Retaining Connectivity In Mobile Communication Mesh Networks

Alessandro Pisano * **Mauro Franceschelli** * **Alessandro Pilloni** *
Yuri Shtessel ** **Elio Usai** *

* *Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy (e-mail: pisano@diee.unica.it).*
** *Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, Huntsville, Alabama (US) (e-mail: shtessy@uah.edu)*

Abstract: This paper deals with the problem of retaining the connectivity in a Mobile Ad-hoc communication mesh Network (MANET). A multi-agent systems perspective is taken, where primary mobile agents (PAs) can only communicate when the relative distance is less than a "visibility range". PAs form a network that can become disconnected depending on how they move to achieve their global task (which does not include the connectivity maintenance). To retain connectivity, a number of Relay Agents (RAs), whose motion is governed by a command center (CC), are sent to the field to act as "communication bridges" enforcing the global connectivity of the network containing both the PAs and the RAs. Graph-oriented concepts and analysis tools, particularly the minimal spanning tree (MST) notion, are adopted in the present work to analyze the connectivity properties of the network and to establish in real time how many additional RAs are required and how they should move in order to prevent the connectivity loss. Artificial potential fields and finite-time control techniques are utilized to drive the relay agents to their waypoints while avoiding the collisions. Numerical examples confirm the efficacy of the proposed multi-layer control strategy.

Keywords: Multi-agent systems, Mobile networks, Connectivity maintenance

1. INTRODUCTION

Mobile Ad-hoc communication mesh NETworks (MANETs) that consist of multiple agents and perform cooperative tasks have many military and civilian applications (Bordetsky et al. (2010)), e.g. coping with crisis situations that arise due to natural or man-made disasters (Kanchanasut et al. (2007)). Reconnaissance, surveillance, and sensing through multi-robot networks have been the subject of much research (see, e.g., McLain et al. (2003); Olfati-Saber et al. (2007); Ren et al. (2007)). Within a related framework, recent results on finite-time consensus-based distributed coordination of uncertain multi-agent systems are also worth to remark (seeFranceschelli et al. (2016, 2015))).

In typical sensing scenarios, for instance, a group of agents investigate a phenomenon of interest, with individual agents performing a primary data-gathering task while also meeting other requirements for network coordination. In such networks, each agent gathers its data and ensures network coordination by adapting to meet the system's goal. The research has based heavily on ideas from control, sensor networks, optimization, and graph theory (Jadbabaie et al. (2003); Mesbahi et al. (2010)), where information exchange and cooperation are key elements.

Loss of communication capability (i.e., loss of connectivity) in the mobile communication mesh networks could result in loss of functionality of the entire application. Predicting and preventing loss of connectivity in the MANETs is a challenging task, which is usually performed using some form of traditional graph theoretic methods (Zavlanos et al. (2011)). Research in the robotics field on MANETs has also provided solutions to building and maintaining a communications bridge of mobile robotic routers (Tian et al. (2005)) to connect a mobile agent to a static base station.

In the majority of the existing approaches (see e.g. Zavlanos et al. (2011) and the references therein) all PAs can move to achieve the task of connectivity maintenance. Here, the motion of the PAs is not at CC disposal, and in order to retain connectivity CC is only in a position of sending the appropriate number of RAs to the field of operation. In Edwards et al. (2013), inspired by Zavlanos et al. (2011); Simonetto et al. (2011), all available RAs are in play and their way-points were computed through the solution of an optimization problem attempting to maximize the Fiedler value of a suitable "predicted" graph, that was called "phantom" graph. This approach brings, however, some drawbacks. The iterative solver could be unable to rapidly find a viable solution if the initial solution is not properly chosen. Additionally, the objective function is not available in analytic form, and thus its gradient has to be calculated numerically, which brings additional errors.

Recent research (see e.g. Misra et al. (2008); Holleran et al. (2010); Shtessel et al. (2012)) has recognized that an agent/node's primary task (for instance, reconnaissance) may potentially conflict with the MANET connectivity requirement. For the agent/node to perform its primary task successfully, it may have to move in such a way as to violate a movement constraint that keeps the MANET connected. In the other words, the primary agents/nodes of MANETs may not cooperate on retaining the MANET connectivity. At any instant of time, this situation can be interpreted and analyzed in terms of a graph, in which the primary mobile agents (PAs) are the nodes and the capability to communicate with other agents constitutes the edges of the graph.

This paper is dedicated to addressing the task of retaining the connectivity in such "conflicting' MANETs. To address this challenge, in this work it is proposed a multi-layered control architecture addressing the following tasks:

1. Predict loss of connectivity using concepts of "phantom graph" and minimal spanning tree. The graphs that are based on predicted positions of PAs are referred to as "phantom graphs" as they represent predicted future scenarios rather than actual ones. Specifically, in order to build the phantom graph, finite-time converging sliding-mode observers are used to estimate the velocities of PAs, then direct integration of kinematic equations of PAs is performed to predict the future trajectory of the PAs.

2. Identify isolated node(s) on the phantom graph. Specifically, the algorithm for identification of isolated nodes is obtained using the concept of the minimum spanning tree (Zavlanos et al. (2011)).

3. Compute and drop waypoints (that yield a connected "phantom' graph), where the additional nodes/relay agents (RAs) should be placed by certain time on the prescribed locations in order to retain the connectivity.

4. Compute feasible paths for the agents to follow. The design algorithm for the paths that connect the RAs with the waypoints while providing the collision avoidance with the PAs is proposed. Specifically artificial potential field and finite-time convergent control are the key ingredients of this item.

Addressing these tasks allows retaining the network connectivity via implementing self-forming and self-healing structure of the network.

