Mobile Heterogeneous Sensor Network for Prioritized Sensing

R. Andres Cortez[a], Rafael Fierro[b], and John Wood[c]
[a] Los Alamos National Laboratory, Los Alamos, NM, USA
Email: racortez@lanl.gov
[b] Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA
Email: rfierro@ece.unm.edu
[c] Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM, USA
Email: jw@unm.edu

***Abstract*—Currently in the literature there does not exist a framework which incorporates a heterogeneous team of agents to solve the sensor network connectivity problem. An approach that makes use of a heterogeneous team of agents has several advantages when cost, integration of capabilities, or possible large search areas need to be investigated. A heterogeneous team allows for the robots to become "specialized" in their abilities and therefore accomplish sub-goals more efficiently which in turn makes the overall mission more efficient.**

In this paper we relax the assumption of network connectivity within the sensor network and introduce mobile communication relays to the network. This addition converts the homogeneous sensor network to a heterogeneous one. Based on the communication geometry of both sensing and communication relay agents we derive communication constraints within the network that guarantee network connectivity. We then define a heterogeneous proximity graph that encodes the communication links that exist within the heterogeneous network. By specifying particular edge weights in the proximity graph, we provide a technique for biasing particular connections within the heterogenous sensor network. Through a minimal spanning tree approach, we show how to minimize communication links within the network which allows for larger feasible motion sets of the sensing agents that guarantee the network remains connected. We also provide an algorithm that allows for adding communication links to the minimal spanning tree of the heterogeneous proximity graph to create a biconnected graph that is robust to a single node failure. We then combine a prioritized search algorithm and the communication constraints to provide a decentralized prioritized sensing control algorithm for a heterogenous sensor network that maintains network connectivity.

I. Introduction

The goal of this paper is to develop a framework that can guarantee connectivity in a group of heterogeneous agents whose mission objective is a prioritized search of an area. Such a framework would help to overcome the limitations imposed by a homogeneous team of agents trying to accomplish the same mission.

A cyber-physical system (CPS) is a network of physically distributed sensors and actuators capable of computation, communication, and control that relies highly on the integration of these capabilities for its operation and interaction with the physical environment in which it is deployed. Our solution, which involves the development of a framework for a heterogeneous sensor network, can been viewed as a cyber-physical system.

We envision a CPS to address this problem as a group of autonomous agents, possibly both ground and aerial vehicles, that are equipped with environmental sensing capabilities, a communication network, as well as having the capability of receiving control inputs. Through the interaction and coordination of the autonomous agents, a connected network topology can be maintained which is critical to accomplishing the mission objective. A visual representation of such a CPS can be seen in Figure 1.

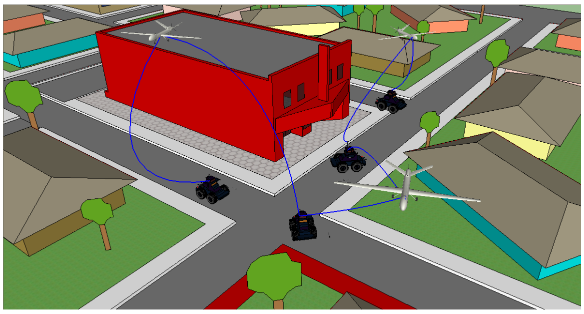

Fig. 1. Envisioned CPS with both ground and aerial vehicles used for obtaining situational awareness in an emergency situation. Communication links between aerial and ground vehicles enables coordination as well as the ability to relay real-time environmental information to an end-user.

A. Related Work

Robotic motion planning is a well-addressed issue in autonomous systems, [1]. However a growing number of applications such as spatial distribution mapping, dynamic sensing coverage, and dynamic target detection, have motivated navigation and control algorithms for teams of goal-oriented mobile sensor networks. When considering control and coordination algorithms of reconfigurable sensor networks one must join the coordination/navigation of the robots with the sensing cost or desired configurations of the sensor network. In problems involving reconfig-

This work was supported by NSF grants ECCS #1027775 and IIS #0812338, and by the Department of Energy URPR Grant: DE-FG52-04NA25590 issued to the Manufacturing Engineering Program.

doi: 10.12720/jcm.8.1.16-31

urable sensor networks, a primary goal is to reconfigure the sensor network in such a way that the time taken to reconfigure is minimized or the sensing coverage is maximized. This has useful applications in target detection and surveillance as well as spatial distribution mapping, among many others.

Recently in the literature, connectivity maintenance has been considered as a constraint on the reconfigurable sensor network. The constraint of maintaining connectivity between sensor nodes is a relaxation to the typically assumed fixed communication topology. The connectivity constraint complicates the motion planning problem for the reconfigurable sensor network in the sense that sensors should only move to areas in the search space where communication can be guaranteed. Typically the connectivity constraint is directly imposed on the sensor network, which may greatly limit its ability to investigate the search space. To overcome this constraint on the reconfigurable sensor network we propose to add mobile communication "relay" agents to the communication topology. This allows the reconfigurable sensor network to have a "longer reach" to investigate the search space, with the added difficulty of having to control a heterogeneous team of robots (sensors and relays).

Research in multi-robot coordination typically assumes that the underlying communication topology is fixed and connected, [2], [3]. Recently however, research has begun to focus on a relaxation of this assumption, namely considering the connectivity of the multi-robot group as being a dynamic topology which should maintain some connectivity properties. In [4], Dimarogonas and Johansson present a distributed control law that guarantees connectivity maintenance in a network of multiple mobile agents. The control law is achieved through a potential field approach with guaranteed boundedness on the agents input. Muhammad *et al.,* [5] derive graph processes to pre-plan for formation tasks, taking into account the graph connectivity. In [6], abstractions are used to enable multiple groups of agents to form desired formations when communication between these groups is limited due to high bandwidth cost. Michael *et al.,* [7] implement a control algorithm that is based on a consensus approach and market based auctions [8] on a group of seven mobile robots. The local connectivity of the group is estimated by computing the second smallest eigenvalue of the graph Laplacian, similar to the work in Kim and Mesbahi [9]. In [10], Ji and Egerstedt address maintaining connectivity in rendezvous and formation control problems. Spanos and Murray [11] derive a function that measures local connectedness of the communication network. This function also provides a sufficient condition for global connectedness of the communication network. Fink and Kumar [12] explore methods for online mapping of Received Signal Strength Indicator (RSSI) with mobile robots where the RSSI map can then be used for control algorithms requiring inter-robot communications. Tekdas *et al.,* [13], [14] study the problem of computing the minimum number of robotic routers in order to maintain connectivity of a single user to a base station. A similar approach is taken by Burdakov *et al.,* [15], where an array of possible relay chains are computed and a user is allowed to choose the one that best fits their needs in terms of cost and number of communication hops.

Works more closely related to the work in this paper can be seen in the following. Reference [16] develops a distributed controller to position a team of UAVs in a configuration that optimizes communication-link quality to support a team of UGVs performing a collaborative task, however the authors must assume the UGVs do not move to guarantee the connectivity of the combined UAV, UGV network. Reference [17] introduces the idea of periodic connectivity where the network must regain connectivity at a fixed interval. The authors propose an implicit coordination algorithm that allows all robots to plan assuming all other robots are stationary, then plans are exchanged to improve performance. It is unclear how many communication rounds are needed to reach a global consensus plan or if this type of algorithm will end up in a deadlock configuration. Reference [18] presents an algorithm that allows a robot to determine when it is feasible for it to move to a desired point by adjusting its own position while maintaining network connectivity. This is achieved by solving a convex optimization problem in an incremental fashion, however the extension to multiple robots moving to multiple desired points is not addressed. In [19], Frew presents an information-theoretic framework to integrate sensing and communication for planning of robot sensor networks. This approach takes uncertainty into account, however it is not clear if any type of connectivity guarantees can be made about the network. Also only static sensing scenarios can currently be addressed in this framework. In [20] the authors derive a flocking controller to regulate the distance between vehicles that address coverage and vehicles that address coordination. The distance requirements for the flocking controller are the communication range of the vehicle types. Stachura and Frew [21] use a fixed planning hierarchy for a finite horizon optimization that addresses cooperative target localization with communication considerations. To determine trajectories for the sensor network the first sensor plans its trajectory over the time horizon then subsequent sensors plan based on the trajectories of the sensors that are higher in the hierarchy.

B. Overview of Proposed Approach

In our approach, we begin by developing a prioritized search algorithm for a homogeneous sensor network where we assume network connectivity is guaranteed over the search area [22]. We then relax the assumption of network connectivity and introduce specialized mobile "relay" agents to the network which are better equipped to communicate over longer distances. We develop communication constraints for the newly formed heterogeneous sensor network comprised of both sensing and relay agents. These constraints lead to feasible motion sets for each sensing agent that can be optimized with respect to the prioritized search and still guarantee network connectivity. Therefore, our approach of controlling a heterogeneous sensor network allows for the unification of prioritized search over an area while maintaining network connectivity.

II. Graph Connectivity

Many definitions given in this section can be found in texts on modern graph theory such as [23]. We begin by considering a heterogeneous team of agents consisting of n sensing agents and m relay agents. For our mathematical formulation we consider each agent x_i to have the following dynamics:

$$\dot{x}_i = Ax_i + Bu_i, \quad (1)$$

where A is the system matrix, B is the input matrix, u_i is the input, and $i = 1, \cdots, n+m$. Without loss of generality, let x_i denote the position of agent i. The network of agents described by the system (1), gives rise to a *dynamic graph* $\mathcal{G}(x)$.

Definition 1: (Dynamic Graph): We Call $\mathcal{G}(x) = (\mathcal{V}, \mathcal{E}(x))$ a dynamic graph consisting of

- a set of vertices $\mathcal{V} = \{v_1, \cdots, v_{n+m},\}$ indexed by the set of agents, and
- a set of edges $\mathcal{E}(x) = \{(i,j) | d_{ij}(x) < \delta\}$, with $d_{ij} = \|x_i - x_j\|_2$ as the Euclidean distance between agents i and j, and $\delta > 0$.

Definition 2: (Path of a Graph): A *path* is a sequence of distinct vertices such that consecutive vertices share a common edge.

Definition 3: (Graph Connectivity): A non-empty graph $\mathcal{G}$ is called *connected* if any two of its vertices are linked by a path in $\mathcal{G}$.

Definition 4: (Tree): A tree is an undirected graph such that any two vertices are connected by exactly one simple path.

Definition 5: (Spanning Tree): Given a non-empty, undirected, and connected graph $\mathcal{G}$ with vertices $\mathcal{V} = \{v_1, \cdots, v_{n+m},\}$, then a spanning tree of $\mathcal{G}$ is a subgraph which is a tree that connects all vertices, $\mathcal{V} = \{v_1, \cdots, v_{n+m},\}$ together.

Definition 6: (Minimum Spanning Tree): A minimum spanning tree (MST) is a spanning tree that has a weight equal to or less than the weight of any other spanning tree. Note that the minimum spanning tree need not be unique.

