Coordinated Control of Mobile Antennas for Ad-Hoc Networks

Gianluca Antonelli⋆, Filippo Arrichiello⋆, Stefano Chiaverini⋆, Roberto Setola†

⋆ Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale
Università degli Studi di Cassino
via G. Di Biasio 43, 03043 Cassino (FR), Italy
{antonelli,f.arrichiello,chiaverini}@unicas.it

† Laboratorio Sistemi Complessi e Sicurezza
Facoltà di Ingegneria
Università Campus Bio-Medico di Roma
via E. Longoni 83, 00155 Roma, Italy
r.setola@unicampus.it

Submitted for publication at the

International Journal of Modelling, Identification and Control

Special/Inaugural issue on *Intelligent Robot Systems*, 1st September 2005

Abstract

This paper investigates the implementation of a wireless mobile ad-hoc network to guarantee that an autonomous agent, i.e., an autonomously driven mobile vehicle or a human, remains connected to a fixed base station while performing its own mission. To the purpose, the use of a platoon of mobile robots is proposed to carry a number of repeater antennas; these are suitably moved to dynamically ensure a multi-hop communication link to the moving agent, hence extending and adapting the area covered by the sole base station. Self configuration of the robots' platoon is achieved by a singularity-robust task-priority inverse kinematics algorithm via the definition of suitable task functions. The obtained simulation results show the effectiveness of the proposed approach.

Keywords: Mobile Ad-hoc NETworks (MANETs), Coverage Area Adaptation, Multi-Robot Systems, Mobile Robots, Motion and Path Planning, Inverse Kinematics Techniques

1 Introduction

Communication among mobile agents is a crucial requirement in many applications to support cooperation of team members in fulfilling a given mission.

Traditional solutions to the problem of mobile communication, e.g., cellular networks, require coverage of a wide area through a number of space-fixed antennas that serve partly overlapping local areas (the cells); an user then communicates with the others through a wireless connection to the local antenna that conveys all the traffic within the cell.

While the above static placement of the antennas is cost/performance effective when there always is a large number of users to be connected within the cell (e.g., metropolitan areas or highway paths), in many applications it would be more convenient to adopt an approach where the antennas are dynamically placed to optimize the coverage needs as in survey and rescue missions to be performed inside a damaged building or in exploration of unknown areas.

In this scenario, wireless Mobile Ad-hoc NETworks (MANETs) represent a promising approach to overcome the drawbacks of fixed-configuration networks. A MANET is a collection of autonomous nodes that communicate each other without using any fixed networking infrastructure. Each node in a MANET operates as both a host and a router. Therefore, any node can communicate directly with nodes that are within its transmission range and, in order to reach a node that is out of its range, data packets are relayed over a sequence of intermediate nodes using a store-and-forward multi-hop transmission principle.

The research in this field has focused mainly on static arrangements of sensors to optimize different cost function (and specifically to maximize coverage area) and the development of protocols to efficiently route information through the network.

This approach is very attractive also for mobile robot applications. For this reason, a number of researchers have started proposing ad-hoc wireless solutions for mobile robot communication (see, e.g.,[10, 15, 16, 21]).

However, these solutions consider position and motion of each node as uncontrollable variables. Assuming instead that at least some of the nodes may be controlled, more performing MANETs can be designed. Indeed, a suitable dynamic reconfiguration of the robots' position allows to adapt the coverage area of the communication network to better support the team's mission, to avoid signal fading area (e.g., that induced by the presence of obstacles), and to handle possible fault of some team members.

First studies on this topic were related to use mobile robots to provide adaptive sensor network in a dynamic environment. In [18] Delaunay tessellation and Voronoi diagram are used to define an algorithm able to maximize the coverage area of the network. A more general formulation for the coverage problem is proposed in [9], where a distributed asynchronous algorithm is developed to dynamically reconfigure a mobile sensors network taking into account sensor performance; specifically, in this approach the control input of each vehicle is proportional to the distance from its actual position and the centroid of the associated Voronoi region. In reference [6], to guarantee communication for a mobile robot involved into a dynamic coverage problem, a static network of markers is autonomously dispersed by the robot during its motion (these markers are also used to improve localization capability). In reference [14] cooperative communication, i.e., enable to a group of mobile robot to establish and maintain a wireless ad-hoc network, is achieved thorough a distributed approach where each robot defines its local optimal trajectory by minimizing a functional that accounts for network connectivity, obstacle avoidance and other aspects of interest. Specifically a first-order prediction model is used to predict one step ahead network topology and to identify the region where there is higher probability of maintaining network connectivity. A behavior-based architecture is proposed in [20] to encourage a team of robots to maintain a local communication network while exploring an area; during exploration, any two robots that form a bridge connection, i.e., a link whose removal disconnects the network, are forced to perform connectivity-behavior task when their distance are greater than a given threshold. An algorithm that, starting from an arbitrary initial connected condition, modifies the robots' location to achieve a fault-tolerant bi-connected configuration is proposed in [5]. Recovery behaviors to autonomously re-establish communication after a failure between some or all of the nodes in the network are discussed in [19]. A rendezvous algorithm for coordinated motion of mobile robots with limited mobility and communication constraints is proposed in [12]; in this approach, network connectivity is guaranteed imposing to each robot, at each sample time, to move toward the circum-center defined by itself and its neighbors and restricting the travel distance.