The paper has the following structure. Section II deals with preliminary statements and definitions. Section III states the problem under investigation and the underlying assumptions. Section IV illustrates the general structure of the two-level proposed architecture, whereas the Sections V and VI explain in detail the upper and lower levels. Section VII formalizes the main result of the present paper, Section VIII gives some comments and Section IX presents numerical simulation results corroborating the theoretical properties.

2. PRELIMINARIES

2.1 Euclidean graphs

To model the connectivity properties of the mobile network, a graph-theoretic approach is taken. Denote as $q_i(t) \in \mathbb{R}^2$ the position vectors of the RAs, with $i \in V$ where $V = \{1, 2, ..., N\}$ denotes the set of nodes indexed by the set of mobile agents. (i, j) denote a communication link between the i-th and j-th RAs, whose position vectors are $q_i(t), q_j(t) \in \mathbb{R}^2$. With every communication link, we associate a weight function

$$w : \mathbb{R}^2 \times \mathbb{R}^2 \to \{0, 1\}$$

such that

$$w_{ij}(t) = w(q_i(t), q_j(t)) = \begin{cases} 1 \ if \ d(q_i(t), q_j(t)) \le \beta \\ 0 \qquad otherwise \end{cases}, \quad (1)$$

where

$$d(q_i(t), q_j(t)) = \|q_i(t) - q_j(t)\|_2 \quad (2)$$

is the euclidean distance between the i-th and j-th RAs and $\beta > 0$ is the *visibility range.*

A weighted time-varying Euclidean Graph (EG) $\mathcal{G}(t) = (V, W(i, j, t))$ is defined (see Zavlanos et al. (2011)) where $W : V \times V \times \mathbb{R}^+ \to \{0, 1\}$ denotes the set of edge weights, such that

$$W(i, j, t) = w_{ij}(t) \quad (3)$$

for $i, j \in V$ and $w_{ij}(t)$ as in (1)-(2). So defined, the graph $\mathcal{G}(t)$ turns out to have symmetric weights.

***Definition** 2.1.* The EG $\mathcal{G}(t) = (V, W(i, j, t))$ is said to be *connected* if there exists a path between any pair of nodes having positive edge weights (1).

Graph connectivity can be captured using the Laplacian matrix $\mathcal{L}(t) \in \mathbb{R}^{n_p \times n_p}$ of the graph $\mathcal{G}(t)$, which is defined by

$$[\mathcal{L}(t)]_{ij} = \begin{cases} -w_{ij}(t) & \text{if} \quad i \ne j \\ \sum_{s \in V/i} w_{is}(t) & \text{if} \quad i = j \end{cases} \quad (4)$$

where n_p is the number of PAs. The Laplacian matrix of a network $\mathcal{G}(t)$ with symmetric weights is always a symmetric positive-semidefinite matrix with spectral properties closely related to network connectivity, as it can be seen from the following theorem (see Godsil et al. (2001))

***Theorem** 2.2.* Let

$$0 = \lambda_1(\mathcal{L}(t)) \le \lambda_2(\mathcal{L}(t)) \le ... \le \lambda_{n_p}(\mathcal{L}(t))$$

be the ordered eigenvalues of the Laplacian matrix $\mathcal{L}(t)$. $\lambda_2(\mathcal{L}(t)) > 0$ if and only if $\mathcal{G}(t)$ is connected at time t.

2.2 Kinematics of primary and relay agents

Motion of the PAs and RAs is considered in the two-dimensional x-y plane. Let

$$p_i(t) = [p_{ix}(t), \ p_{iy}(t)]^T \quad i = 1, 2, ..., n_p$$

and

$$r_j(t) = [r_{jx}(t), \ r_{jy}(t)]^T \quad j = 1, 2, ..., n_r$$

denote the position of the i-th PA and j-th RA, respectively. A kinematic representation is given for the moving PAs and RAs. The motion of the $i - th$ PA is modeled by

$$\dot{p}_i(t) = [v^p_{ix}(t), \ v^p_{iy}(t)]^T \quad (5)$$

whereas motion of the $j-th$ RA is modeled by

$$\dot{r}_j(t) = [v^r_{jx}(t),\ v^r_{jy}(t)]^T \tag{6}$$

3. ASSUMPTIONS AND PROBLEM FORMULATION

Two types of agents operate in the 2-D field of operation with quite different agendas: a) so-called primary mission agents (PAs) and b) so-called relay agents (RAs).

The dynamic Euclidean graph $\mathcal{G}(t)$ of the studied MANET consists of moving PAs (nodes) that perform, for instance, reconnaissance tasks which may potentially conflict with the MANET connectivity requirement. The task of retaining connectivity of this dynamic graph is based on the use of the RAs employed in a special way and governed by a command center (CC). PAs and RAs have integrated wireless communication capabilities allowing information exchange with the other PAs and RAs, as well as the CC, provided that the relative distance does not exceed the visibility range β. The task of the present paper is to control the motion of the RAs to preserve the network connectivity where $v^r_{jx}(t)$ and $v^r_{jy}(t)$ serve as the control functions.

We are now ready to describe in general terms our proposed method. Let $t_k = kT$ $(k = 0, 1, 2...)$, then the receding horizon windows $\mathcal{T}_k = [t_k, t_{k+1}]$ are defined. At the initial instant t_k, prediction of the future motion of the PAs is made, and the phantom graph is created on the basis of the predicted PA positions at the final time instant t_{k+1}. If such phantom graph is found disconnected then the way points are to be computed so that, being taken by the RAs, make the phantom graph connected. Next, the assigned RAs are to be dispatched and arrive at these way points by the time instant t_{k+1} while avoiding possible collisions.

The next assumptions are made throughout.

***Assumption** 1.* PAs have norm-bounded velocity and acceleration respectively

$$\|\dot{p}_i(t)\| \le v_M, \tag{7}$$

$$\|\ddot{p}_i(t)\| \le a_M, \quad i \in V, \tag{8}$$

where v_M and a_M are a-priori known positive constants.