Definition 7: (Euclidean Minimum Spanning Tree): A Euclidean minimum spanning tree (EMST) is a minimum spanning tree such that the edge weights between vertices are taken to be the Euclidean distance.

Definition 8: (Adjacency Matrix): Given a non-empty graph $\mathcal{G}$ with vertices $\mathcal{V} = \{v_1, \cdots, v_{n+m},\}$ and edges in the set $\mathcal{E}$, we define the adjacency matrix $A = (a_{ij})$ such that, $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise.

Definition 9: (Graph Laplacian): Given a non-empty graph $\mathcal{G}$ with vertices $\mathcal{V} = \{v_1, \cdots, v_{n+m},\}$ and edges in the set $\mathcal{E}$, the graph laplacian is then, $L(x) = D - A$ where D is the valency or degree matrix, $D = \text{diag}(\sum_{j=1}^{n+m} a_{ij})$ and A is the adjacency matrix.

Definition 10: (Algebraic Connectivity): Let $\lambda_1 \leq \ldots \leq \lambda_n$ be the ordered eigenvalues of the Laplacian Matrix $L(x)$, then $\lambda_2 > 0$ if and only if $\mathcal{G}(x)$ is connected. $\lambda_2 > 0$ is also known as the algebraic connectivity of the network.

With these definitions, we can view the dynamic graph induced by the sensor network in a matrix representation. This allows for a straight forward check if the network is connected at any given configuration. Viewing the connectivity of the network from a graph-theoretic point-of-view, the question now becomes how to control the agents (vertices) such that the dynamic graph, $\mathcal{G}(x)$, induced by the agents remains connected throughout the mission.

III. Problem Formulation

We begin by considering a heterogeneous team of agents consisting of n sensing agents, which we will consider for this application as Unmanned Ground Vehicles (UGVs), and m relay agents, which we will assume are Unmanned Aerial Vehicles (UAVs), in two and three dimensions. Assume the n sensing agents are equipped with sensors capable of sensing an environmental phenomena within a finite radius R_s and communicating within a finite radius $R_c(q) \leq R_{c_{max}}$. Here we assume that the communication radius will change based on the positions of the robots. This relaxation in the communication range allows us to model, to some degree, the path loss in the communication channel [24]. Incorporating communication channel characteristics, which has been largely ignored in the literature to date, allows for a better system model. Also let us assume that the m relay agents are capable of communicating over a finite radius R_{rc} such that $R_{rc} > R_{c_{max}}$ *i.e.*, the relay robots are better equipped for communication than the sensing agents and the relay robots communication range is not dependent on location. Consider the area of interest Q, assumed to be a simple convex polygon with boundary ∂Q, including its interior. Define C_{obs} as the union of all obstacles in the region Q, and let $Q_f = Q \setminus C_{obs}$ be the area within Q that is free of obstacles. Let us define the probability of detection map $M(q)$, which reflects the probability of detecting an environmental phenomena over the area to be searched [22].

We are assuming a linear controllable system, equation (1), under the premise that the dynamics from both ground vehicles as well as aerial vehicles with an autopilot system can be conservatively estimated in such a way. This is an abstraction to the real dynamics of both ground and aerial vehicles with the assumption that there exists well tuned low-level controllers for each vehicle type. Here we are considering our sensor network to be heterogeneous not only because relay and sensing agents have different communication ranges but also because they play different roles in the sensor network.

IV. Communication Constraints

In our scenario there exists three particular communication link possibilities. The first being, *relay/sensor* communication, where a sensor communicates directly to a relay agent. The second, *relay/relay* communication, where a relay shares a communication link with another relay agent. The last communication link possibility is *sensor/sensor* communication where sensors communicate directly with each other. For the following formulation let us consider the case where the communication radius of the sensing robots is not location dependent, *i.e.* $R_c(q) = R_c$. Note that each agents communication range describes the range over which the agent can both send and receive information. For ease of notation let us also consider that the relay agents fly at constant altitude, h, and their communication range will be taken as its two dimensional projection at this constant height.

Similar to the work on homogeneous networks of Bullo *et al.*, [25], we now formulate the *connectivity constraint set* for each particular communication link possibility of our heterogeneous network based on the geometry of the communication radii. For the following definitions we will use $\bar{\mathcal{B}}(p,r)$ to denote a closed ball of radius r centered at p in $\mathbb{R}^2$.

Definition 11: (*Relay/Sensor* connectivity constraint set) Consider two agents, one relay agent i located at position p_i and one sensing agent j located at position p_j such that $||p_i - p_j||_2 \leq R_{rc}$. Then the connectivity constraint set of agent i with respect to agent j is

$$\Upsilon_{\mathrm{d}_{rs}}(p_i,p_j) = \bar{\mathcal{B}}(\frac{p_i+p_j}{2}, \frac{R_{rc}}{2}). \quad (2)$$

Definition 12: (*Relay/Relay* connectivity constraint set) Consider two relay agents, one agent i located at position p_i and one agent j located at position p_j such that $\|p_i - p_j\|_2 \leq R_{rc}$. Then the connectivity constraint set of agent i with respect to agent j is

$$\Upsilon_{\mathrm{d}_{rr}}(p_i,p_j) = \bar{\mathcal{B}}(\frac{p_i+p_j}{2}, \frac{R_{rc}}{2}). \quad (3)$$

Definition 13: (*Sensor/Sensor* connectivity constraint set) Consider two sensing agents, one agent i located at position p_i and one agent j located at position p_j such that $\|p_i - p_j\|_2 \leq R_c$. Then the connectivity constraint set of agent i with respect to agent j is

$$\Upsilon_{\mathrm{d}_{ss}}(p_i,p_j) = \bar{\mathcal{B}}(\frac{p_i+p_j}{2}, \frac{R_c}{2}). \quad (4)$$

Remark 1: Notice that definition 11 and definition 12 are the same. This is due to the fact that sensor's communication radius can be ignored when a communication link exists between a sensor and relay, since the sensors communication radius is smaller that that of the relay agent. This is also a consequence of the assumption that the sending and receiving channels are symmetric.

Definition 14: (Connectivity constraint set for relay agent w.r.t. heterogeneous network) Consider a group of agents containing both sensing and relay agents located at $\mathcal{P} = \{p_1, p_2, \ldots, p_{n+m}\}$. Then the connectivity constraint set of relay agent i with respect to all other agents in the group is

$$\Upsilon_{\mathrm{d}_{hr}}(p_i,\mathcal{P}) = \{x \in \Upsilon_{\mathrm{d}_{rr}}(p_i,p_j) | q \in \mathcal{P}\backslash\{p_i\} \text{ s.t. } \|q-p_i\|_2 \leq R_{rc}\}. \quad (5)$$

Figure 2 shows an example of a relay connectivity constraint set w.r.t. the heterogeneous network.

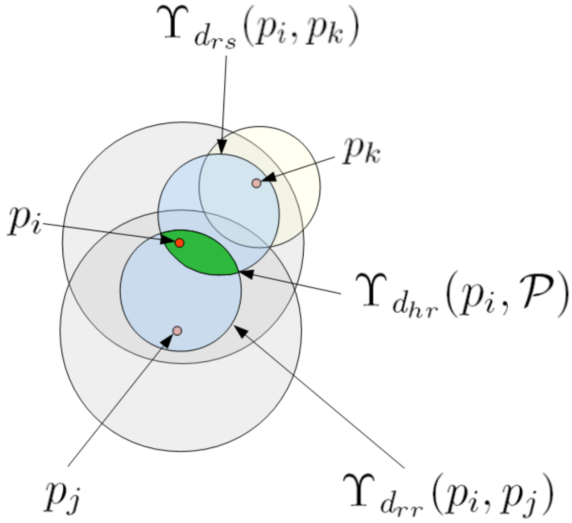

Fig. 2. Motion constraint set for a relay agent w.r.t. the network, $\Upsilon_{\mathrm{d}_{hr}}(p_i,\mathcal{P})$. The green area represents the motion constraint set that guarantees connectivity for relay agent p_i w.r.t. the heterogeneous network.

Before we can state the definition of the connectivity constraint set for a sensing agent with respect to the heterogeneous network we need some preliminaries. Let p_i be a sensing agent, then

$$\Lambda_{ss} = \cap_{j=1}^{n} \Upsilon_{ss}(p_i,p_j), \text{ where } p_j \in \text{ sensors}, \quad (6)$$

$$\Lambda_{sr} = \cap_{k=1}^{m} \Upsilon_{sr}(p_i,p_k), \text{ where } p_k \in \text{ relays}. \quad (7)$$

Now we can define the connectivity constraint set for a sensing agent with respect to the heterogeneous network.

Definition 15: (Connectivity constraint set for sensor agent w.r.t. heterogeneous network) Consider a group of agents containing both sensing and relay agents located at $\mathcal{P} = \{p_1, p_2, \ldots, p_{n+m}\}$. Then the connectivity constraint set of a sensor agent i with respect to all other agents in the group is

$$\Upsilon_{\mathrm{d}_{hs}}(p_i,\mathcal{P}) = \Lambda_{ss} \cap \Lambda_{sr}. \quad (8)$$

Figure 3 shows an example of a sensors connectivity constraint set w.r.t. the heterogeneous network. The connectivity constraint sets defined in (11) - (15) define the set of allowable positions that each robot may take such that the communication network will remain connected. Thus the connectivity constraint sets defines the feasible

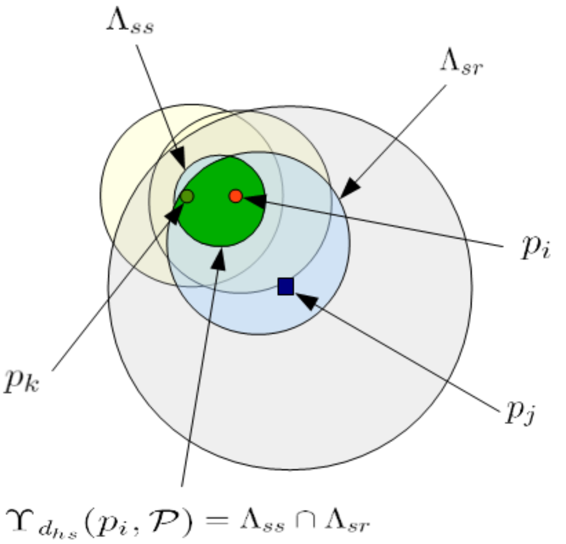

Fig. 3. Motion constraint set for a sensor agent w.r.t. the network, $\Upsilon_{\mathrm{d}_{hs}}(p_i, \mathcal{P})$. The green area represents the motion constraint set that guarantees connectivity for sensor agent p_i w.r.t. the heterogeneous network.

