. Flushing, M. Kudelski, L.M. Gambardella, G.A. Di Caro, Connectivity-aware planning of search and rescue missions, in: Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), IEEE, 2013, pp. 1–8.

[10] M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, Decentralized estimation of Laplacian eigenvalues in multi-agent systems, Automatica 49 (4) (2013) 1031–1036.

[11] A. Gasparri, A. Leccese, L. Sabattini, G. Ulivi, Collective control objective and connectivity preservation for multi-robot systems with bounded input, in: Proceedings of the American Control Conference (ACC), IEEE, 2014, pp. 813–818.

[12] A. Gasparri, L. Sabattini, G. Ulivi, Bounded control law for global connectivity maintenance in cooperative multirobot systems, IEEE Trans. Robot. 33 (3) (2017) 760–777.

[13] S. Hayat, E. Yannaz, T.X. Brown, C. Bettstetter, Multi-objective UAV path planning for search and rescue, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 5569–5574.

[14] M.A. Hsieh, A. Cowley, V. Kumar, C.J. Taylor, Maintaining network connectivity and performance inrobot teams, J. Field Robot. 25 (1–2) (2008) 111–131.

[15] P. Jacquet, P. Multiethaler, T. Clausen, A. Lautoisi, A. Quayum, L. Vicenzi, Optimized link state routing protocol for ad hoc networks, in: Proceedings of the IEEE International Multi Topic Conference. IEEE INMIC 2001, Technology for the 21st Century, IEEE, 2001, pp. 62–68.

[16] F. Janati, E. Abdollahi, S.S. Ghidary, M. Jannatfar, J. Baltes, S. Sadeghejadi, Multi-robot task allocation using clustering method, in: Robot Intelligence Technology and Applications 4, Springer, 2017, pp. 233–247.

[17] M. Ji, M. Egerstedt, Distributed coordination control of multiagent systems while preserving connectedness, IEEE Trans. Robot. 23 (4) (2007) 693–703.

[18] Z.-P. Jiang, Y. Wang, Input-to-state stability for discrete-time nonlinear systems, Automatica 37 (6) (2001) 857–869.

[19] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: Mobile Computing, Springer, 1996, pp. 153–181.

[20] Z. Kan, L. Navarawong, J.M. Shea, E.L. Pasiliao, W.E. Dixon, Graph matching-based formation reconfiguration of networked agents with connectivity maintenance, IEEE Trans. Control Netw. Syst. 2 (1) (2015) 24–35.

[21] N. Karmiinejad, S.A. Taher, M. Shahidepour, K. Khateri, A hierarchical governor/turbine and electric vehicles optimal control framework for primary frequency support in power systems, IEEE Trans. Smart Grid 9 (6) (2018) 6702–6712.

[22] K. Khateri, M. Pourgholi, M. Montazeri, L. Sabattini, A comparison between decentralized local and global methods for connectivity maintenance of multi-robot networks, IEEE Robot. Autom. Lett. 4 (2) (2019) 633–640.

[23] X. Li, D. Sun, J. Yang, A bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles, Automatica 49 (1) (2013) 285–292.

[24] D.Q. Mayne, M.M. Seron, S. Rakovic, Robust model predictive control of constrained linear systems with bounded disturbances, Automatica 41 (2) (2005) 219–224.

[25] H.-N. Nguyen, S. Olaru, P.O. Gutman, M. Hovd, Constrained control of uncertain, time-varying linear discrete-time systems subject to bounded disturbances, IEEE Trans. Autom. Control 60 (3) (2015) 831–836.

[26] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE 95 (1) (2007) 215–233.

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

[28] L. Sabattini, N. Chopra, C. Secchi, On decentralized connectivity maintenance for mobile robotic systems, in: Proceedings of the Fiftieth IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), IEEE, 2011, pp. 988–993.

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

[30] L. Sabattini, C. Secchi, N. Chopra, Decentralized estimation and control for preserving the strong connectivity of directed graphs, IEEE Trans. Cybern. 45 (10) (2015) 2273–2286.

[31] L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, Distributed control of multirobot systems with global connectivity maintenance, IEEE Trans. Robot. 29 (5) (2013) 1326–1332.

[32] C. Secchi, L. Sabattini, C. Fantuzzi, Decentralized global connectivity maintenance for interconnected homogeneous systems with communication delays, IFAC Proc. Vol. 45 (19) (2012) 78–83.

[33] A. Swidan, H.B. Abdelghany, R. Safian, Z. Zilic, Mobility and direction aware ad-hoc on demand distance vector routing protocol, Proc. Comput. Sci. 94 (2016) 49–56.

[34] G.A. Wolkar, R.C. Biradar, A survey on hybrid routing mechanisms in mobile ad hoc networks, J. Netw. Comput. Appl. 77 (2017) 48–63.

[35] Q. Wang, H. Fang, J. Chen, Y. Mao, L. Dou, Flocking with obstacle avoidance and connectivity maintenance in multi-agent systems, in: Proceedings of the IEEE Fifty First Annual Conference on Decision and Control (CDC), IEEE, 2012, pp. 4009–4014.

[36] G. Wen, Z. Duan, H. Su, G. Chen, W. Yu, A connectivity-preserving flocking algorithm for nonlinear multi-agent systems with bounded potential function, in: Proceedings of the Thirteenth Chinese Control Conference (CCC), IEEE, 2011, pp. 6018–6024.

[37] G. Wen, Z. Duan, H. Su, G. Chen, W. Yu, A connectivity-preserving flocking algorithm for multi-agent dynamical systems with bounded potential function, IET Control Theory Appl. 6 (6) (2012) 813–821.

[38] G. Wen, H.-T. Zhang, W. Yu, Z. Zuo, Y. Zhao, Coordination tracking of multi-agent dynamical systems with general linear node dynamics, Int. J. Robust Nonlinear Control 27 (9) (2017) 1526–1546.

[39] F. Yang, R.A. Freeman, G.J. Gordon, K.M. Lynch, S.S. Srinivasa, R. Sukthankar, Decentralized estimation and control of graph connectivity for mobile sensor networks, Automatica 46 (2) (2010) 390–396.

[40] Y. Yang, Y. Shi, D. Constantinescu, Connectivity-preserving coordination control of multi-agent systems with time-varying delays, arXiv:arXiv:1803.08152 (2018).

Please cite this article as: K. Khateri, M. Pourgholi and M. Montazeri et al., Decentralized local-global connectivity maintenance for networked robotic teams, European Journal of Control, https://doi.org/10.1016/j.ejcon.2019.06.008.