***Assumption** 2.* PAs and RAs locations are supposed to be continuously available to the CC.

***Assumption** 3.* The studied MANET covers a compact area $(x, y) \in \Omega(x, y)$, so that the physically available velocity $v_j(t) = [v^r_{jx}(t),\ v^r_{jy}(t)]^T$ of j^th RA, $1, 2, ..., n_r$, is sufficient for reaching any point $(x, y) \in \Omega(x, y)$ during a time interval $t \in [(k-1)T, kT]$.

4. CONTROL SYSTEM ARCHITECTURE

A two-level control system (see Fig. 1), whose goal is to preserve connectivity in the multi-agent mobile mesh communication network, is executed in the CC.

Specifically, the designated modules implement the following functionalities

M_1 : A module to predict, at $t = kT$ $(k = 0, 1, 2, ...)$, the PAs locations at the future time $t = (k+1)T$, and to build the phantom graph based on the predicted PAs locations.

Upper level	
M1	Motion prediction
M2	Connectivity analysis and way-points computation
M3	*RAs assignment*

Lower level	
M4	*Path planning and collision avoidance*
M5	*Path tracking via finite-time velocity control*

Fig. 1. A structure of two-level control system for preserving connectivity

M_2 : A module to check connectivity of the phantom graph, to predict the loss of connectivity, and to identify the disconnected subcomponents of the phantom graph. The module outputs the way-points to be dropped in the field of operation in order to retain the connectivity of the network.

M_3 : A module to assign each waypoints to a specific RA according to optimality requirements (e.g., minimal distance to be traveled).

The proposed lower-level algorithms contain the following modules:

M_4 : A module that generates paths that connect the initial positions of the RAs and the computed waypoints while avoiding collisions.

M_5 : A module that generates velocity profiles for the RAs to reach the computed waypoints along the prescribed path by time $t \le (k+1)T$

The precise contents of the modules of the proposed algorithm that solves the problem of retaining connectivity in the considered mobile communication network are disclosed in the rest of the paper.

5. THE UPPER LEVEL OF THE CONTROL SYSTEM

5.1 Module M_1: Motion prediction.

The algorithm that is described in this subsection is devoted to predicting, at $t = (k-1)T$, $k = 1, 2, ...,$ the position of the PAs at future time $t = kT$. This information is used for predicting a possible loss of network connectivity if any, to be repaired by dropping into the field the suitable number of RAs,

To facilitate such prediction the velocities v^p_{ix} and v^p_{iy} of the i-th PA are to be estimated for all $i = 1, 2, ..., n_p$.

Assume that the position $(p_{ix}(t), p_{iy}(t))$ of the i-th PA is measured on-line, then the PA's velocities are proposed to be estimated using the robust finite-time convergent super-twisting differentiators Levant (2005):

$$\begin{aligned} \dot{\xi}^x_{1i} &= -1.5\sqrt{L^x_i}|\xi^x_{1i} - p_{ix}|^{1/2} sign(\xi^x_{1i} - p_{ix}) + \xi^x_{2i} \\ \dot{\xi}^x_{2i} &= -1.1 L^x_i sign(\xi^x_{1i} - p_{ix}) \\ \hat{v}^p_{ix} &= \xi^x_{2i} \end{aligned} \tag{9}$$

$$\begin{aligned} \dot{\xi}^y_{1i} &= -1.5\sqrt{L^y_i}|\xi^y_{1i} - p_{iy}|^{1/2} sign(\xi^y_{1i} - p_{iy}) + \xi^y_{2i} \\ \dot{\xi}^y_{2i} &= -1.1 L^y_i sign(\xi^y_{1i} - p_{iy}) \\ \hat{v}^p_{iy} &= \xi^y_{2i} \end{aligned} \tag{10}$$

where $i = 1, 2, ..., n_p$, and L_i^x, L_i^y are constants such that

$$L_i^x \geq a_M, \quad L_i^y \geq a_M. \tag{11}$$

The main characteristic of the super-twisting differentiator is that it guarantees the finite time convergence property, i.e. the existence of a finite τ_1 such that

$$\xi_{2i}^x(t) = \dot{p}_{ix}(t), \quad \xi_{2i}^y(t) = \dot{p}_{iy}(t), \quad \forall t \geq \tau_1 \tag{12}$$

while also providing remarkable robustness against the measurement noise. With large enough parameters L_i^x and L_i^y convergence property (12) can be enforced with a maximal settling time $\tau_1 \ll T$.

The estimated velocities and measured position are used to predict the future PA positions on the designated time window:

$$\hat{p}_{ix}(kT) = p_{ix}((k-1)T) + T\hat{v}_{ix}^p((k-1)T), \quad k = 1, 2, ...$$
$$\hat{p}_{iy}(kT) = p_{iy}((k-1)T) + T\hat{v}_{iy}^p((k-1)T), \quad k = 1, 2, ... \tag{13}$$

The estimation formula (13) basically assumes that the PA velocities will keep constant to the estimated values at $t_{k-1} = (k-1)T$ in the whole time interval $\mathcal{T}_{k-1} = [(k-1)T, kT]$. Straightforwardly, by virtue of Assumption 1 the error in the estimated position $\hat{p}_i(kT)$ can be upper estimated for all $k = 1, 2, ...$ as

$$\|\hat{p}_i(kT) - p_i(kT)\|_2 \leq E_p = \min\left\{2v_M T, \frac{1}{2}a_M T^2\right\} \tag{14}$$

5.2 Module M_2: Connectivity analysis and way-points computation

At each $t_{k-1} = (k-1)T$, on the basis of the estimated PAs position vectors $\hat{p}_i(kT)$ delivered by the module M_1, the "phantom" graph $\hat{\mathcal{G}}(kT)$, whose nodes are associated with the predicted PAs positions, is constructed.