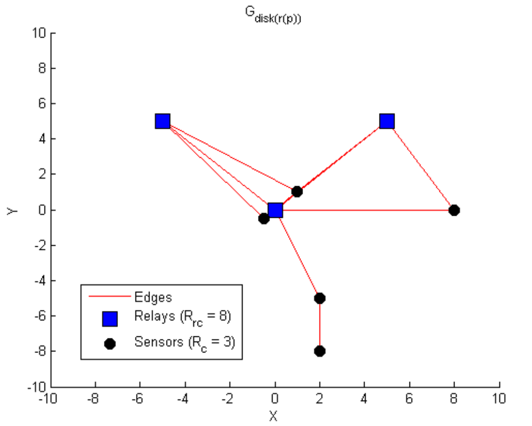

Fig. 4. Example of Heterogeneous r(p)-disk graph, $\mathcal{G}_{\mathrm{disk}(r(p))}(\mathcal{P})$ with three relay robots (blue squares) and five sensing robots (black circles). The red lines represent edges in the graph between respective agents.

motion for each individual robot to remain connected with the network.

V. Heterogeneous Proximity Graph

Due to the heterogeneity of our sensor network, we must define an appropriate proximity graph. As a reminder, a proximity graph describes connections between a set of vertexes based on their relative distances.

Definition 16: (Proximity Graph, [25]) Let $S \subset \mathbb{R}^N$. A proximity graph $\mathcal{G}$ associates to a set of distinct points $\mathcal{P} = \{p_1, \ldots, p_n\} \subset S$, an undirected graph with vertex set $\mathcal{P}$ and whose edge set is given by $\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \{(p, q) \in \mathcal{P} \times \mathcal{P} | p \neq q\}$.

We see that due to the heterogeneity of our network, the edge set of our proximity graph should depend on the agent type. The following definition describes how the edge set should be created for our heterogeneous proximity graph.

Definition 17: (Heterogeneous r(p)-disk graph) Two agents p_i and p_j are neighbors if they are located within a distance $r(p) = R_c$ if both p_i and p_j are sensing agents or $r(p) = R_{rc}$ if one of the agents is a relay agent, i.e.,

$$(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{disk}(r(p))}}(\mathcal{P}) \tag{9}$$

$$\text{if} \begin{cases} \|p_i - p_j\|_2 \leq R_c \text{ and } p_i,\, p_j \text{ both sensing agents} \\ \|p_i - p_j\|_2 \leq R_{rc} \text{ and } p_i \text{ or } p_j \text{ is a relay agent.} \end{cases} \tag{10}$$

An example of the $\mathcal{G}_{\mathrm{disk}(r(p))}(\mathcal{P})$ graph is shown in Figure 4. In the $\mathcal{G}_{\mathrm{disk}(r(p))}(\mathcal{P})$ graph, edges depend on the agent distances as well as agent connection combinations.

The heterogeneous r(p)-disk proximity graph, $\mathcal{G}_{\mathrm{disk}(r(p))}(\mathcal{P})$, allows us to represent the network topology of our heterogeneous system. It is seen that depending on the configuration of the network there may exist heavy redundancy in the connections (Figure 4). This redundancy comes at the cost of more constraints on each agent, therefore reducing the size of the set of possible inputs that guarantee connectivity. This reduction stems from the fact of the intersection of multiple sets.

Let us now define the the weighted complete graph which we will denote as, $\mathcal{G}$ throughout the rest of this paper.

Definition 18: (Weighted Complete Graph, $\mathcal{G}$) Let $S \subset \mathbb{R}^N$. The weighted complete graph $\mathcal{G}$ associates to a set of distinct points $\mathcal{P} = \{p_1, \ldots, p_{n+m}\} \subset S$, an undirected graph with vertex set $\mathcal{P}$ and whose edges $e = (p_i, p_j) \in \mathcal{E}_{\mathcal{G}}(\mathcal{P})$ has the following weights $w(e)$,

$$w(e) = \begin{cases} ||p_i - p_j||_2 + R_{rc} & \text{if } p_i,\, p_j \text{ both sensing agents} \\ ||p_i - p_j||_2 & \text{if } p_i \text{ or } p_j \text{ is a relay agent.} \end{cases} \tag{11}$$

VI. Minimizing Motion Constraints

With a formal way of representing the motion constraints for each agent with respect to the heterogeneous group, we now are left with trying to minimize the constraints (links) in such a way that we expand the input set the agents can choose from that still guarantees connectivity at the next time step. One solution is to take $\mathcal{G}_{\mathrm{disk}(r(p))}(\mathcal{P})$ and run a minimum spanning tree algorithm to determine a subgraph of the r(p)-disk graph that has the minimum number of connections needed to remain connected. A key result from modern graph theory is that assuming $\mathcal{G}_{\mathrm{disk}(r(p))}(\mathcal{P})$ is connected their always exist a minimal spanning tree [26]. The usefulness of the minimum spanning tree approach is that it allows us to weigh connections between agents. This may be useful in enforcing *relay/sensor* connections over *sensor/sensor* connections since *relay/sensor* connections offer a greater motion set for the agents as opposed to the *sensor/sensor* connections because of the larger communication radius.

Another reason to bias certain network connections when possible is because relay nodes are better equipped to handle communication data, *i.e.,* higher bandwidth. Figure 5 shows the Euclidean Minimum Spanning Tree (EMST) for the r(p)-disk graph weighted by the Euclidean distance between connected vertexes, $\mathcal{G}_{EMST,\mathcal{G}}$. It is key to note that the EMST is a subgraph of the $\mathcal{G}_{\text{disk}(r(p))}(\mathcal{P})$ graph and contains the minimal number of connection to maintain a connected graph.

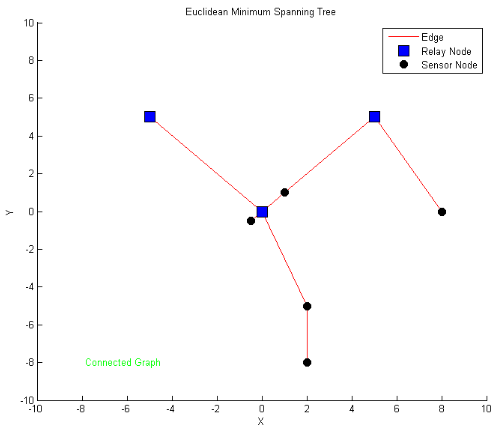

Fig. 5. Example of $\mathcal{G}_{EMST,\mathcal{G}}$ with three relay robots (blue squares) and five sensing robots (black circles). The red lines represent edges in the graph between respective agents. Notice that the graph is connected.

A. Shaping the Network Configuration

To help bias *relay/sensor* connections over *sensor/sensor* connections with respect to the Minimum Spanning Tree (MST) we now formulate a weighting factor for *sensor/sensor* connections. From definitions (11) and (13) we see that the motion constraint set for *relay/sensor* connections is larger than *sensor/sensor* connections due to a larger communication radius. With the help of Figure 6 we look at the scenario of one relay and two sensing agents. In terms of the MST, all connections that have a possibility of being biased can be broken down in this way. For ease of notation we will refer to the $MST_{\mathcal{G}_{\text{disk}(r(p))}}$ as just the MST.

Let $\|p_i - p_j\|_2 = l$, $\|p_i - p_k\|_2 = l_1$ and $\|p_j - p_k\|_2 = l_2$. Let us assume that $l < l_1 \leq R_{rc}$ and $l_2 \leq R_c$. From construction of the MST, the red solid edges between p_i, p_j, and p_k in Figure 6 will be chosen since

$$l + l_2 < l_1 + l_2,$$

$$l + l_2 < l + l_1.$$

Let $\epsilon > 0$ denote the minimum distance between two sensing agents, *i.e.,* physical footprint. Under other circumstance ϵ can also be considered the threshold distance where two sensing agents should communicate directly. To bias the *relay/sensor* connection (red dotted line) a weighting factor ξ_1, must be constructed such that when

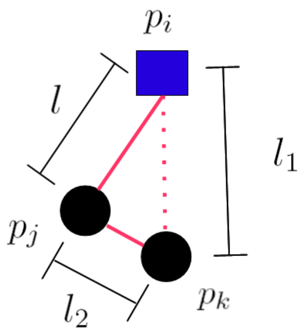

Fig. 6. Figure of one relay agent (blue square) and two sensing agents (black circle) used to formulate weighting factor for *sensor/sensor* connections.

$l_2 = \epsilon$, $\xi_1 l_2 \geq R_{rc}$. Defining $\xi_1 = \left(\frac{R_{rc}}{\epsilon} + \delta_1\right)$ with $\delta_1 \geq 0$ we get the following,

$$\begin{aligned} \xi_1 l_2 &= \left(\frac{R_{rc}}{\epsilon} + \delta_1\right) l_2 \\ \xi_1 l_2 &= R_{rc} + \delta_1 \epsilon \\ \xi_1 l_2 &\geq R_{rc}. \end{aligned} \tag{12}$$

Therefore, with the connection weighting factor ξ_1 we now have the following,

$$l + l_1 \leq l + \xi_1 l_2,$$

$$l + l_1 \leq l_1 + \xi_1 l_2.$$

Weighting the *sensor/sensor* connection (edge) by a factor of ξ_1 allows us to bias the MST to chose the *relay/sensor* connections. Figure 7 shows a connected $\mathcal{G}_{\text{disk}(r(p))}$ graph with many redundant connections. Figure 8 and Figure 9 show the difference in network connections between the MST and the connection weighted MST (MST_{CW}) respectively, where *sensor/sensor* connections are weighted by the factor ξ_1. Notice that in the MST_{CW} graph, the *relay/sensor* connections are chosen over *sensor/sensor* connections.

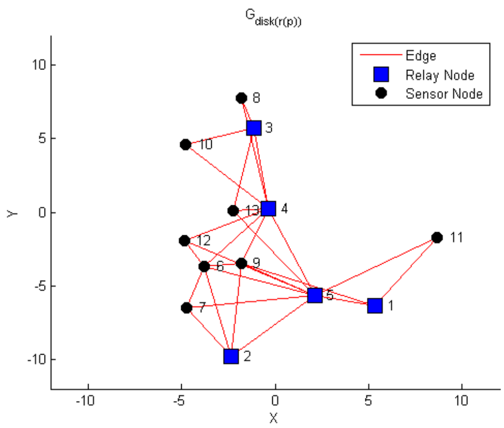

Fig. 7. The $\mathcal{G}_{\text{disk}(r(p))}$ with many redundant connections.

To understand the effect of choosing *relay/sensor* connections over *sensor/sensor* connections, we calculate the

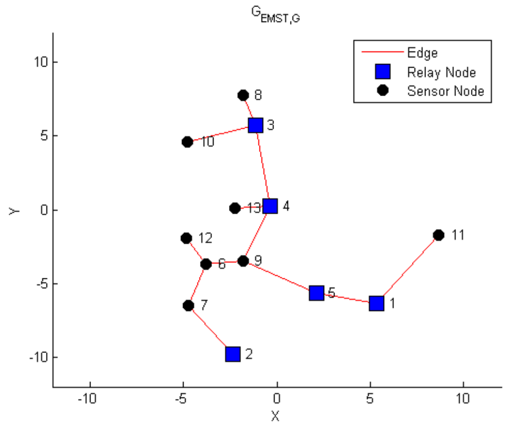

Fig. 8. Example of the MST for thirteen agents with no connection weights.