In this paper, following the preliminary results in [1], we consider the problem of dynamically adapting the configuration of a platoon of mobile robots equipped with a wireless device (that we will call *antennas*) so as to realize an ad-hoc network to support communication of an autonomous agent, i.e., an autonomously driven vehicle or an human, with a *fixed base station*. Antennas coordination for agent coverage is achieved in the framework of kinematic control of platoon of mobile robots [2, 4], by resorting to properly defined task functions. This approach has shown to be efficient and reliable in simultaneous handling of several control objectives, namely, agent-antennas-station connectivity, obstacle avoidance and fault tolerance of the mobile antennas.

The paper is organized as follows. First, we provide a background on the use of inverse kinematics techniques for mobile robots' coordination and how this approach allows to handle several tasks. Then, we present in detail the MANET problem considered and illustrate the proposed solution. Simulation results showing the effectiveness of the proposed method are reported in Section 4. Conclusive remarks are the given in the last Section.

2 Coordination of Multi-Vehicle Systems

This Section presents a brief review on the use of the Singularity-Robust Task-Priority (SRTP) inverse kinematics for controlling a platoon of autonomous mobile robots. This approach has been proposed in [2], and represents an extension of the technique proposed in [7] by inheriting the singularity robustness capabilities developed in [8] for ground-fixed redundant manipulators. The possibility to control a platoon of robots using the SRTP approach has been deeply investigated in [2] and [4], where task functions such as the platoon average position, the variance, or complex mission such as to escort a target in presence of obstacles are discussed.

In view of the MANET application to be developed, the term *vehicle* will denote in this Section either an antenna engaged in ensuring the network connectivity or, eventually, the robot executing the mission; accordingly, the term *platoon* will denote a multi-vehicle system, that is a mobile network in case of sole antennas. While considering a system of n vehicles, our aim is to control the value taken by some generic task function which suitably depends on the platoon state.

Let define the position of the i-th vehicle as

$$\boldsymbol{p}_i = [x_i \; y_i \; z_i]^{\mathrm{T}}\,.$$

To keep the notation compact it is also useful to define the vectors

$$\boldsymbol{p} = [\boldsymbol{p}_1 \;\ldots\; \boldsymbol{p}_n]^{\mathrm{T}}$$

and

$$\boldsymbol{v} = [\dot{\boldsymbol{p}}_1 \;\ldots\; \dot{\boldsymbol{p}}_n]^{\mathrm{T}}.$$

By defining as $\boldsymbol{\sigma} \in \mathbb{R}^m$ the task variable to be controlled, it is:

$$\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) \tag{1}$$

with

$$\dot{\boldsymbol{\sigma}} = \sum_{i=1}^{n} \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_i} \dot{\boldsymbol{p}}_i = \boldsymbol{J}(\boldsymbol{p})\boldsymbol{v}\,, \tag{2}$$

where $\boldsymbol{J} \in \mathbb{R}^{m \times 3n}$ is the configuration-dependent task Jacobian matrix.

An effective way to generate motion references $\boldsymbol{p}_{d,i}(t)$ for the vehicles starting from desired values $\boldsymbol{\sigma}_d(t)$ of the task function is to act at the differential level by inverting the (locally linear) mapping (2); in fact, this problem has been widely studied in robotics (see, e.g., [17] for a tutorial). For low-rectangular matrices —which is usually the case in platoon formation control, being $3n \gg m$— the problem admits infinite solutions and such *redundancy* is often exploited to optimize some criteria. A typical requirement is to pursue minimum-norm velocity, leading to the least-squares solution:

$$\boldsymbol{v}_d = \boldsymbol{J}^{\dagger} \dot{\boldsymbol{\sigma}}_d = \boldsymbol{J}^{\mathrm{T}} \left(\boldsymbol{J} \boldsymbol{J}^{\mathrm{T}} \right)^{-1} \dot{\boldsymbol{\sigma}}_d\,. \tag{3}$$

At this point, the vehicle motion controllers need reference position trajectories besides the velocity references; these can be obtained by time integration of $\boldsymbol{v}_d$. However, discrete-time integration of the vehicles' reference velocities would result in a numerical drift of the reconstructed vehicles' positions; the drift can be counteracted by a so-called Closed Loop Inverse Kinematics (CLIK) version of the algorithm, namely,

$$\boldsymbol{v}_d(t_k) = \boldsymbol{J}^{\dagger} \left(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \widetilde{\boldsymbol{\sigma}} \right) \Big|_{t=t_k} \tag{4}$$

$$\boldsymbol{p}_d(t_k) = \boldsymbol{p}_d(t_{k-1}) + \boldsymbol{v}_d(t_k) T\,, \tag{5}$$

where t_k is the k-th time sample, the error vector $\widetilde{\boldsymbol{\sigma}}$ is defined as

$$\widetilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}\,, \tag{6}$$

T is the sampling period, and $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times m}$ is a suitable constant positive-definite matrix of gains.

Different task functions may be used at the same time by stacking the corresponding single task variables in an overall task vector. As a result, the corresponding single task Jacobians are stacked too in an overall task Jacobian, and the inverse solution (3) acts by simply adding the partial vehicles' velocities that would be obtained if (locally at the current configuration) each task were executed alone. It is simple to recognize that this approach is poorly effective since conflicting tasks would generate counteracting partial vehicles' velocities.