The approach is taken in the present work of exploiting the concepts of Euclidean minimum spanning tree of a graph to easliy find a viable solution to the problem of retaining connectivity in the MANET.

RA's way-point computation via Minimum Spanning Tree

***Definition** 5.1.* The minimum spanning tree is a spanning tree of a connected undirected graph which connects all the vertices together with the minimal total weighting for its edges.

The Euclidean minimum spanning tree (EMST) is particularly a minimum spanning tree of a set of agents in the plane (or more generally in n-dimensional space), where the weight of the edge between each pair of agents is the Euclidean distance between those two points. The EMST turns out to be particularly useful in the present scenario as a tool to compute the way points for the RAs to restore the connectivity of the phantom graph whenever it is predicted to be lost at $t = kT$.

The EMST of a graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{W}(i,j,t))$ will be denoted as

$$\mathcal{ST}(\mathcal{G}(t)) = \left(\mathcal{V}_{st}^{\mathcal{G}}, \mathcal{W}_{st}^{\mathcal{G}}(i,j,t)\right) \tag{15}$$

where $\mathcal{V}_{st}^{\mathcal{G}} = V$ denotes the corresponding set of nodes, and $\mathcal{W}_{st}^{\mathcal{G}}(i,j,t)$ the corresponding set of edge weights. The set of edges $\mathcal{E}_{st}^{\mathcal{G}}(t) \subseteq V \times V$ of graph $\mathcal{ST}(\mathcal{G}(t))$ is the set

$$\mathcal{E}_{st}^{\mathcal{G}}(t) = \{(i,j) \in (V \times V) : \mathcal{W}_{st}^{\mathcal{G}}(i,j,t) > 0\} \tag{16}$$

and its cardinality (i.e., the number of edges of $\mathcal{ST}(\mathcal{G}(t))$) is

$$|\mathcal{E}_{st}^{\mathcal{G}}(t)| = n_p - 1 \tag{17}$$

It is important to notice that algorithms for computing the EMST in the plane are available with complexity $O(\text{n log n})$, and faster randomized algorithms of complexity $O(\text{n log log n})$ have also been presented in the literature (see Kleinberg et al. (2005)). Thus, on-line EMST computation is feasible.

The following problem is under investigation:

***Problem** 1.* Given the phantom graph $\hat{\mathcal{G}}(kT)$, find the minimal number $n_r(kT)$ of RAs to be deployed in the field to restore the connectivity of the overall graph, including both the PAs and the employed RAs, and compute admissible waypoints $r_i^*(kT) \in \mathbb{R}^2$ $(i = 1, 2, ..., n_r(kT))$.

The following procedure that gives a viable solution to the Problem 1 is outlined.

At $t = (k-1)T$, $k = 0, 1, 2, ...$, compute the EMST of the phantom graph $\hat{\mathcal{G}}(kT)$ and let L_i, $i = 1, 2, ..., n_p$ be the lengths of the corresponding arcs.

Clearly, the phantom graph $\hat{\mathcal{G}}(kT)$ will be connected if and only if $L_i \leq \beta \ \forall i$. If any of the $L_i's$ is greater then β the phantom graph is disconnected and we need to drop in the field at lest one RA to retain the connectivity. To account for the uncertainty in the estimated PAs positions, quantified by (14), the more conservative visibility range $\beta - E_p$ is going to be considered. Starting from the EMST, it can be identified a viable solution to Problem 1 (way points computation).

(1) In accordance with Algorithm 1, compute the phantom graph $\hat{\mathcal{G}}(kT)$, and its EMST
$$\mathcal{ST}\left(\hat{\mathcal{G}}(kT)\right) = \left(\mathcal{V}_{st}^{\hat{\mathcal{G}}}, \mathcal{W}_{st}^{\hat{\mathcal{G}}}(i,j,kT)\right)$$
having the edge set $\mathcal{E}_{st}^{\hat{\mathcal{G}}}(kT)$.

(2) Compute the subset
$$\mathcal{D}_{st}^{\hat{\mathcal{G}}}(kT) = \left\{(i,j) \in \mathcal{E}_{st}^{\hat{\mathcal{G}}}(kT) : \mathcal{W}_{st}^{\mathcal{G}}(i,j,t) > \beta - E_p\right\} \tag{18}$$
of "disconnected" edges in the EMST of the phantom graph having length greater than the visibility range β minus the maximal uncertainty E_p in the predicted PAs positions.

(3) Construct the vector $r(kT)$ of the RAs way-points according to the following Algorithm 2 (way-points computation)
   (a) *Set* $s = 0$;
   (b) *For all* $(i,j) \in \mathcal{D}_{st}^{\hat{\mathcal{G}}}(kT)$ *do:*
      - set $m_{ij} = ceil\left(\frac{\|q_j - q_i\|}{\beta - E_p}\right)$
      - For $h = 1 : m_{ij}$
      - $s := s + 1$
      - set $\alpha_s = \frac{h}{m_{ij}+1}$
      - set $r_s^*(kT) = q_i(kT) + \alpha_s(q_j(kT) - q_i(kT))$
      - EndFor
   (c) *Set* $n_r^k = s$;

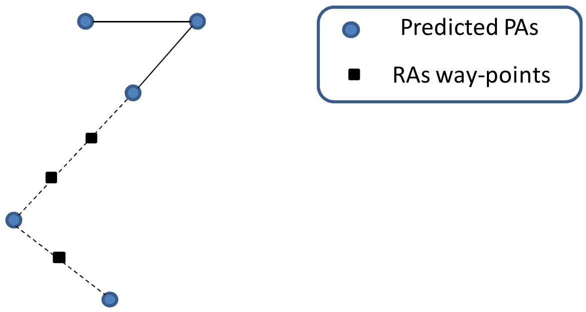

Fig. 2. Example of application of Algorithm 2

Algorithms 1 and 2 compute (a) the total number n_r^k of RAs and (b) the vector $r^*(kT) = [r_1^*(kT), ..., r_{n_r^k}^*(kT)]$ that constitutes the positions of the way points the RAs are supposed to be dispatched to.