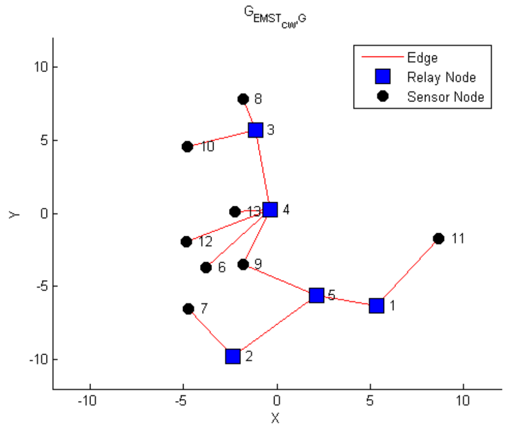

Fig. 9. The MST_{CW} graph for the thirteen agents. Notice how the *relay/sensor* connections are chosen over *sensor/sensor* connections.

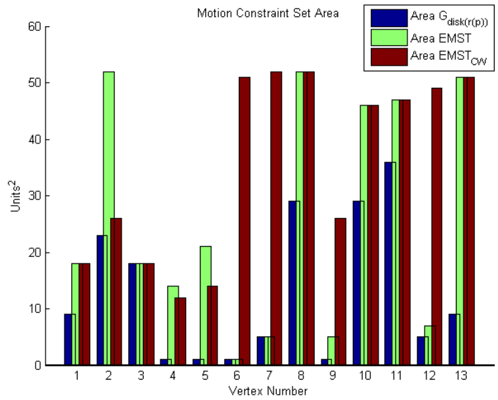

Fig. 10. Comparison of the area of the motion constraint sets for the $\mathcal{G}_{\text{disk}(r(p))}$, MST, and MST_{CW}. Notice that in the MST_{CW} graph the motion constrain set area is higher for sensors (vertexes 6-13) than in the MST graph.

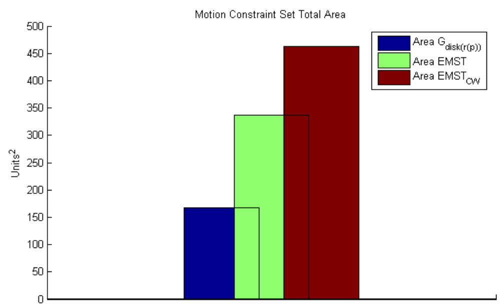

Fig. 11. Figure showing the total area covered by the motion constraint sets by the three different graph representations.

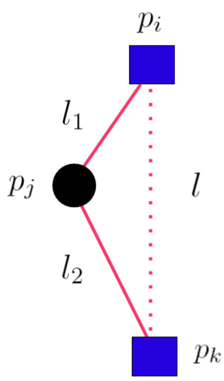

Fig. 12. Figure of two relay agents (blue squares) and one sensing agent (black circle) used to formulate weighting factor for *relay/relay* connections.

area covered by the motion sets for the various graph representations. Figure 10 shows the difference in the area of the motion constraint sets for the $\mathcal{G}_{\text{disk}(r(p))}$, $\mathcal{G}_{MST}$, and the $\mathcal{G}_{MST_{CW}}$ graph for each agent in the network. Notice that $\mathcal{G}_{MST_{CW}}$ graph allows for the largest motion constraint set area for sensing agents (vertex 6-13).

We sum the total areas of the feasible motion sets for each graph representation of the network in Figure 8 and Figure 9, and see that the $\mathcal{G}_{\text{disk}(r(p))}$ totalled 167 units^2, the MST totalled 337 units^2, and the MST_{CW} totalled 462 units^2. This gives us a good indication that the MST_{CW} graph "frees" up more area for the sensing agents to investigate than the other network graph representations. This is attributed to the fact that relay agents have a larger communication radius.

In a similar fashion we can bias *relay/relay* connections. This may be advantageous for certain mission objectives or when large amounts of data may need to be transferred directly to a relay node. It may not be efficient or even possible to send large amounts of data through a sensing node to reach another relay node.

Using Figure 12, as previously stated let us assume the minimum distance between any two agents is $\epsilon > 0$. Let us also assume that from Figure 12 that $l, l_1, l_2 < R_{rc}$ and for convenience assume $l_1 < l_2 < l$. From the point of view of the MST the red edges between p_i, p_j, and p_k in Figure 12 will be chosen since

$$l_1 + l_2 < l_1 + l,$$

$$l_1 + l_2 < l + l_2.$$

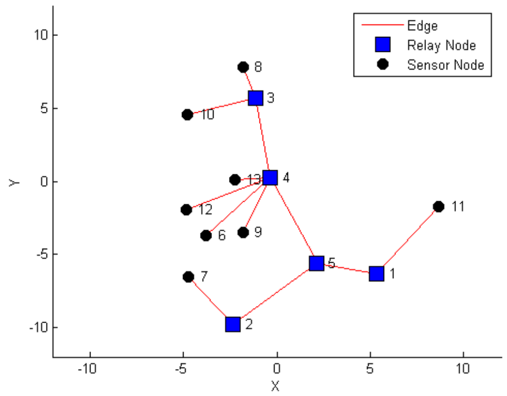

Fig. 13. Example MST_{CW} for thirteen agents with both *sensor/sensor* and *relay/relay* connection weights. Notice the *relay/sensor* connections are chosen over *sensor/sensor* connections and *relay/relay* connections are chosen.

To bias direct *relay/relay* connections (Figure 12 red dotted line), we use a weighting factor $\xi_2 = \frac{\epsilon}{l}$. Choosing ξ_2 in this way insures that a direct *relay/relay* connection will be chosen over the multi-hop connection by the MST algorithm in Figure 12, *i.e.*, relay$\rightarrow$ sensor$\rightarrow$ relay. This is seen from the fact that,

$$\xi_2 l = \frac{\epsilon}{l} l,$$

$$\xi_2 l = \epsilon.$$

Therefore, now the distance between p_i and p_k is ϵ from the point of view of the MST algorithm. Since the minimum distance of any two agents is ϵ the MST will choose the direct *relay/relay* link. Figure 13 shows the network configuration using both ξ_1 and ξ_2 as connection weights (MST_{CW}).

1) Distributed Minimum Spanning Tree: In a classic paper by Gallager *et al.* [27], it was shown that there exists a distributed algorithm to compute the minimum weight spanning tree of a connected, undirected graph with $\mathcal{N}$ nodes and $\mathcal{E}$ edges. It was also shown that at most $5\mathcal{N}\log_2\mathcal{N} + 2\mathcal{E}$ messages need to be passed to determine the minimum weight spanning tree. Much work has focused on improving the time complexity of this algorithm and an approximate MST can be calculated in almost optimal time [28].

These key results allow the robots to compute the MST of the heterogeneous proximity graph with only local information from adjacent robots. This is very useful because it also allows the feasible motion constraint sets to be calculated in a distributed fashion. For a summary of these distributed algorithms the reader is referred to [27] and [28].

VII. Properties of the Heterogeneous Motion Constraints

This section details some properties of the heterogeneous motion constraint for agents described in (1).

Theorem 2: Consider a relay agent, p_i, with dynamics described in (1), such that (1) is at least stabilizable and having a motion constraint set as defined in (11). If p_i takes a goal point $g_{p_i} \in \Upsilon_{\text{d}_{rs}}(p_i, p_j)$ at time t_1, then p_i will be connected to p_j when it reaches g_{p_i} at time t_2.

Proof: Given the fact that the dynamics of p_i are at least stabilizable implies that there exists a static control law $u(t) = -Kx(t)$ such that the closed loop system is asymptotically stable, *i.e.*, $\lim_{t\to\infty} x(t) = g_{p_i}$.

By definition $g_{p_i} \in \Upsilon_{\text{d}_{rs}}(p_i, p_j)$ which implies that $\|g_{p_i} - p_j\|_2 < R_{rc}$, hence p_i at position g_{p_i} at time time t_2 is connected with p_j. ∎

Theorem 3: Consider a relay agent, p_i, with dynamics described in (1), such that (1) is at least stabilizable and having a motion constraint set as defined in (12). If p_i takes a goal point $g_{p_i} \in \Upsilon_{\text{d}_{rr}}(p_i, p_j)$, at time t_1, then p_i will be connected to p_j when it reaches g_{p_i} at time t_2.

Proof: The proof is similar to the proof in Theorem 2. ∎

Theorem 4: Consider a sensing agent, p_i, with dynamics described in (1), such that (1) is at least stabilizable and having a motion constraint set as defined in (13). If p_i takes a goal point $g_{p_i} \in \Upsilon_{\text{d}_{ss}}(p_i, p_j)$, at time t_1 then p_i will be connected with p_j when it reaches g_{p_i} at time t_2.

Proof: The proof is similar to the proof in Theorem 2. ∎

Theorem 5: Consider a relay agent, p_i, with dynamics described in (1), such that (1) is at least stabilizable and having a motion constraint set as defined in (14). If p_i takes a goal point $g_{p_i} \in \Upsilon_{\text{d}_{hr}}(p_i, \mathcal{P})$ at time t_1, then p_i will be connected with all agents at time t_2 that it was connected with at t_1 when it reaches g_{p_i}.

Proof: Given the fact that the dynamics of p_i are at least stabilizable implies that there exists a static control law $u(t) = -Kx(t)$ such that the closed loop system is asymptotically stable, *i.e.*, $\lim_{t\to\infty} x(t) = g_{p_i}$.

By definition $g_{p_i} \in \Upsilon_{\text{d}_{hr}}(p_i, \mathcal{P})$ which implies that

$$\forall q \in \mathcal{P} \backslash \{p_i\} s.t. \|q - g_{p_i}\|_2 < R_{rc},$$

hence p_i at position g_{p_i} at time time t_2 is connected with all $p_j \in \mathcal{P}$ that it was connected to at time t_1. ∎

Theorem 6: Consider a sensing agent, p_i, with dynamics described in (1), such that (1) is at least stabilizable and having a motion constraint set as defined in (15). If p_i takes a goal point $g_{p_i} \in \Upsilon_{\text{d}_{hs}}(p_i, \mathcal{P})$ at time t_1, then p_i will be connected with all agents at time t_2 that it was connected with at t_1 when it reaches g_{p_i}.

Proof: Given the fact that the dynamics of p_i are at least stabilizable implies that there exists a static control law $u(t) = -Kx(t)$ such that the closed loop system is asymptotically stable, *i.e.* $\lim_{t\to\infty} x(t) = g_{p_i}$.

By definition $g_{p_i} \in \Upsilon_{\text{d}_{hs}}(p_i, \mathcal{P})$ which implies that

$$\forall q \in \mathcal{P} \backslash \{p_i\} s.t. \|q - g_{p_i}\|_2 < R_{rc},$$

if q is a relay agent and

$$\forall q \in \mathcal{P} \backslash \{p_i\} s.t. \|q - g_{p_i}\|_2 < R_c,$$

if q is a sensing agent. Hence p_i at position g_{p_i} at time time t_2 is connected with all $p_j \in \mathcal{P}$ that it was connected to at time t_1. ■

Theorem 7: For $R_{rc}, R_c \in \mathbb{R}^+$ and $R_c < R_{rc}$, then $\mathcal{G}_{\text{MST},\mathcal{G}} \subset \mathcal{G}_{\text{disk}(r(p))}$ if and only if $\mathcal{G}_{\text{disk}(r(p))}$ is connected, where $\mathcal{G}$ is the weighted complete graph described in definition 18 with vertex set $\mathcal{P} = \{p_1, p_2, \ldots, p_{n+m}\}$.