A possible technique to handle this problem has been proposed in [7], which consists in assigning a relative priority to the single task functions, thus resorting to the task-priority inverse kinematics introduced in [11, 13] for ground-fixed redundant manipulators. Nevertheless, as discussed in [8], just in case of conflicting tasks it is necessary to devise singularity-robust algorithms that ensure proper behavior of the inverse velocity mapping. For this reason, according to [8], the contributions of each task are calculated following the formula (4) and, starting from the lowest level, each contribution has to be properly projected onto the null-space of the higher-priority task. Denoting with $\boldsymbol{v}_i$ the generic contribute of the i-th task quantity

$$\boldsymbol{v}_i = \boldsymbol{J}_i^{\dagger} \left(\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{\Lambda}_i \widetilde{\boldsymbol{\sigma}}_i \right) \tag{7}$$

and supposing that i denotes also the priority order, then, the formula for three tasks is:

$$\boldsymbol{v}_d = \boldsymbol{v}_1 + \left(\boldsymbol{I} - \boldsymbol{J}_1^{\dagger} \boldsymbol{J}_1 \right) \left[\boldsymbol{v}_2 + \left(\boldsymbol{I} - \boldsymbol{J}_2^{\dagger} \boldsymbol{J}_2 \right) \boldsymbol{v}_3 \right]\,, \tag{8}$$

whose graphical representation is reported in Figure 1; notice that the supervisor is in charge of assigning the priority among generic tasks (denoted with the a letter subscript in the figure) simply by changing their order in the above formula.

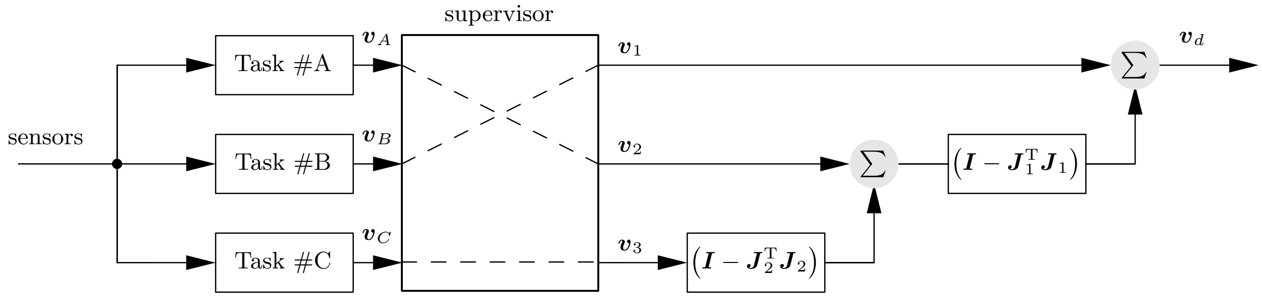

Figure 1: Sketch of the task-priority algorithm implemented. At each sample time, the *supervisor* block, on the base of the actual antennas' configuration, dynamically assign the priority of the tasks (e.g., as in the figure inverting the priority between task #A and #B).

Equation (8) has a nice geometrical interpretation. Each task is projected onto the vehicles' velocity space by the use of the corresponding pseudoinverse, i.e., as if it were acting alone; then, before adding its contributions, the lower-priority, e.g., secondary, task is further projected onto the null space of the higher-priority, e.g., primary, task so as to remove those velocity components that would conflict with it. In is worth noticing that this approach is different with respect to energy-based formulation because it automatically modify on the base of actual configuration the degree of importance of each single task function.

3 The MANET case

The algorithm described in the previous Section can be used to suitably coordinate the motion of a set of mobile antennas so as to implement a MANET that dynamically adapts its coverage area (see Figure 2).

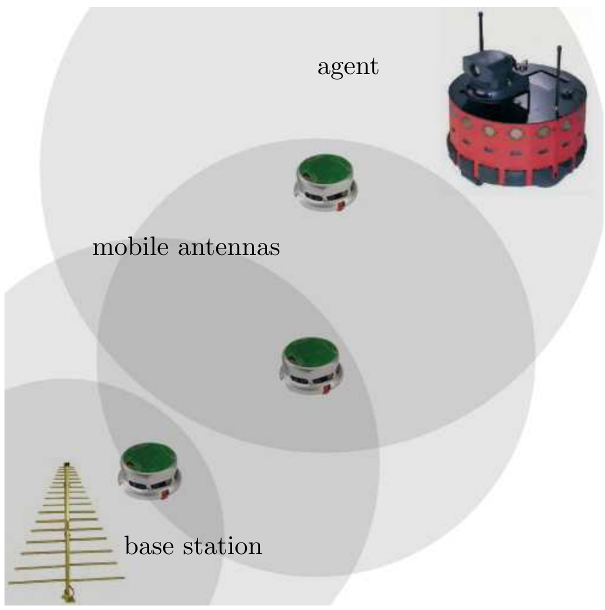

Figure 2: Sketch of the coverage problem to be solved; the autonomous agent needs to be connected with the base station by the use of a platoon of mobile antennas.