Basically, for each (i, j) edge in the subset $\mathcal{D}_{st}^{\hat{\mathcal{G}}}(kT)$ of the "disconnected" edges, $m_{ij} = ceil\left(\frac{\|q_j - q_i\|}{\beta - E_p}\right)$ RAs are deployed in equi-spaced manner along the straight line connecting the vertices.

Thus, the total number of RAs to be dispatched at $t = kT$ is given by

$$n_r^k = \sum_{(i,j)\in\mathcal{D}_{st}^{\hat{\mathcal{G}}}(kT)} ceil\left(\frac{\|q_j(kT) - q_i(kT)\|}{\beta - E_p}\right) \quad (19)$$

Note that Algorithm 1 is executed at each instant $t = (k-1)T$, $k = 1, 2, ...$ After each execution of the Algorithm 1, path planning and motion control modules are activated to steer the RAs towards their way-points by the time $t = kT$, $k = 1, 2, ...$ Figure 2 shows an example of application of Algorithm 2. The continous lines denote the connected edges having the weight smaller than $\beta - E_p$, whereas the dotted lines are the disconnected ones having the weight larger than $\beta - E_p$. It is seen that the proper number of RAs is suitably located in equi-spaced manner along the disconnected edges of the phantom graph.

5.3 Module M_3: RAs assignment.

Step 1. After having calculated (at $t_{k-1} = (k-1)T$) the number of RAs needed to retain the connectivity of the network at $t_k = kT$, and having computed the corresponding way-points, the problem of making an optimal assignment of the way-points to the RAs already being in the field emerges. The following algorithm is proposed for solving this problem.

Step 2. Let n_r^{k-1} be the number of RAs already in the field at the time $(k-1)T$, and let $r((k-1)T) = r^*((k-1)T) = [r_1^*((k-1)T), r_2^*((k-1)T), ..., r_{n_r^{k-1}}^*((k-1)T)]$ be their actual locations. Let n_r^k be the number of RAs to be dispatched at $t = kT$ to the corresponding way-points $r^*(kT) = [r_1^*(kT), r_2^*(kT), ..., r_{n_r^k}^*(kT)]$.

Step 3. If $n_r^k < n_r^{k-1}$ then some RAs can be recalled back to the depot, whereas if $n_r^k > n_r^{k-1}$ new RAs should enter the field. Finally, if $n_r^k = n_r^{k-1}$ the number of RAs already in the field at $t = (k-1)T$ is enough to retain connectivity at the future time $t = kT$ as well.

Note that in all cases, the way-points $r_j^*(kT)$, $j = 1, 2, ..., n_r^k$ need to be properly assigned to the RAs (either being already in the field, or staying in the depot). This assignment can be done differently. Specifically, each way point in the vector $r^*(kT)$ is assigned to the RA which is the closest one to that way point at the time $(k-1)T$. This algorithm is employed in this preliminary work. Another possible algorithm of the way point assignment to each RA is to minimize the overall distance to be traveled by all RAs to reach their new way points $r_j^*(kT)$.

6. THE LOWER LEVEL OF THE CONTROL SYSTEM

6.1 Module M_4: Path Planning and Collision Avoidance

The problem addressed in this section is that of finding suitable paths in the plane connecting the initial RAs location $r((k-1)T)$ to their way-points $r^*(kT)$.

Without the loss of generality the proposed solution is illustrated for the $j - th$ RA only, whose current location at time t is $r_j(t) = [r_{jx}(t), r_{jy}(t)]$, and that must be moved to the way point $r_j^*(kT)$ by the time $t = kT$.

We take advantage of the artificial potential strategy outlined in Ferrara et al. (2008) and Guldner et al. (1996) in order to propose the collision avoidance strategy. Circular security regions of a radius $R = E_p$ centered around the actual positions $p_i((k-1)T)$ of each PA ($i = 1, 2, ..., n_p$) are considered.

Note that the selected radius of the security region guarantees that the motion of the PAs cannot leave the security circle during the time interval $[(k-1)T, kT]$ as a consequence of Assumption 1.

Let $g = (x, y)^T$ denotes an arbitrary position vector in the plane, and define

$$U(g, q_i, w_i) = ln\left(\frac{1}{d(g, w_i)}\right) - \frac{R}{R + d(q_i, w_i)} ln\left(\frac{1}{d(g, q_i)}\right) \quad (20)$$

Let q_a be the closest PA to the j-th RA at time t, and denote

$$U_j(g) = U(g, q_a, r_j^*(kT)) \quad (21)$$

Compute the gradient of the potential field $U_j(g)$

$$E_j(g) = -\nabla U_j(g) = [E_x(g), E_y(g)]^T \quad (22)$$

The motion of the j-th RA is controlled according to (6) by

$$\begin{aligned} v_{jx}^r(t) &= E_x(r_j(t))V_j^r(t), \\ v_{jy}^r(t) &= E_y(r_j(t))V_j^r(t) \end{aligned} \quad (23)$$

in a way to track the gradient of the harmonic potential field $U_j(g)$. The scalar command signal $V_j^r(t)$ sets the speed at which the j-th RA travels along the gradient lines, and it must be selected in order to steer the j -th RA to its way-points by the time kT. This is done in the next subsection by a continuous finite-time control technique.