Proof: ($\Rightarrow$) If $\mathcal{G}_{\text{MST},\mathcal{G}} \subseteq \mathcal{G}_{\text{disk}(r(p))}$ then $\mathcal{G}_{\text{disk}(r(p))}$ is connected by definition of $\mathcal{G}_{\text{MST},\mathcal{G}}$, *i.e.,* MST is connected.

($\Leftarrow$) (By Contradiction) Assuming $\mathcal{G}_{\text{disk}(r(p))}$ is connected but $\mathcal{G}_{\text{MST},\mathcal{G}} \not\subseteq \mathcal{G}_{\text{disk}(r(p))}$ implies two possible scenarios, (i) there exists two vertices $p_i, p_j \in \mathcal{E}_{\mathcal{G}_{\text{MST},\mathcal{G}}}$ such that $||p_i - p_j||_2 > R_c$ where p_i, p_j are both sensing agents or, (ii) there exists two vertices $p_i, p_j \in \mathcal{E}_{\mathcal{G}_{\text{MST},\mathcal{G}}}$ such that $||p_i - p_j||_2 > R_{rc}$ where p_i or p_j is a relay agent.

(i) If we remove the edge linking p_i and p_j from $\mathcal{E}_{\mathcal{G}_{\text{MST},\mathcal{G}}}$ then the tree becomes disconnected with two connected components, T_1 and T_2 such that $p_i \in T_1$ and $p_j \in T_2$. Since by assumption $\mathcal{G}_{\text{disk}(r(p))}$ is connected, there must exist $p_k, p_l \in \mathcal{P}$ such that $p_k \in T_1$, $p_l \in T_2$ and $||p_k - p_l||_2 < R_c$ if both p_k and p_l are sensing agents or $||p_k - p_l||_2 < R_{rc}$ if either p_k or p_l is a relay agent. If we add the edge (p_k, p_l) to the set of edges of $T_1 \cup T_2$, then the resulting graph $\mathcal{G}^*$ is acyclic, connected, and contains all vertices in $\mathcal{P}$. This implies $\mathcal{G}^*$ is a spanning tree. Since $||p_k - p_l||_2 \leq R_c < ||p_i - p_j||_2$ we can conclude that $\mathcal{G}^*$ has a smaller length than $\mathcal{E}_{\mathcal{G}_{\text{MST},\mathcal{G}}}$ from the definition of the edge weights of the complete weighted graph $\mathcal{G}$ in definition 18. This is a contradiction of the definition of the MST.

(ii) Under the same argument as (i) and replacing R_c with $R_r c$ it can be shown similarly that the result is a contradiction of the MST. ■

For the $\mathcal{G}_{\text{disk}(r(p))}$ graph we take the edge weights between two connected vertices to be Euclidean distance between the two agents represented by the nodes in the graph. Therefore the MST for the $\mathcal{G}_{\text{disk}(r(p))}$ graph becomes the EMST.

Theorem 8: For $R_{rc}, R_c \in \mathbb{R}^+$ and $R_c < R_{rc}$, if $\mathcal{G}_{\text{disk}(r(p))}$ is connected then,

$$\sum_{e \in \mathcal{G}_{\text{MST},\mathcal{G}_{\text{disk}(r(p))}}} w(e) \leq \sum_{e \in \mathcal{G}_{\text{MST},\mathcal{G}}} w(e).$$

Proof: From theorem 7 we have that $\mathcal{G}_{\text{MST},\mathcal{G}} \subset \mathcal{G}_{\text{disk}(r(p))}$ which implies that $\mathcal{E}_{\mathcal{G}_{\text{MST},\mathcal{G}}} \in \mathcal{E}_{\mathcal{G}_{\text{disk}(r(p))}}$. We also know that by definition $\mathcal{E}_{\mathcal{G}_{\text{MST},\mathcal{G}_{\text{disk}(r(p))}}} \in \mathcal{E}_{\mathcal{G}_{\text{disk}(r(p))}}$. Looking at the edge weights of $\mathcal{G}_{\text{MST},\mathcal{G}_{\text{disk}(r(p))}}$ (Euclidean distance) we have,

$$w(e) = \begin{cases} ||p_i - p_j||_2 \leq R_c & \text{if } p_i,\, p_j \text{ both sensing agents} \\ ||p_i - p_j||_2 \leq R_{rc} & \text{if } p_i \text{ or } p_j \text{ is a relay agent.} \end{cases}$$

For the edge weights of $\mathcal{G}_{\text{MST},\mathcal{G}}$ described in definition 18 we see that,

$$w(e) = \begin{cases} ||p_i - p_j||_2 + R_{rc} & \text{if } p_i,\, p_j \text{ both sensing agents} \\ ||p_i - p_j||_2 & \text{if } p_i \text{ or } p_j \text{ is a relay agent.} \end{cases}$$

Since $R_{rc} > 0$ by definition, the edge weights of all possible links chosen by the MST algorithm for the $\mathcal{G}_{\text{MST},\mathcal{G}}$ graph will be greater than or equal to those chosen for the $\mathcal{G}_{\text{MST},\mathcal{G}_{\text{disk}(r(p))}}$ graph. ■

Remark 9: In a centralized scenario or when the $\mathcal{G}_{\text{disk}(r(p))}$ graph is the complete graph, theorem 8 provides a straightforward method for checking whether the MST of the $\mathcal{G}_{\text{disk}(r(p))}$ was computed correctly.

VIII. Case Study: Centroidal Heterogeneous Motion Constraint Set Configurations

This section looks at the particular situation when the heterogeneous team moves towards their respective centroid of their feasible motion set. This centroidal configuration is a straightforward way to test the claims of Theorem 5 and Theorem 6. It is key to note that the centroidal configuration is just one of many different possible ways of testing the claims of Theorem 5 and Theorem 6.

A. Simulations

By construction, the centroid $\mathcal{C}_i^*$ of each agents motion constraint set (MCS_i) lives in the interior of its motion constraint set. Therefore by setting $\mathcal{C}_i^*$ as the goal point for each agent $i, \forall i = 1, \ldots, n+m$, then the heterogeneous team should remain connected when each goal point is reached by the respective agents. Figure 14 depicts the centroidal heterogeneous motion constrain set configuration for two relay agents and one sensing agent. The red stars denote the centroid of each agent's constraint set.

Algorithm 1 Centroidal Behavior ($g_{p_i} = \mathcal{C}_i^*$)

```
while t < t_final do
    for x_i = 1, ..., n + m do
        Calculate C_i^* from MCS_i (Equations (2)-(8))
        g_{p_i} ⇐ C_i^*
        while Δt < T_s do
            u_i(t) = −K x_i(t)
        end while
    end for
end while
```

For this simulation we set $R_c = 3$m, and $R_{rc} = 10$m. We use $m = 4$ relay agents and $n = 10$ sensing agents initially in a random configuration but in such a way that the heterogeneous team is initially connected. Each agent $i, \forall i = 1, \ldots, n+m$, then uses its neighbors of the $\mathcal{G}_{MST_{CW}}$ graph to calculate the centroid of their respective motion constraint set. Each agent is modeled as a double integrator (1), and a state feedback control law is used to drive the agents from their current position to their respective goal points ($\mathcal{C}_i^*$). Every $T_s = 0.05$ seconds, position information is exchanged among the team members and an updated $\mathcal{G}_{MST_{CW}}$ graph is calculated. Based on the new information, new centroids are updated and used as the goal point. An outline is given in Algorithm 1. The simulation lasts for a total of five seconds.

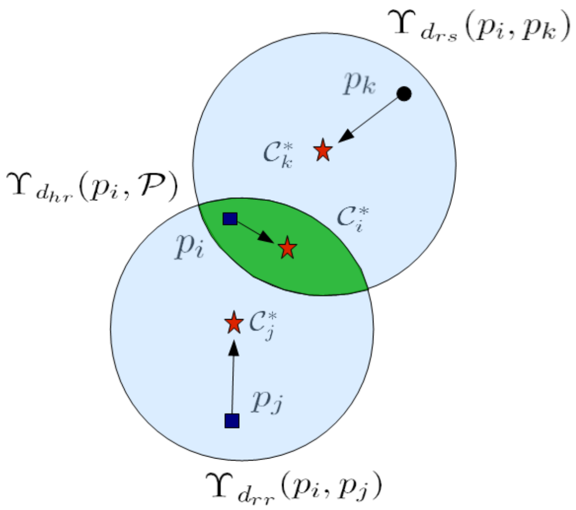

Fig. 14. Figure showing the centroidal heterogeneous motion constraint set configurations. Each respective agent calculates its own centroid w.r.t. its constraint set and then moves towards it.

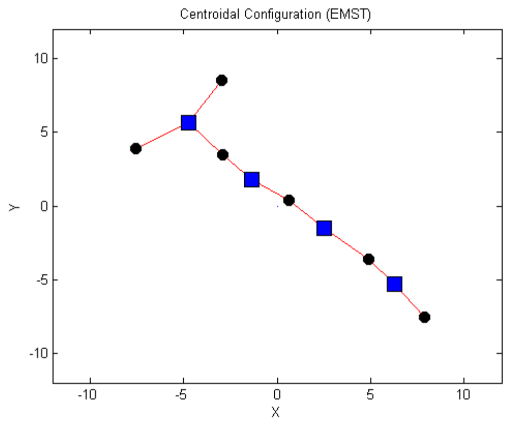

Fig. 15. Centroidal configuration for a heterogeneous team that moves towards goal points that are the centroid of their respective motion constraint set.

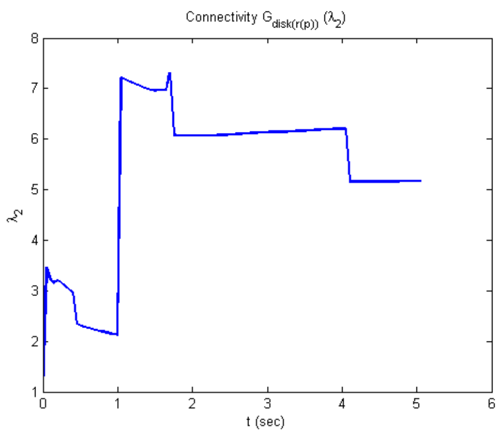

Fig. 16. Graph of the second smallest eigenvalue for the $\mathcal{G}_{\text{disk}(r(p))}$ graph for the centroidal behavior.