Specifically, in the following we consider a platoon of n mobile antennas that must ensure the communication between a mobile agent executing its mission and a fixed base station (e.g., an Internet access point). Of course, the mobile agent and the antennas have also to avoid the obstacles eventually present in the environment.

Let us denote with the subscript i the generic antenna as well as the agent, i.e., it is $i \in [0, n+1]$ where $i = 0$ denotes the base station and $i = n+1$ denotes the agent.

Each antenna is supposed to cover in open space a circular area $\mathcal{I}_i$ centered in $\boldsymbol{p}_i$ and of radius

$$r_{\max,i} = r_{\max,i}\left(i, pw_i\right) \, , \tag{9}$$

where pw_i is the current level of power used for transmission. It can be noticed that $r_{\max,i}$ depends on the characteristic of the i-th antenna. Moreover, it depends also on the level of power used for transmission pw_i. During the execution of a mission the level of power used for the transmission may be varied in order to preserve the battery level or, on the other side, can be decreased when the battery level is low.

In order to guarantee adequate operative margins, we will constraint each antenna to dialogue in a smaller region, identified via the radius

$$d_{\max,i} = (1 - \varDelta)\, r_{\max,i} \qquad\qquad 0 < \varDelta < 1 \, , \tag{10}$$

where $\varDelta$ is a design parameter. Notice that, in presence of obstacles, it is possible to decrease $d_{\max,i}$ to take into account the possible reduced propagation. Moreover, in order to avoid collisions, all other bodies in the environment (i.e., base station, other antennas, agent and obstacles) must be farther than $d_{\min,i}$ from each antenna. Notice that this definition is similar to that of *comfort zone* in [20], therefore, this term will be used in the following to denote the area shown in Figure 3.

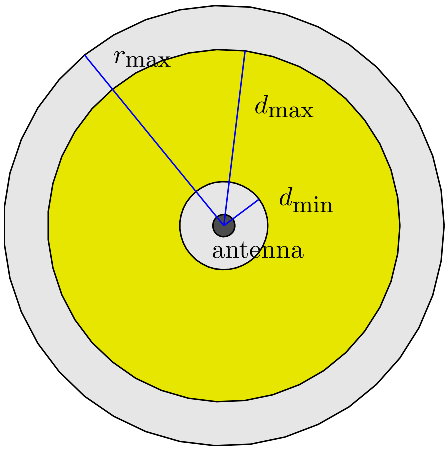

Figure 3: Comfort zone of an antenna and relevant distances.

Then, at time t the i-th vehicle is able to directly communicate with all vehicles which are inside the region $\mathcal{I}_i(t)$. The set of all direct communication paths constitute a graph that can be synthetically represented via the associated incidence matrix $\boldsymbol{G}$ whose entries satisfy

$$g_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{p}_j \in \mathcal{I}_i \\ 0 & \text{otherwise} \, . \end{cases}$$

Notice that, since $\boldsymbol{p}_i \notin \mathcal{I}_i$, it results $g_{ii} = 0$ by definition.

If all the antennas have the same transmission's characteristics and are using the same level of power, then $\boldsymbol{G}$ is a symmetric matrix, otherwise this may not be the case.

Let us introduce the communication matrix $\boldsymbol{C}(t)$ defined as

$$\boldsymbol{C}(t) = \boldsymbol{G}(t) + \boldsymbol{G}^2(t) + \cdots + \boldsymbol{G}^n(t) \qquad\qquad \forall t \, , \tag{11}$$

where the i-th row of the k-th term in the sum represents the set of all the vehicles that at time t can be reached by a message generated by the i-th vehicle in exactly k hops.

It is immediate to recognize that bi-directional communication between the i-th and the j-th vehicles is guaranteed if and only if the product

$$c_{ij}(t) \cdot c_{ji}(t) \neq 0 \, . \tag{12}$$

Notice that the value assumed by c_{ij} represents a crude measurement of the degree of redundancy for the related communication channel; indeed it represents the number of different paths existing among the i-th and the j-th vehicles.

Our goal is to guarantee the presence of a radio bridge between the agent and the base station, i.e.,

$$c_{0\,n+1}(t) \cdot c_{n+1\,0}(t) \neq 0 \qquad \forall t \, , \tag{13}$$

while the agent to be covered performs its own mission moving along unknown trajectories.

3.1 The proposed algorithm

Let us define as *virtual chain* a virtual connection that, starting from the base station reaches all the n antennas. The agent is connected, if $\exists i : \boldsymbol{p}_{n+1} \in \mathcal{I}_i$, i.e., if it belongs to the comfort zone of one of the antennas in the virtual chain. We impose to the antennas to autonomously self-arrange in order to create a proper virtual chain or, in other words, that the matrix $\boldsymbol{C}$ satisfies (13).

Let define a n-dimensional vector $\boldsymbol{k}$ that collects the indexes of the antennas in their order from the base station. Thus, k_j (i.e., the j-th component of vector $\boldsymbol{k}$) is the index that identifies the antenna at the j-th place —not necessarily the j-th antenna— and k_j, k_{j+1} index two consecutive vehicles along the circle. As described in Subsection 3.2, the vector $\boldsymbol{k}$ may be modified at each sample time.