6.2 Module M_5: path tracking via finite-time velocity control

The following Lemma presents the proposed strategy for designining the finite-time convergent velocity command signal $V_j^r(t)$.

***Theorem** 6.1.* Let the motion of the j-th RA in the time interval $[(k-1)T, kT]$ be governed by (6), (23), (20)-(22) with

$$V_j^r(t) = -\frac{\gamma}{H(r_j(t), r_j^*(kT))} \times \left(\sqrt{d(r_j(t), r_j^*(kT))} + d(r_j(t), r_j^*(kT))\right) \quad (24)$$

where

$$H(r_j(t), r_j^*(kT)) = \frac{\left(r_j(t) - r_j^*(kT)\right)^T E_j(r_j(t))}{d(r_j(t), r_j^*(kT))} \quad (25)$$

$$\gamma \geq \frac{2}{T} ln \left(1 + \sqrt{d(r_j((k-1)T), r_j^*(kT))}\right) \quad (26)$$

Then, the way-point is reached in finite-time by $t = kT$, i.e.

$$r_j(kT) = r_j^*(kT) \quad (27)$$

Proof of Lemma 6.1. Denote the distance between the j-th RA and its way-point as

$$d_j(t) = d(r_j(t), r_j^*(kT)) = \sqrt{\left(r_{jx}(t) - r_{jx}^*(kT)\right)^2 + \left(r_{jy}(t) - r_{jy}^*(kT)\right)^2} \quad (28)$$

The time derivative of (28) takes the form

$$\dot{d}_j(t) = -\frac{1}{d(r_j(t), r_j^*(kT))} \left(r_j(t) - r_j^*(kT)\right)^T \times E_j(r_j(t)) V_j^r(t) = H(r_j(t), r_j^*(kT)) V_j^r(t) \quad (29)$$

Substituting (24) in the right hand side of (29) yields

$$\dot{d}_j(t) = -\gamma\sqrt{d_j(t)} - \gamma d_j(t), \quad t \in [(k-1)T, kT] \quad (30)$$

The dynamics (30) converges to zero in finite time for all $\gamma > 0$, with a transient time (see Yu et al. (2005))

$$\tau = \frac{2}{\gamma} ln \left(1 + \sqrt{d_j((k-1)T)}\right) \quad (31)$$

It follows from (26) and (31) that condition

$$\tau \leq T \quad (32)$$

holds, from which the relation $d(T) = 0$ follows, which straightforwardly yields (27). Lemma 6.1 is proven. □

7. MAIN RESULT

The guaranteed connectivity properties for the MANET are specified in the following Theorem.

***Theorem** 7.1.* Consider the MANET consisting of n_p moving PAs, fulfilling the Assumption 1 and with the visibility range β. Assume that Assumptions 2 and 3 hold and the motion of the RAs is governed by the CC according to the two-level control algorithm consisting of the modules M1-M5. Let $T < \min\left\{\beta/v_M, \sqrt{2\beta/a_M}\right\}$. Then, the MANET is guaranteed to be connected at each $t = kT$, $k = 1, 2, ...$

Proof of Theorem 7.1. The predicted PA positions (13), (9)-(11) feature an uncertainty quantified by (14). Due to this uncertainty, the connectivity of the phantom graph should be enforced in a robust manner, i.e. by allowing the position of each PA to be inside a circle of center $(\hat{p}_{ix}(kT), \hat{p}_{iy}(kT))$ and radius Ta_M. Let $r^*(kT)$ be computed according to Algorithm 1. The complete graph $\hat{\mathcal{G}}^*(kT)$, including both the PAs $\hat{q}(kT)$ and the employed RAs while $r(kT) = r^*(kT)$ is now shown to be connected. Due to Algorithm 2, the distance h between the adjacent RAs along the (i, j) arc in the set $\mathcal{D}_{st}^{\hat{\mathcal{G}}}(kT)$ is

$$h = \frac{\|q_j - q_i\|}{m_{ij} + 1}$$

Since by construction $\|q_j - q_i\| < (m_{ij} + 1)\beta - E_p$, it straightforwardly follows that $h < \beta - E_p < \beta$ which implies that the graph $\hat{\mathcal{G}}^*(kT)$ admits a minimum spanning tree with all edges having length not exceeding the threshold value $\beta - E_p$. Thus, the phantom graph $\hat{\mathcal{G}}^*(kT)$ is connected. The path planning and collision avoidance module M_4, and the finite time velocity control module M_5 guarantee that all RAs are steered towards the respective way points by the time $t = kT$ as proven in Lemma (6.1). Thus, the graph based on the *actual* PA and RA positions at $t = kT$ admits a minimum spanning tree with all edges having length not exceeding the threshold value β. This proves the Lemma. □.

8. SIMULATION RESULTS

The considered architecture has been verified via computer simulations. The unit square $[0, 1] \times [0, 1]$ has been considered as the area of operation of the mobile vehicles. The depot location of the RAs is assumed to be in the origin. The visibility radius is set to $\beta = 0.3$ whereas the length T of the receding horizon window is $T = 0.5$.

In the preliminary tests, the PAs are considered to be at rest. Figure 3 shows the stationary locations of the 5 PAs and the RAs trajectories in the plane during a simulation tests of duration T. PAs are denoted by red circles, the way points are denoted by green squares, and the blue dotted trajectory denotes the trajectories of the RAs, sampled every 0.05 seconds. The continuous blue arcs between the PAs denotes those edges of the graph with length smaller than the visibility radius β, whereas the dotted red arcs correspond to edges with length bigger than β. It is seen that two RAs must be sent to the field, and the corresponding way points are located in the middle of the dotted "disconnected" edges. It is also seen that by the time T all RAs have reached the corresponding way point, thereby ensuring that the MANET is connected at $t = T$. Figure 4 shows the temporal profile of the x and y coordinates of an RA, along with the corresponding target values (i.e., the way point coordinates). It is clear that the RAs convergence takes place by the prescribed time T. The results of a similar test, corresponding to the more evolved situation with $n_p = 10$ static PAs, is shown in the Figure 5. The collision avoidance imposed by the repulsive action of the PAs over the moving RAs is apparent from the inspection of the RAs trajectories.