Figure 15 show the final configuration of the heterogeneous team after five seconds of the centroid seeking behavior. Figure 16 and Figure 17 show the connectivity with respect to the second smallest eigenvalue criteria ($\lambda_2(\mathcal{G}) > 0 \Rightarrow \mathcal{G}$ is connected) for the $\mathcal{G}_{\text{disk}(r(p))}$ and $\mathcal{G}_{MST_{CW}}$ graph respectively. We see that both graphs stay connected at all times, however only the $\mathcal{G}_{MST_{CW}}$ graph is used to calculate the constraint sets. Note that the larger $\lambda_2\left(\mathcal{G}_{\text{disk}(r(p))}\right)$, the more connections exist in the graph. Figure 16 shows that in the $\mathcal{G}_{\text{disk}(r(p))}$ graph there exist redundant links that allows for some robustness to node failures, however it is unclear at this point to what extent this is true.

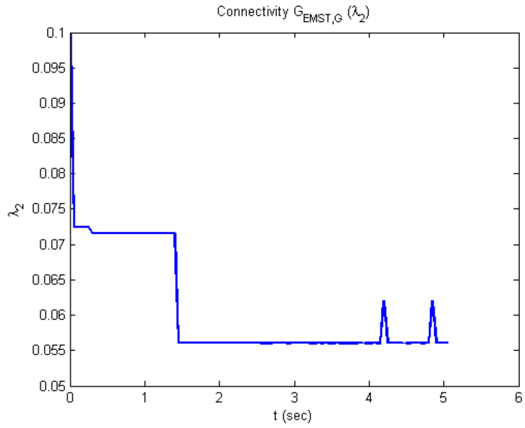

Fig. 17. Graph of the second smallest eigenvalue for the $\mathcal{G}_{MST,\mathcal{G}}$ graph for a simulation of the centroidal behavior.

B. Node Redundancy and Network Robustness

We can see from Figure 15 that utilizing the $\mathcal{G}_{MST_{CW}}$ graph to compute connectivity constraint sets may leave the network vulnerable to single point failures. We can also notice that from Figure 16 that in the underlying heterogenous proximity graph, $\mathcal{G}_{\text{disk}(r(p))}$, there exists redundant links during the simulation. However, from the nature of the dynamic graph, constraints must be imposed on the agents to insure that redundant links exist within the network. One approach to insuring redundant links in the network and therefore network robustness to node failure is computing a *k-vertex-connected* graph and then enforcing this connections through our connectivity constraint sets.

Definition 19: (Menger's Vertex Connectivity) [29] A graph is k-vertex-connected if and only if any two distinct vertices are connected by at least k independent paths. Note that two independent paths do not have any internal vertex in common.

One important property of a *k-vertex-connected* graph is that the graph remains connected when fewer than k vertices are removed [29]. If we can compute a 2-vertex-connected graph, also known as a biconnected graph (G_B), we will maintain network connectivity if one node fails in the network. This redundancy in the network does come at a small cost however. We showed that the fewer links that we take into account when calculating the connectivity constraints, leads to larger motion sets for the agents. By enforcing redundancy in the the network we will be shrinking these motion sets.

In the computer science literature computing a biconnected graph is not a new problem. Work has been done on determining approximations to this problem as well as optimal solutions to special cases of the biconnected graph [30]. Although algorithms do exist in the literature, the problem we are looking at may also be considered a special case of the general problem. By default we are computing the MST of our heterogeneous proximity graph, we would like to compute the biconnected graph using the paths already computed by the MST. In other words we would like to compute a second path from each pair of nodes that is independent of the MST paths. Algorithm 2 outlines how to compute a biconnected graph using the MST of a complete graph.

Algorithm 2 Biconnected Graph Algorithm (G_B)

Assume a complete graph, G with $n+m$ vertices, compute MST_G.
Label each vertex from $i = 1, \ldots, n+m$.
for each vertex pair (i, j) **do**
  remove each internal vertex within the MST_G path connecting the vertices and replace with the remaining set or subset of vertices.
  if no internal vertex **then**
    add an internal vertex within the MST_G path.
  end if
  if remaining set of vertices is empty **then**
    the path is the direct path between (i, j).
  end if
end for

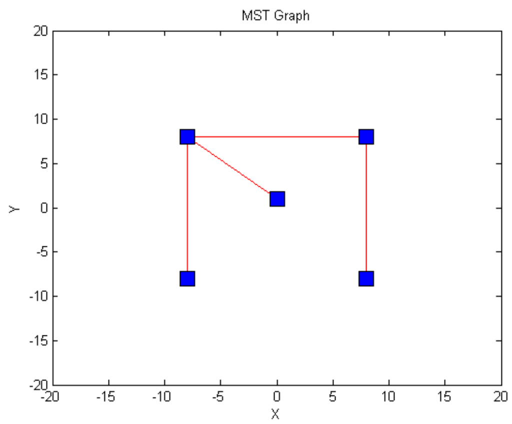

Fig. 18. Minimum Spanning Tree (MST) of five agents. With one node failure the network will become disconnected.

It is straightforward to show that the graph G_B obtained from Algorithm 2 is a biconnected graph. With the construction of the biconnected graph we now have the connections (links) that will insure redundancy in the network. Using the biconnected graph to compute connectivity constraints allows the network to be robust

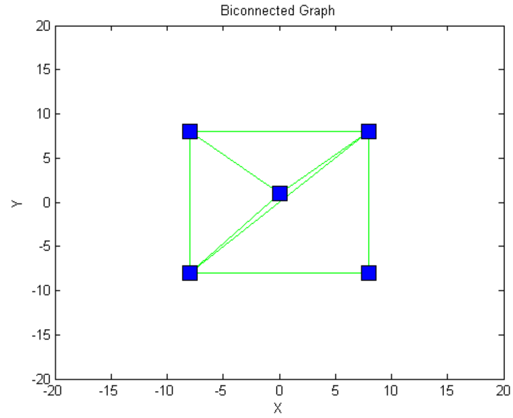

Fig. 19. Biconnectivity graph of five agents utilizing the MST. Notice that one node failure will not make the network disconnected.

to single point failures. An example of the MST and a biconnected graph is shown in Figure 18 and Figure 19 respectively. Algorithm 2 is a general way of obtaining a biconnected graph given the MST of a complete graph and does not address how to compute the minimally weighted biconnected graph. Further investigation is needed to modify the algorithm in order to address this particular problem as well as when the particular graph is not the complete graph.

IX. Prioritized Sensing with Connectivity Constraints

In this section we combine our prioritized sensing behavior with the heterogeneous connectivity constraints to create an algorithm that can send sensing agents to areas with the highest possibility of having good information while also guaranteing that the heterogenous network will remain connected. We also evaluate the performance of the heterogeneous algorithm against its homogeneous equivalent to determine the usefulness of heterogeneity within our approach. We also test the prioritized sensing behavior with various number of relays to understand when adding relays to the network begins to show diminishing returns.

A. Feasible Motion Sets: Sensing Agents

To combine the prioritized sensing objective with the network connectivity constraints we need to merge our probability of detection map, $(M(q))$, and the connectivity constraint sets we computed we previously computed. To do this we refer back to the prioritized sensing algorithm presented in [22], particularly the set D_i, which is the set of points within robot *i's* Voronoi partition not occupied by obstacles. For ease of notation let us define Υ_i as the connectivity constraint set for robot *i* as were defined in Section IV. Let us now define the feasible motion set, $\mathcal{S}_i$, for robot *i* as the following,

$$\mathcal{S}_i = D_i \cap \Upsilon_i. \tag{13}$$

The feasible motion set $\mathcal{S}_i$ for robot *i* is the set of points within its Voronoi partition which is not occupied by obstacles and is limited to the points where network communication links can be maintained. Now we have a set, $\mathcal{S}_i$, which can be optimized such that network connectivity can be guaranteed throughout the search process. The updated prioritized search algorithm which takes into account network connectivity is outlined in Algorithm 3.

Algorithm 3 Prioritized Sensing with Connectivity Constraints

```
while t < t_final do
  for x_i = 1, ..., n do
    Calculate Υ_i from G_{MST_CW} (Equations (2)-(8))
    Determine V_i ∈ Q
    Calculate S_i = D_i ∩ Υ_i
    Optimize over S_i to determine an approximate
    maximum g̃* in S_i of M(q)
    if g̃* is reachable then
      g_{p_i} ⇐ g̃*
    else
      g̃_i* = max(Y_1, ..., Y_{N-1}) excluding g̃* that
      was previously calculated
    end if
    Calculate f_i(p_i, q_{xi}, d) in S_i with g_{xi} set as the
    goal point.
    u_i = -k∇f_i(·) where k = |p_i - g_{xi}|
    ∀q ∈ R_s, M(q) = βM(q)
    Exchange map information with neighbors in
    G_{MST_CW} graph
  end for
end while
```

In Algorithm 3 it is seen that when network connectivity is taken into consideration, the set over which the probability of detection map is being optimized may "shrink" due to the intersection of the two sets D_i and Υ_i. This makes sense since now the algorithm has to come to a compromise between the sensing objective and the network connectivity objective. Proposition 10 states a property of the feasible motion set.

Proposition 10: Given the set D_i and motion constraint set Υ_i, then the feasible motion set $\mathcal{S}_i \neq \emptyset$.

Proof: This comes as a consequence of the construction of D_i and Υ_i. Mainly the fact that D_i will always contain at least agent *i's* current position p_i. Also by construction Υ_i will also always contain at least p_i. Therefore in the worst case scenario, $\mathcal{S}_i = D_i \cap \Upsilon_i = p_i$. ■

It can be seen by the construction of the feasible motion sets for the sensing agents that taking network connectivity into consideration may produce a suboptimal prioritized search in some cases. This is due to the fact that the set of optimal solutions that guarantee connectivity Υ_i, may not contain the global optimal over an agent's particular Voronoi partition. Proposition 11 states when the algorithm will produce the optimal solution with respect to the prioritized sensing algorithm.

Proposition 11: If $\mathcal{S}_i = D_i \cap \Upsilon_i = D_i$ then the goal point g_{p_i} calculated in Algorithm 3 is an optimal solution to the prioritized sensing algorithm presented in [22].

Proof: $\mathcal{S}_i = D_i \cap \Upsilon_i = D_i$ implies that $D_i \subset \Upsilon_i$, *i.e.* the set $\mathcal{S}_i$ includes all possible global optima as if connectivity constraints were ignored. Therefore the goal point g_{p_i} is the optimal point in the set D_i which is the optimal solution to the prioritized sensing algorithm outlined in [22]. ■

Proposition 11 tells us that if the communication constraints sets are "sufficiently large," then the connectivity constraints can be ignored without losing network connectivity. This is very intuitive in the sense that if an agent can communicate over a larger area than it has been allotted to search, it shouldn't have to consider network connectivity as a constraint on its search/sensing behavior.

B. Feasible Motion Sets: Relay Agents

Our approach to addressing connectivity maintenance of a sensor network is to convert the network to a heterogeneous one by adding relay agents capable of better communication capabilities. For this particular application we are assuming that sensing agents are UGVs and relay agents are UAVs flying at a constant altitude. With these assumptions relay agent collisions with sensing agents in the network are not considered. Also, relay agents communication range is considered to be its projection on the two dimensional space.