Let us define as T the time step of the algorithm, i.e., the time needed to compute the new reference velocity for all the antennas and as T_c the communication time, i.e., the time needed to communicate between two generic antennas. It is required that $T > T_c$. Moreover, each antenna has a maximum velocity denoted as $v_{\max,i}$.

The following assumptions are made:

- $T_c v_{\max,i} \ll d_{\min,i}$, i.e., the configuration of the antennas can be considered as constant during the communication among them;
- $T v_{\max,i} \ll d_{\min,i}$, i.e., the movement is *slow* with respect to the distance $d_{\min}$;
- The environment is considered as known; the information can be available from an off-line map or acquired on-line by resorting to sensor fusion techniques;
- At the initial time, the virtual chain is correctly formed (see Subsection 3.2).

The coverage problem described in Section 3 can be solved by resorting to the technique briefly recalled in Section 2, using the reasonable assumptions above. It is necessary to properly define the task functions and properly arrange them in a priority order. It is worth noticing that the priority is dynamic, in the sense that a supervisor (Figure 1) is in charge of modifying the relative priorities among the tasks based on the comments below.

The task functions to be achieved by the antennas in order of decreasing priority, being 1 the highest and assuming that k_0 is the base station, are:

- for an antenna along the virtual chain k_j with $j \in \{1, n-1\}$:
  1. avoid the obstacles;
  2. keep the next antenna k_{j+1} in the comfort zone;
  3. keep the previous antenna k_{j-1} in the comfort zone.
- for the last antenna in the virtual chain k_n:
  1. avoid the obstacles;
  2. keep the previous antenna k_{j-1} in the comfort zone.

Moreover, for the antenna closest to the agent (not necessarily k_n), the task of keeping the agent in the comfort zone is considered with a priority lower than the obstacle avoidance and higher than the others. With respect to the above list, thus, this task is at a priority 2 while the other tasks from priority 2 skip of one unity downward.

The agent to be covered might move independently from the antennas such as, e.g., in case of a human. Notice that it is possible that it *pulls* them away too far from the base station when, e.g., the antennas are all stretched in a line but the agent is still moving away. Better performance can be achieved if the agent is a robot and it is included as a part of the platoon to be coordinated. This may be achieved by imposing to the robot the following task function:

1. avoid the obstacles;
2. reach a goal or track a path.

that, when the robot is close to the limit of the comfort zone, needs to be modified in:

1. avoid the obstacles;
2. stay within the comfort zone of the closest antenna;
3. reach a goal or track a path.

As extensively verified in simulation, when the robot is close to the limit of the comfort zone, this formulation imposes on the robot some kind of *stop and wait* command until the antennas, after a self reconfiguration, are able to accommodate further motion. In this way, any break in the communication bridge is prevented but, obviously, it is not ensured that the robot reaches its goal position. In case of absence of obstacles this means that, at most, the robot can stretches all the antennas in a line and then stop, preventing its movement outside the coverage area.

The task function aimed at ensuring connection of the chain is

$$\sigma_c = \sum_{i=1}^{n} \sigma_{c,i}\,, \tag{14}$$

where $\sigma_{c,i} \in \mathbb{R}$ provides connection of the antenna i at position k_j to the previous one as

$$\sigma_{c,i} = \begin{cases} \|\boldsymbol{r}\| & \text{if } \|\boldsymbol{r}\| \le d_{\min} \\ 0 & \text{if } d_{\min} < \|\boldsymbol{r}\| < d_{\max} \\ \|\boldsymbol{r}\| & \text{if } \|\boldsymbol{r}\| \ge d_{\max} \end{cases} \tag{15}$$

with $\boldsymbol{r} = \boldsymbol{p}_{k_j} - \boldsymbol{p}_{k_{j-1}}$ and

$$\boldsymbol{J}_{c,i} = \begin{cases} 0 & \text{if } \sigma_{c,i} = 0 \\ \boldsymbol{r}^{\mathrm{T}} & \text{otherwise}\,. \end{cases}$$

According to the above definition, the desired value for the task function is selected as:

$$\sigma_{d,i} = \begin{cases} d_{\min} & \text{if } \|\boldsymbol{r}\| \le d_{\min} \\ 0 & \text{if } d_{\min} < \|\boldsymbol{r}\| < d_{\max} \\ d_{\max} & \text{if } \|\boldsymbol{r}\| \ge d_{\max}\,. \end{cases} \tag{16}$$

In a similar way, it is also possible to define the task so as to impose connection between k_j and k_{j+1}.

Obstacle avoidance is achieved by using a properly defined task function that, for a generic antenna k_j is given by:

$$\sigma_{o,l} = \left\| \boldsymbol{p}_{k_j} - \boldsymbol{p}_{o,l} \right\|\,, \tag{17}$$

where $\boldsymbol{p}_{o,l}$ is the vector representing the closest point of the obstacle to the antenna. Wider discussion on the obstacle avoidance possibilities as well as implementation details will not be given here for seek of space, in the simulation Section a rather complex environment will be handled with this approach. The limit of this obstacle avoidance approach, common to all the path planning algorithm relying on *local* information only, is the possibility to get trapped in local minima. Further details can be found in [3, 4].

Once the task function have been defined, it is possible to compute the error variable in eq. (6) to output the reference velocities for the specific task. Further applying eq. (8) allow to implement the singularity robust task priority inverse kinematics algorithm and compose the finale, reference, velocity for each of the antennas.