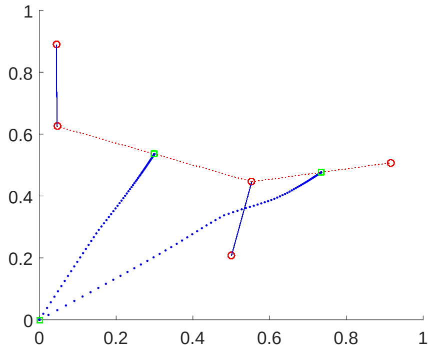

Fig. 3. Static PAs test with $n_p = 5$. The RAs evolving towards their way points

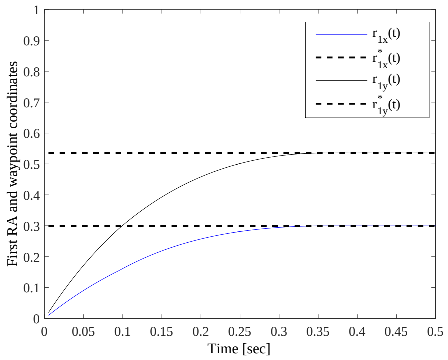

Fig. 4. Static PAs test with $n_p = 5$. RA and way point coordinate vs time

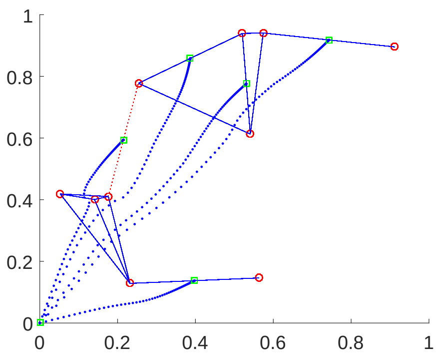

Fig. 5. Static PAs test with $n_p = 10$. The RAs evolving towards their way points.

Further simulations with $n_p = 5$ moving PAs have been carried out. The window length has been reduced to $T = 0.1$. The maximal velocity ad acceleration of the PAs were selected as $v_M = 1$ and $a_M = 0.5$. Figure 6 shows the temporal evolution of the x coordinate of the first RA, along with the x coordinate of the corresponding way point. As expected, the way point is updated each T seconds, and it is apparent that the finite-time velocity control strategy guarantees that the way points are successfully reached by the prescribed times. Figure 7 shows the time evolution of an ad-hoc boolean variable which holds the zero value when the MANET is disconnected, and the unit value when the MANET is connected. Numerical simulations show that the proposed scheme is able to ensure the connectivity of the MANET not only at the time instants $t = kT$ $(k = 1, 2, ...)$, as demonstrated in the Lemma 7.1, but also in almost all the time instants after an initial transient in the considered example. This suggest that the method has good potential and that the performance of the proposed two-layer control architecture goes beyond to that proven by the theoretical analysis. A video showing the dynamical evolution of the network with PAs and RAs with $n_p = 20$ is available in URL (2016) to allow the reader to inspect the performance and scalability of the proposed control logic in such a challenging scenario.

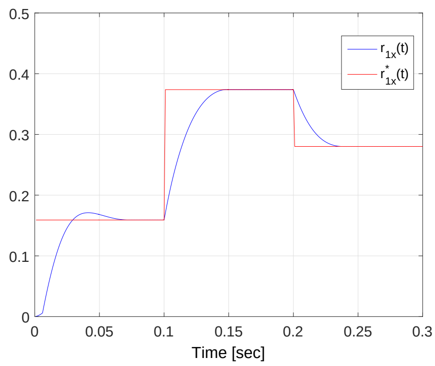

Fig. 6. Dynamic PAs test with $n_p = 5$. Actual and reference x coordinates vs time of the first RA

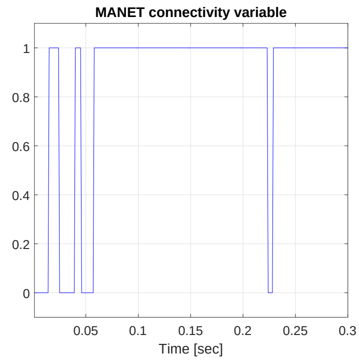

Fig. 7. Dynamic PAs test with $n_p = 5$. The MANET's connectivity variable vs. time

9. CONCLUSIONS

The proposed scheme guarantees the connectivity of the MANET at the time instants $t = kT$, $k = 1, 2, ...$ However, it is worth noting that the simulation results have shown that the proposed two-level control algorithm provides connectivity at the much broader time interval. Future research efforts will be dedicated to improving the proposed control algorithms to the level that guarantees connectivity on on broader time intervals.

Several alternative improvements are possible. Note that Algorithm 1 yields a viable solution to Problem 1 where the number n_r of RAs is not minimized. A solution can be improved by reducing the number of required RAs. The underlying idea could be based on the Steiner tree concept. The main difference between the Steiner tree (see Van Laarhoven (2010)) and the MST problems is that in the Steiner tree problem extra intermediate vertices and edges may be added to the graph in order to reduce the length of the spanning tree. These new vertices are known as Steiner vertices. It is clear that there is a deep connection between the Steiner tree and the problem at hand, with the Steiner vertices being good candidates for the RAs waypoints. Unfortunately, the problem of Steiner tree computation in NP complete, but interestingly the Steiner tree for three and four nodes has a known solution (see figures 8) which can be employed in future developments to to devise heuristics improving the viable solution given by Algorithm 1.