For the motion planning of relay agents in the network, Algorithm 1 is used to compute the centroid of each relay agent's motion constraint set (MCS_i). Therefore, relay agent *i's* feasible motion set is MCS_i. It was observed in section VIII that Algorithm 1 produces behavior of the relay agents that acts to balance the network in terms of the distances between its connected links. This is a desirable behavior in sensing/search problems because it allows the network to "stretch" as sensing agents move towards the outer regions of the search space. In essence, as a sensing agent moves towards uninvestigated areas it "pulls" a relay agent with it in order to maintain connectivity of the network.

From algorithm 1 we have the following property which addresses collisions between relay agents at a constant altitude.

Proposition 12: Given $MST_{\mathcal{G}_{\text{disk}(r(p))}}$ and goal points g_{p_i} computed from Algorithm 1, no two relay agents will occupy the same position at time $kT_s, \forall k \in \mathbb{Z}^+$.

Proof: Let us assume that agent *i* and agent *j* do occupy the position at time kT_s. From Algorithm 1 it seen that this can only occur when agent *i* and agent *j* share an edge in the $MST_{\mathcal{G}_{\text{disk}(r(p))}}$ and also share edges with the same agents (vertexes) in the $MST_{\mathcal{G}_{\text{disk}(r(p))}}$. This however is a contradiction of the construction of the minimum spanning tree (MST), *i.e.* the MST is a tree. In other words, links between any two agents (vertexes) are unique. Therefore, no two agents can occupy the same position at time $kT_s, \forall k \in \mathbb{Z}^+$. ■

Proposition 12 states that at each iteration of the algorithm no two relay agents will occupy the same position within the search space. This however is not enough to guarantee collision avoidance for all time t. Since we are assuming a second order system for the agents, we cannot speak to the trajectories between time kT_s and $(k+1)T_s$, *i.e.,* the agents may experience overshoot, steady state error, etc. However, with a well tuned system, collisions between relay agents can be addressed during their arrivals to their respective goal points.

This leads us to the final property when Algorithm 1 and Algorithm 3 are utilized to control a heterogenous sensor network composed of relay (UAVs) and sensing (UGVs) agents.

Theorem 13: Given a heterogeneous sensor network as described in Section III where relay agents utilize inputs described by Algorithm 1 and sensing agents utilize inputs described in Algorithm 3, then the heterogeneous sensor network will remain connected and collision free throughout the search process.

Proof: Collision avoidance for the sensing agents within the network is guaranteed from [22] (Proposition 1). Also, since we are assuming that relay agents are UAVs flying above sensing agents (UGVs), sensor and relay agent collisions are avoided. Lastly, collisions avoidance between relay agents is guaranteed by Proposition 12. Connectivity maintenance is guaranteed by Theorems 5 and Theorem 6. ∎

Remark 14: It is also key to notice that by construction Algorithm 1 and Algorithm 3 can be implemented in a decentralized fashion. With an initially connected network, the Voronoi partitions can be constructed with only neighboring agents knowledge. Under the assumption that there exist synchronization within the network and all agents can broadcast position information with a unique ID, then the heterogeneous proximity graph as well as the MST of the heterogeneous proximity graph can be constructed in a decentralized fashion [28].

X. Simulations

For the following simulations relay agents apply Algorithm 1 while sensing agents apply Algorithm 3. To understand how the communication constraints as well as adding relay agents to the sensor network affects the prioritized sensing behavior, we simulate two scenarios. For the first simulation we implement Algorithm 3 on a heterogeneous network made up of 7 sensing agents and 4 relay agents. The search space is taken to be $60m \times 60m$ square area, with the communication ranges for the sensing agents, $R_c = 3m$ and $R_{rc} = 16m$ for the relay agents. The sensing agents in the network are initialized in a random configuration with the relay agents situated at (-8m,-8m),(-8m,8m),(8m,-8m),(8m,8m) such that the initial configuration of the heterogeneous network is connected. The sensing radius of the sensing agents are taken to be $3m$ and the parameter that reflects the reduction of the probability of detection map, β is taken to be 0.8. Each simulation lasts for 50 iterations of the algorithm, approximately 120 seconds. Table I summarizes the results for five simulations.

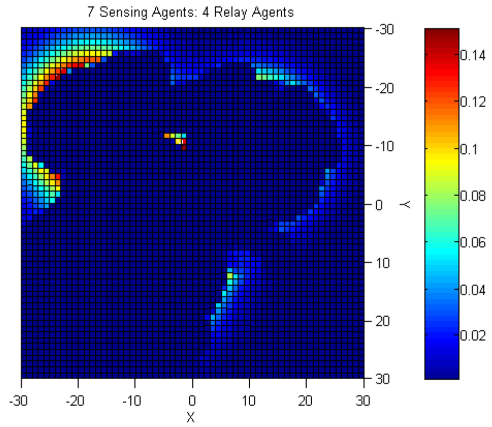

Fig. 20. Probability of Detection Map after 50 iterations of the algorithm. Notice that most of the area has been searched with exception to the upper left hand corner.

TABLE I
Heterogeneous Sensor Network (7 sensing, 4 relay agents): Prioritized Sensing

Simulation Number	Average POD per m^2	Max POD Value
1	0.0043	0.178
2	0.0104	0.337
3	0.0034	0.168
4	0.0031	0.124
5	0.0147	0.333

Figure 20 shows the reduced POD map, $M(q)$, after 50 iterations of the algorithm with seven sensing agents and four relay agents. Notice that most of the area has already been searched and has been reduced to around 0.14.

In the second set of simulations we assume there are no relay agents in the network. All parameters were kept the same as in the first set of simulations except only 7 sensing agents were used in the sensor network. This set of simulations shows how the prioritized sensing algorithm performs when the communication constraints are imposed on the homogeneous network of only sensing agents without the help of specialized relay agents. The results of this set of simulations are summarized in Table II.

TABLE II
Homogeneous Sensor Network (7 sensing agents): Prioritized Sensing

Simulation Number	Average POD per m^2	Max POD Value
1	0.143	0.98
2	0.144	0.99
3	0.144	0.98
4	0.142	0.97
5	0.146	0.99

Figure 21 shows the reduction of the POD map after 50 iterations when the connectivity constraints are imposed

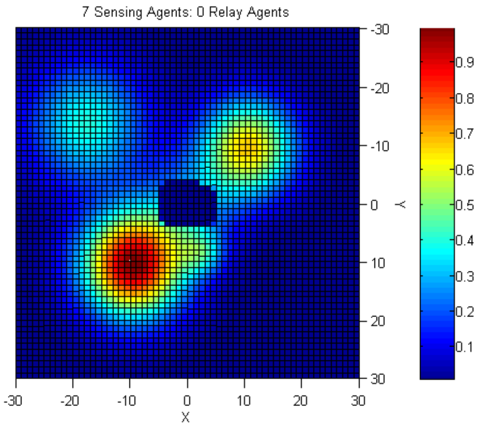

Fig. 21. Probability of Detection Map after 50 iterations of the algorithm with a homogeneous sensor network with connectivity constraints. Notice that most of the area has not been searched within the time allotted.

directly on the homogeneous sensor network of seven sensing agents. Notice that the majority of the area has yet to be searched and the most probable areas of containing good information have not been searched. This shows how much the connectivity constraints of the network can inhibit an efficient search of the area.

From Table II we see that imposing the communication constraints directly on the network with only sensing agents hinders the prioritized sensing algorithm significantly. The main reason for such poor results as compared to the heterogeneous network with relays (Table I) is the fact the communication radius of each sensing agent is very small, $3m$, as compared to the search area they are tasked to explore, $60m \times 60m$. From the first two sets of simulations it is clear that utilizing a heterogeneous sensor network is advantageous when the area to be scanned is much larger than that of the communication radius of the sensing agents. As we first hypothesized, the relay agents enable the sensor network to have a larger reach, thus enabling a more efficient behavior of the prioritized sensing algorithm when communication constraints are taken into consideration.

The next set of simulations we look at is when relay agents are replaced by more sensing agents. In this scenario the sensor network is comprised of 15 sensing agents and zero relay agents. This set of simulations should give us an indication if simply adding more sensing agents to the network can overcome the constriction of the network constraints. Table III shows the results for this particular scenario.

Comparing Table I with Table III we see that even with more than twice the amount of sensing agents in the network compared to the previous simulations, the communication constraints do not allow the network to "stretch out" enough to efficiently search the area in the time allotted. This clearly shows the advantage of using specialized agents, relays in this case, to help extend the range of the sensor network.

To understand the effect of adding relay agents to the sensor network we conduct several simulations of the algorithm with different numbers of relay agents for 50 iterations, approximately 120 seconds. Table IV summarizes the outcome of these particular simulations. Note that when 16 relay agents are used in the sensor network, complete communication coverage of the search area is obtained.

TABLE III
LARGE HOMOGENEOUS SENSOR NETWORK (15 SENSING AGENTS): PRIORITIZED SENSING

Simulation Number	Average POD per m^2	Max POD Value
1	0.128	0.98
2	0.125	0.97
3	0.122	0.97
4	0.124	0.98
5	0.131	0.98

TABLE IV
ADDING RELAY AGENTS TO SENSOR NETWORK (15 SENSING AGENTS): PRIORITIZED SENSING

# of Relay Agents	Average POD per m^2	Max POD Value
1	0.044	0.504
2	0.031	0.396
3	0.018	0.387
4	0.002	0.174
5	0.002	0.161
8	7.9×10^{-5}	0.020
10	7.2×10^{-5}	0.009
16(Full Coverage)	6.3×10^{-7}	0.001

Figure 22 shows a graph of how the maximum POD value after the algorithm was run for 50 iterations changed with additional relay agents in the network. We can see a fairly good improvement of the maximum POD value with the addition of up to 8 relay agents. In these simulations, adding more than 8 relay agents to the network did not significantly change the outcome of the algorithm in the allotted time.

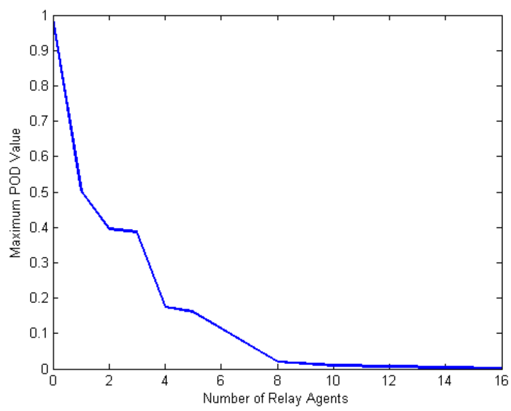

Fig. 22. Change in maximum POD map value with additional relay agents in the sensor network. The simulations lasted 50 iterations of the algorithm or about 120 seconds.

XI. Experimental Results

This hardware experiment was conducted to validate the claims of Theorem 5 and Theorem 6. This section looks at the particular situation when the relay agent of the heterogeneous team moves towards its centroid of its feasible motion set. The rest of the heterogeneous network consists of two sensing agents. In this particular network configuration, the behavior of the relay robots mimics that of balancing the network in the form of keeping equidistance between the two sensing agents. In this sense, the relay agent is trying to give equal network considerations to each sensing agent.