3.2 Dynamic handling of the virtual chain

As mentioned before, in order to guarantee the proper connection between the agent and the base station, the antennas dynamically self-organize to realize a *virtual chain.* This configuration is obtained using the follow algorithm:

1 from the base station k_0 look for the closest antenna inside $\mathcal{I}_0$ and label it as the first in the virtual chain k_1; make k_1 the current antenna;

2 from the current antenna in the virtual chain k_j look for the closest antenna inside $\mathcal{I}_{k_j}$ in the remaining antennas and label it as the successive in the virtual chain k_{j+1}; make k_{j+1} the current antenna;

3 repeat the point **2** increasing the index of the current antenna until there is no remaining antenna; if $j+1 = n$ go to step **5**, else keep the virtual chain formed at t_{k-1}, i.e., $\boldsymbol{k}(t_k) := \boldsymbol{k}(t_{k-1})$

4 virtually connect the agent to its closest antenna (not necessarily k_n).

It is *algoritmically* proven that, with the implemented kinematic control, starting from a connected virtual chain and keeping this chain for all the mission, the connection is never broken. In fact, if the antenna in position k_j-th is moving outside the comfort area of the k_{j-1}-th antenna the associated velocity component is immediately nullified because in (8) it is projected in to the null of pseudo-inverse associated with the connection task function (16).

It is worth noticing that the simplest case is given by a *static* virtual chain, i.e., the order of the antennas is never changed during a mission.

4 Simulation Results

The proposed algorithm aimed at guaranteing the wireless communication bridge between the agent and the base station by means of several mobile antennas has been extensively tested in numerical simulations. In all the simulations run the agent is a robot, i.e, it performs its missions independently from the antennas, i.e., the algorithm acquires its trajectory in real-time. Moreover, the robot's movements are coordinated with the MANET so that it can eventually receive *stop* to allow reconfiguration of the antennas. Notice that, as explained above, this behavior is done autonomously by considering the proposed kinematic control and not explicitly commanded to the robot. In this paper two case studies will be reported corresponding to two different environments. For the simulations reported the following parameters have been used

$$\begin{aligned} r_{\max,i} &= 24\,\mathrm{m} \\ d_{\max,i} &= 16\,\mathrm{m} \\ d_{\min,i} &= 6\,\mathrm{m} \\ T &= 300\,\mathrm{ms} \end{aligned}$$

while the maximum velocities are 1 m/s for the robot and 1.5 m/s for the antennas. All the CLIK gains have been set to 0.5. It can be noticed that $d_{\max,i}$ has been kept constant $\forall i$. In both case studies the robot moves in an environment with obstacles and it is required that it stays at a minimum distance of 10 m from them.

In the first scenario the robot has to reach a goal position in presence of two line-shaped obstacles in the environment. The simulation starts with the 7 antennas randomly distributed around the base station following an uniform distribution centered in position $[0\ 0]^{\mathrm{T}}$ m with an interval of 50 m on the coordinates and verifying that a virtual chain is formed and the robot in position $[10\ 10]^{\mathrm{T}}$ m. The robot has to reach a goal in position $[0\ 110]^{\mathrm{T}}$ m. The antennas have to ensure the communication between the robot and the base station avoiding collisions with walls, robot, base station and among themselves. Figure 4 reports six snapshots of the mission execution; the robot moves to the final goal in about 120 s and the antennas do ensure the communication coverage during all the motion. Notice that the virtual serial chain among the antennas can dynamically change during the mission, giving more flexibility to the approach; moreover, differently from [5], this implies that there is no need to impose an explicit order to the antennas.

The second scenario simulates a navigation in a building with several rooms. The robot has to reach different goals passing trough predefined via-points and, obviously, avoiding the walls. The simulation starts with the 10 antennas randomly distributed around the base station following an uniform distribution centered in position $[0\ 0]^{\mathrm{T}}$ m with an interval of 50 m on the coordinates and verifying that a virtual chain is formed and the robot in the initial position $[10\ 10]^{\mathrm{T}}$ m. Figure 5 reports several snapshots of the mission execution. The antennas do maintain the coverage with the base station during all the mission and avoid the obstacles themselves. During the simulation, the robot sometimes can reach the boundary of the area covered by the nearest antenna; in these cases the robot stops and waits for reconfiguration of the network.

5 Conclusions

In this paper we have proposed an algorithm to generate coordinated motion of a platoon of mobile antennas for dynamic reconfiguration of the coverage area of a mobile ad-hoc networks to ensure a communication link between an autonomous agent, that moves independently from them, and a fixed base station.

The algorithm is based on the singularity-robust task-priority inverse kinematics approach that can simultaneously handle multiple tasks. Remarkably, the configuration of the communication chain and the relative priority of the tasks can be dynamically changed during the mission execution for increased effectiveness of the coverage.

In this preliminary simulation study, two different scenarios have been considered. Future work will be devoted at considering coverage of more than one moving agent, occurrence of faults in the antennas, and non-chained link topologies.

References

[1] G. Antonelli, F. Arrichiello, S. Chiaverini, and R. Setola. A self-configuring MANET for coverage area adaptation through kinematic control of a platoon of mobile robots. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1307–1312, Edmonton, CA, Aug. 2005.