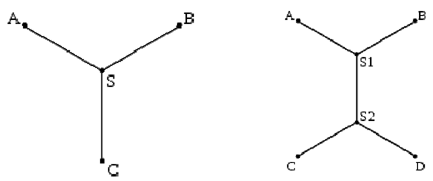

Fig. 8. Steiner patterns for 3 and 4 vertices.

Additional ingredients for improving the proposed methodology could be a more precise prediction of the future PA positions (using, for instance, the estimated velocity and acceleration provided by higher order robust sliding-mode differentiators Levant (2005)).

REFERENCES

Bordetsky, A., and Netzer, D. (2010) Testbed for tactical networking and collaboration. *Int. C2 Journal*, volume 4,(3), 1–31.

Edwards, C., Menon, P.P., Shtessel, Y., and Bordetsky, A. (2013). On Connectivity Preservation in Mobile Wireless Multi-Agent/Node Mesh Networks. Proc. 2013 American Control Conference (ACC 2013), Washington DC, US, 5901–5906.

Godsil, C., and Royle, G. (2001). *Algebraic Graph Theory*, vol. 207. Berlin, Germany: Springer-Verlag, Graduate Texts in Mathematics Series.

Ferrara, A., and Rubagotti, M. (2008). Second-order sliding-mode control of a mobile robot based on a harmonic potential field. *IET Contr. Th. Applicat.*, 2,(9), 807–818.

Franceschelli, M., Giua, A., and Pisano, A. (2016). Finite-Time Consensus on the Median Value with Robustness Properties. *IEEE Trans. Aut. Contr.*, Accepted as regular paper. Available in IEEE Xplore. DOI: 10.1109/TAC.2016.2590602.

Franceschelli, M., Giua, A., Pisano, A., and Usai. E. (2015). Finite-Time Consensus with Disturbance Rejection by Discontinuous Local Interactions in Directed Graphs. *IEEE Trans. Aut. Contr.*, 60,(3), 1133–1138.

Guldner, J., and Utkin, V.I. (1996). Tracking the gradient of artificial potential fields: sliding mode control for mobile robots. *Int. J. Contr.*, 6(3), 417–432.

Holleran, S., Baev, S., and Shtessel, Y. (2010). Preventing Disruption of a Mobile Communication Network using Higher Order Sliding Mode Control. Proc. 49-th IEEE CDC 2010, Atlanta, US, 2366–2371.

Jadbabaie, A., Lin, J. and Morse, A.S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Aut. Contr.*, 48(6), 988–1001.

Kanchanasut K., Tunpan A., Awal M., Das D., Wongsaardsakul T., and Tsuchimoto Y. (2007). A Multimedia communication System for Collaborative Emergency Response Operations in Disaster Affected Areas. *Int. J. of Emergency Management*, 4(4), 670–681.

Kleinberg, J. and Tardos, E. *Algorithm Design*. Addison Wesley, 2005.

Levant, A. (2005). Higher-order sliding modes, differentiation and output-feedback control. *Int. J. Contr.*, 76, 924–941.

Lloyd, E.L. and Xue, G. (2007). Relay node placement in wireless sensor networks, *IEEE Trans on Computers*, 56, 134–138.

McLain, T. W., and Beard, R. W. (2003) Cooperative path planning for timing-critical missions. Proc. 2003 American Control Conference (ACC 2003), 296–301.

Mesbahi, M., and Egerstedt, M. *Graph theoretic methods in multi-agent networks*, Princeton Series in Applied Mathematics, 2010.

Misra, S., Hong, S.D., Xue, G., and Thang, J. (2008). Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements. Proc. 2008 IEEE Conference on Computer Communications, Phoenix, 281–285.

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

Pallottino, L., Scordio, V.G., Frazzoli, E. and Bicchi, A. (2006). Decentralized and scalable conflict resolution strategy for multi-agents systems. Proc. 2006 Int. Symp. on Mathematical Theory of Networks and Systems (MTNS 2006).

Ren, W., Beard, R.W., and Atkins, E.M. (2007). Information consensus in multivehicle cooperative control. *IEEE Contr. Syst. Mag.*, 27(2), 71–82.

Simonetto, A., Keviczky, T., and Babuk, R. (2011). On Distributed Maximization of Algebraic Connectivity in Robotic Networks. Proc. 2011 American Control Conference (ACC 2011), San Francisco, US, 2180-2185.

Shtessel, Y., Edwards, C., Menon, P., Cosby, A., and Bordetsky A. (2012). Predictive modeling and retaining connectivity for mobile wireless communication network. Proc. 2012 IEEE Conference on Technologies for Homeland Security, November.

Sabattini, L., and Secchi, C., and Chopra, N., and Gasparri, A. (2013) Distributed Control of Multirobot Systems With Global Connectivity Maintenance. *IEEE Transactions on Robotics*, vol. 29, num. 5, 1326–1332.

Tian, D. and Georgeanas, N. (2005). Connectiviety maintenance and coverage preservation in wireless sensor networks. *Ad hoc Networks*, 3,(6) 744-761.

Van Laarhoven, J.W. (2010). Exact and heuristic algorithms for the Euclidean Steiner tree problem. PhD Thesis, Univ. of Iowa, 2010. Available online.

Yu, S., Yu, X., Shirinzadeh, B., and Man, Z. (2005) Continuous finite-time control for robotic manipulators with terminal sliding-mode. *Automatica*, 41(11), 1957–1964.

Zavlanos, M.M., Egersted, M.B., Pappas, G.J. (2011). Graph-Theoretic Connectivity Control of Mobile Robot Networks. *Proc. IEEE*, 99(9), 1525–1540.

URL (2016). https://www.youtube.com/watch?v=MbFFSOcdP68