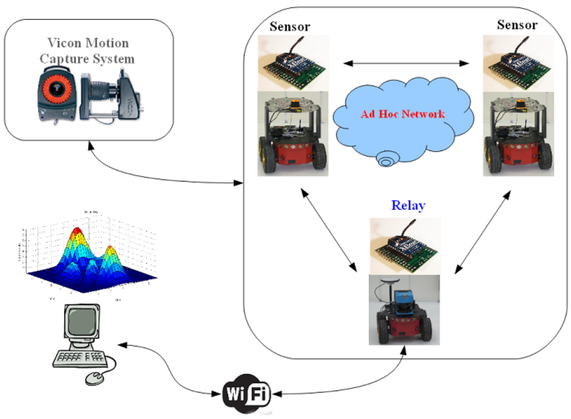

Fig. 23. Diagram of hardware experiments using the centroid seeking behavior.

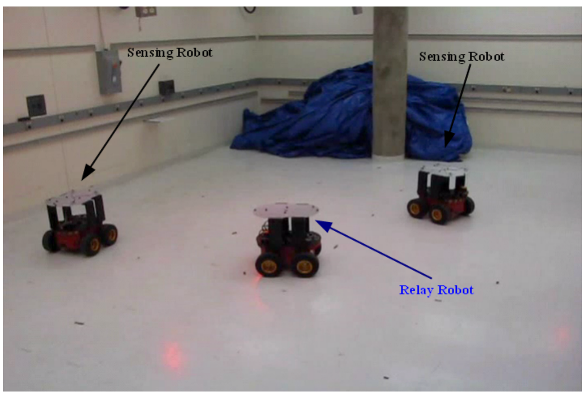

Fig. 24. Experimental snapshot showing the two sensing agents and a single relay agent.

For the hardware implementation of the communication constraints, we chose to implement Algorithm 1 on a single relay robot. Two sensing robots were given predefined trajectories and were tasked with taking light intensity measurements along these trajectories. The relay robot calculates its feasible motion set based on the positions of the sensing agents and then moves towards its centroid. Position information of the sensing agents were updated every 0.5 seconds. For this experiment we used $R_{rc} =$ 3.2m and $R_c = 1.0$m for the communication radius of the relay and sensing agents respectively. Figure 23 shows a diagram of how the experiment was implemented and Figure 24 shows a snapshot of the hardware setup. The ad-hoc network consisting of three XBee wireless RF Modules was used to communicate sensing data between robots. This allowed for the light intensity map to be built in a distributed fashion. A wireless local area network (WLAN) was used to send a real-time light intensity map from the relay robot to an end user using a laptop outside the experimental area. Figure 25 shows the evolution of the second smallest eigenvalue of the $\mathcal{G}_{\text{disk}(r(p))}$ graph and shows that the heterogeneous network stays connected for the entire experiment.

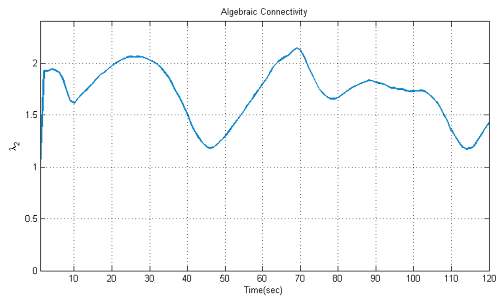

Fig. 25. The second smallest eigenvalue of the $\mathcal{G}_{\text{disk}(r(p))}$ graph during experiments of the centroidal behavior. The network remains connected throughout the experiment.

XII. Conclusion

In this paper we addressed prioritized sensing behaviors with communication constraints for a heterogeneous sensor network made up of sensing agents and mobile communication relays. We derived connectivity constraints for a heterogeneous sensor network which allowed for the development of feasible motion sets that guarantee network connectivity for agents within the network. Lastly, we showed how to reduce the number of communication constraints to allow the sensing agents to maximize their feasible motion sets and thus allow for a larger search area while maintaining network connectivity. A technique for shaping the network configuration was also presented that allows for biasing particular communication links within the network which shapes the flow of information within the sensor network.

Future research directions include formulating network connectivity in a probabilistic sense, *i.e.*, assign a probability of becoming disconnected given certain configurations rather than a strick geometric approach to connectivity maintenance. Extending our formulation for deriving communication constraints when the sensing agents are not limited to planer motion is also an area of future work. Another area for extending this framework lies in relaxing the assumption that the sending and receiving range of each agent is symmetric. One question that can be asked is, how does having a larger sending range than receiving range change the heterogeneous proximity graph construction and what are the implications to the efficiency of the prioritized search? Also a more in depth investigation is needed to understand the robustness of the network to node failures and how it may be incorporated in our current framework.

REFERENCES

[1] J. C. Latombe, *Robot Motion Planning*. Kluwer, 1991.

[2] J. Fax and R. Murray, "Graph laplacian and stabilization of vehicle formations," *Proceeding of the 15th IFAC*, pp. 283–288, 2002.

[3] H. G. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part 1: Fixed topology," *IEEE Conference on Decision and Control*, pp. 2010–2015, December 2003.

[4] D. V. Dimarogonas and K. H. Johansson, "Decentralized connectivity maintenance in mobile networks with bounded inputs," *IEEE International Conference on Robotics and Automation*, pp. 1507–1512, May 2008.

[5] A. Muhammad, M. Ji, and M. Egerstedt, "Applications of connectivity graph processes in networked sensing and control," *Networked Embedded Sensing and Control, Lecture Notes in Control and Information Sciences(LNCIS)*, vol. 331, 2006.

[6] N. Ayanian, V. Kumar, and D. Koditschek, "Synthesis of controllers to create, maintain, and reconfigure robot formations with communication constraints," in *Proceedings of International Symposium of Robotics Research*, Lucerne, Switzerland, August 2009.

[7] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Maintaining connectivity in mobile robot networks," *O. Khatib et. al (EDS.): Experimental Robotics: The 11th International Symposium., STAR 54*, pp. 117–126, 2008.

[8] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, December 2008.

[9] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[10] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions On Robotics*, vol. 23, no. 4, pp. 693–703, August 2007.

[11] D. P. Spanos and R. M. Murray, "Robust connectivity of networked vehicles," *IEEE Conference on Decision and Control*, pp. 2893–2898, December 2004.

[12] J. Fink and V. Kumar, "Online methods for radio signal mapping with mobile robots," *IEEE International Conference on Robotics and Automation*, pp. 1940–1945, May 2010.

[13] O. Tekdas, P. A. Plonski, N. Karnad, and V. Isler, "Maintaining connectivity in environments with obstacles," *IEEE International Conference on Robotics and Automation*, pp. 1952–1957, May 2010.

[14] O. Tekdas, W. Wang, and V. Isler, "Robotic routers: Algorithms and implementation," *The International Journal of Robotics Research*, vol. 29, no. 1, pp. 110–126, January 2010.

[15] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnstrom, and P. R. Olson, "Positioning unmanned aerial vehicles as communication relays for surveillance tasks," in *Conference on Robotics Science and Systems*, 2009.

[16] S. Gil, M. Schwager, B. J. Julian, and D. Rus, "Optimizing communication in air-ground robot networks using decentralized control," *IEEE International Conference on Robotics and Automation*, pp. 1964–1971, May 2010.

[17] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," *IEEE International Conference on Robotics and Automation*, pp. 4457–4462, May 2010.

[18] N. Chakraborty and K. Sycara, "Reconfiguration algorithms for mobile robotic networks," *IEEE International Conference on Robotics and Automation*, pp. 5484–5489, May 2010.

[19] E. W. Frew, "Information-theoretic integration of sensing and communication for active robot networks," *Mobile Networks and Applications*, vol. 14, no. 3, pp. 267–280, June 2009.

[20] I. Hussein, D. Stipanović, and Y. Wang, "Reliable coverage control using heterogeneous vehicles," in *IEEE Conference on Decision and Control*, New Orleans, 2007, pp. 6142–6147.

[21] M. Stachura and E. W. Frew, "Cooperative target localization with a communication aware active sensor network," in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, August 2010.

[22] R. A. Cortez, R. Fierro, and J. Wood, "Prioritized sensor detection via dynamic voronoi-based navigation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2009, 5815-5820.

[23] B. Bollobás, *Modern Graph Theory*. Springer-Verlag New York Inc., 1998.

[24] A. Ghaffarkhah and Y. Mostofi, "Communication-aware target tracking using navigation functions - centralized case," *International Conference on Robot Communication and Coordination (ROBOCOMM)*, pp. 1–8, 2009.

[25] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009, electronically available at http://coordinationbook.info.

[26] B. Bollobàs, *Modern Graph Theory*. Springer-Verlag New York, 1998.

[27] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and Systems*, vol. 5, no. 1, pp. 66–77, 1983.

[28] M. Khan and G. Pandurangan, "A fast distributed approximation algorithm for minimum spanning trees," *Lecture Notes in Computer Science*, vol. 4167, pp. 355–369, 2006.

[29] A. Gibbons, *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[30] S. Khuller and U. Vishkin, "Biconnectivity approximations and graph carvings," in *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1992, pp. 759–770.

R. Andres Cortez received his B.S. degree in mathematics from New Mexico Highlands University, Las Vegas, NM, in 2005. He received his M.S. and Ph.D., both in mechanical engineering, from the University of New Mexico, Albuquerque, NM, in 2007 and 2010 respectively. In 2011 he joined Los Alamos National Laboratory as a research and development engineer with the Weapons Test Engineering Group.

Rafael Fierro is an Associate Professor of the Department of Electrical & Computer Engineering, University of New Mexico where he has been since 2007. He received a M.Sc. degree in control engineering from the University of Bradford, England in 1990 and a Ph.D. degree in electrical engineering from the University of Texas-Arlington in 1997. Prior to joining UNM, he held a postdoctoral appointment with the GRASP Lab at the University of Pennsylvania and a faculty position with the Department of Electrical and Computer Engineering at Oklahoma State University. His research interests include nonlinear and adaptive control, robotics, hybrid systems, autonomous vehicles, and multi-agent systems. He directs the Multi-Agent, Robotics, Hybrid and Embedded Systems (MARHES) Laboratory. Rafael Fierro was the recipient of a Fulbright Scholarship, a 2004 National Science Foundation CAREER Award, and the 2007 International Society of Automation (ISA) Transactions Best Paper Award. He is serving as Associate Editor for the IEEE Control Systems Magazine and IEEE Transactions on Automation Science and Engineering.

John Wood is a Professor of Mechanical Engineering, and Director of the Manufacturing Engineering Program, at the University of New Mexico. He has a Ph.D. in mechanical engineering from MIT, and B.Sci. in engineering design and economic evaluations from the University of Colorado. Prior to UNM, he was a Professor in Bioengineering, and Associate Director of the Center for Engineering Design, both at the University of Utah. Prof. Wood is the Coordinating Editor of the Micromechanics Section of the international Journal of Sensors and Actuators.