[2] G. Antonelli and S. Chiaverini. Kinematic control of a platoon of autonomous vehicles. In *Proceedings 2003 IEEE International Conference on Robotics and Automation*, pages 1464–1469, Taipei, TW, Sept. 2003.

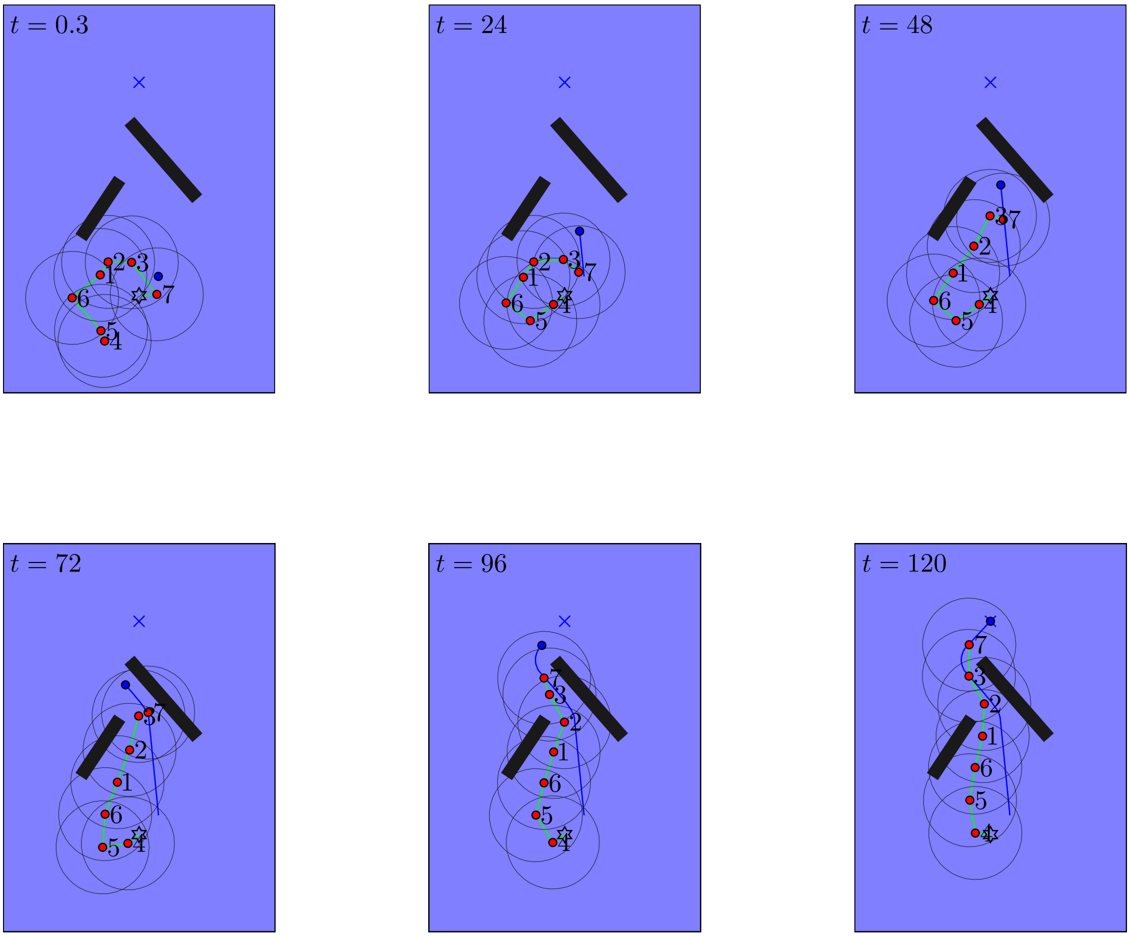

Figure 4: Snapshots of a coverage mission for a platoon of mobile antennas in presence of 2 linear-shaped obstacles. The robot (blue dot) has to reach the goal (marked by a $\times$), while the antennas (red dots) have to ensure connection between the robot and the fixed station (the star).

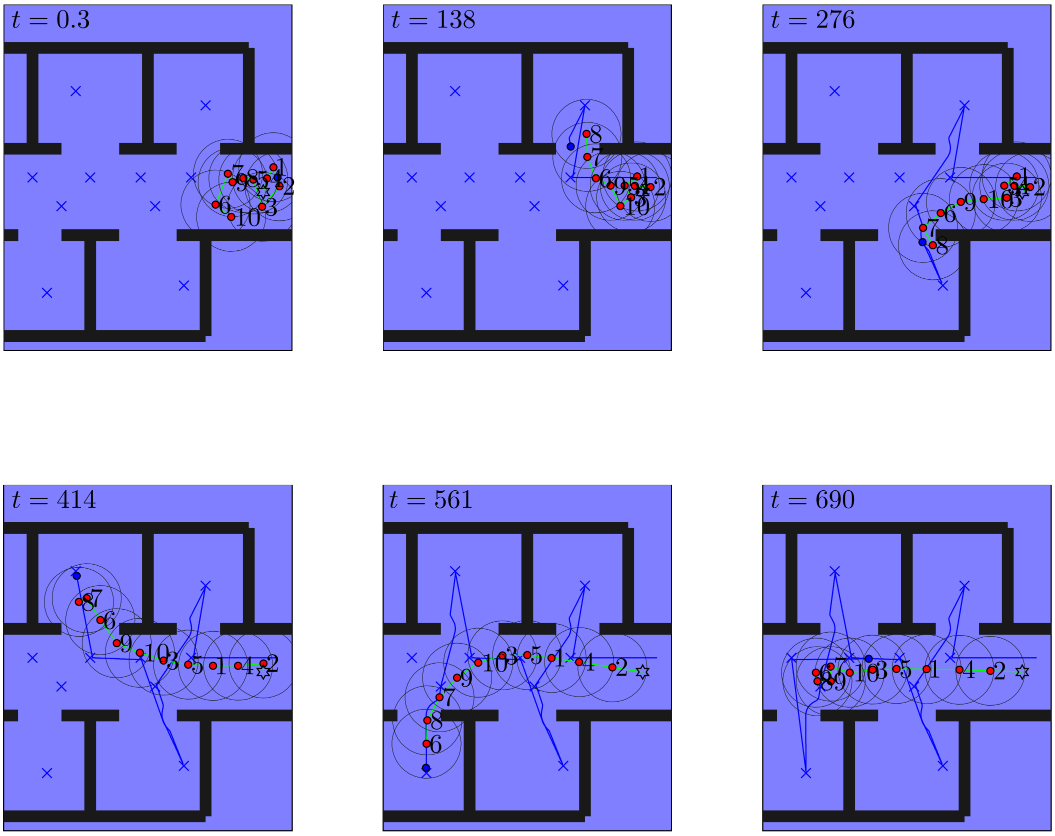

Figure 5: Snapshots of a coverage mission for a platoon of mobile antennas in a complex environment. The robot (blue dot) has to reach the several goals (marked by a $\times$), while the antennas (red dots) have to ensure connection between the robot and the fixed station (the star).

[3] G. Antonelli and S. Chiaverini. Obstacle avoidance for a platoon of autonomous underwater vehicles. In *Proceedings 6th IFAC Conference on Manoeuvring and Control of Marine Craft*, pages 143–148, Girona, SP, Sept. 2003.

[4] G. Antonelli and S. Chiaverini. Fault tolerant kinematic control of platoons of autonomous vehicles. In *Proceedings 2004 IEEE International Conference on Robotics and Automation*, pages 3313–3318, New Orleans, LA, April 2004.

[5] P. Basu and J. Redi. Movement control algorithms for realization of fault-tolerant ad hoc robot networks. *IEEE Network*, 18(4):36–44, 2004.

[6] M.A. Batalin and G.S. Sukhatme. Coverage, exploration and deployment by a mobile robot and communication network. *Telecommunication Systems – Special Issue on Wireless Sensor Networks*, 26(2):181–196, 2004.

[7] B.E. Bishop and D.J. Stilwell. On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles. In *Proceedings 2001 IEEE International Conference on Control Applications*, pages 823–828, México City, MX, Sept. 2001.

[8] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[9] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics an Automation*, 20(2):243–255, 2004.

[10] A. Das, J. Spletzer, V. Cumar, and C. Taylor. Ad hoc network for localization and control. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 2978–2983, Las Vegas, NE, Dec. 2002.

[11] A.A. Maciejewski. Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Robotic Systems*, 5(6):527–552, 1988.

[12] S. Martínez, J. Cortés, and F. Bullo. On robust rendezvous for mobile autonomous agents. In *16th IFAC World Congress*, Praha, CZ, July 2005.

[13] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.

[14] B. Pimentel and M. Campos. Cooperative communication in ad hoc networked mobile robots. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2876–2881, Las Vegas, NE, Oct. 2003.

[15] P.E. Rybski, N.P. Papanikolopoulos, S.A. Stoter, D.G. Krantz, K.B. Yein, M. Gini, R. Voyles, D.F. Hougen, B. Nelson, and M.D. Ericksn. Enlisting rangers and scouts for reconnaissance and surveillance. *IEEE Robotics & Automation Magazine*, 7(4):14–24, 2000.

[16] G.T. Sibley, M.H. Rahmi, and G.S. Sukhatme. Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1143–1148, Washington, DC, May 2002.

[17] B. Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent Robotic Systems*, 3:201–212, 1990.

[18] J. Tan, O.M. Lozano, N. Xi, and W. Sheng. Multiple vehicle systems for sensor network area coverage. In *Proceedings 5th World Congress on Intelligent Control and Automation*, pages 4666–4670, Hangzhou, PRC, June 2004.

[19] P. Ulam and R.C. Arkin. When good comms go bad: Communications recovery for multi-robot teams. In *Proceedings 2004 IEEE International Conference on Robotics and Automation*, pages 3727–3734, New Orleans, LA, April 2004.

[20] J. Vazquez and C. Malcolm. Distributed multirobot exploration maintaining a mobile netwok. In *Proceedings 2nd IEEE International Conference on Intelligent Systems*, volume 3, pages 113–118, Varna, Bulgaria, June 2004.

[21] Z. Wang, M. Zhou, and N. Ansari. Ad-hoc robot wireless communication. In *Proceedings IEEE Conference on Systems, Man and Cybernetics*, pages 4045–4050, Washington, DC, Oct. 2003.