Adaptive Control Network for Multi-Robot Exploration

Jose Vazquez and Malcolm Chris

IPAB, School of Informatics, Mayfield Road,
EH75RG, Edinburgh, UK

Abstract. This work addresses the problem of exploring an environment with a team of communicating robots. Exploration can be performed more efficiently when robots are able to communicate and coordinate their actions. We propose an adaptive control approach to keep the robots as a single connected network. In this approach a control network is created at the beginning of the exploration based on the communication network. As the robots traverse the environment the control network is updated to enhance connectivity. The approach has been implemented for Line of Sight and Radio Frequency technologies. Our approach has been compared with coordination approaches that rely on fixed networks. The results show that our approach performs better than these fixed network approaches.

1 Introduction

The problem of exploration is an important problems in robotics because of applications such as rescue, mowing, or cleaning in which the complete exploration of area is the main objective. In this paper, we consider the problem of exploring unknown environments using teams of mobile robots with local communication systems.

Previous research in multi-robot exploration indicates that effective exploration using multiple robots requires coordination through communication [6]. Low power robots are likely to have limited communication range. As such robots traverse the environment they form a Mobile *AD HOC* communication NETwork (*MANET*). Keeping this *MANET* connected is a difficult task. Previous work relies on a fixed control network to keep the communication network connected. The imposition of a fixed network limits the mobility of the robot network, slowing down the exploration process.

In this paper we present the *BERODE* (Behaviour based decentralized) architecture. The *BERODE* architecture implements an adaptive control approach to maintain the communication network. In *BERODE* the robots update the control network to improve the communication conditions. The robots assume behaviours according to their status in the control network. The behaviours are designed to avoid collisions between the robots, encourage the exploration of unknown areas and keep the *MANET* connected. In effect the exploration algorithm of the system emerges from the interaction of individual behaviours within the environment.

The *BERODE* architecture has been implemented for *RF* (Radio Frequency) and *LOS* (Line of Sight) communication models. In the *RF* implementation the robots are assumed to be able to measure the Received Signal Strength Level (*RSSL*) and use this

Vazquez J. and Chris M. (2006).
Adaptive Control Network for Multi-Robot Exploration.
In *Proceedings of the 2nd International Workshop on Multi-Agent Robotic Systems*, pages 44-53
DOI: 10.5220/0001222700440053

value to keep the communication network connected. In the *LOS* implementation the robots use their Cartesian distances to emulate the *RSSL* value.

To enable scalability to large numbers of robots the *BERODE* architecture implements a hierarchic approach to distributing information. In *BERODE* the robots share information frequently at the local level and less frequently at the global level. The local level is formed by the robots that are within a *k-hop* distance in the control network.

Compared to coordination approaches that rely on fixed networks, *BERODE* has a better performance, in that it explores the environments more efficiently and keeps the *MANET* connected for more time.

2 Related Work

The use of robot teams to explore unknown environments has been the subject of extensive research [8]. Exploration tasks can be completed more efficiently when robots share relevant information with each other [6]. To enable information exchange the robots have to maintain communication between the members of the team.

In previous research this problem has been addressed by maintaining fixed *LOS* communication network topologies. *Leader-follower* relations are imposed on all the robots with the exception of a team leader. The team leader directs the exploration while the rest of the team follows it.

Wagner [12] developed algorithms to cover an area. He identified trade-offs between area coverage and communication safety concluding that coverage degrades when plans are communication-focused. Powers [4] proposed a navigation behaviour called VBCP (Value-Based Communication Preservation) to preserve communication while traversing an environment. VBCP calculates movement vectors using the *RSSL* from the robots, the robots' positions and map-based predictions of the *RSSL* for nearby positions to the robot position. Ulam [9] proposed a reactive approach to recover communication in a surveillance mission. Several recovery strategies were proposed. He concluded that there are still remaining issues to determine the best strategies to recover communications in large-scale teams. Kantor's [1] work focuses on surveillance where navigation is achieved by relying on an *RF* network deployed *a priori*. Navigation was successfully achieved but the authors remarked that the minimal density of sensors required to achieve successful navigation was unclear.

Recently Thibodeau [7] proposed an adaptive topology algorithm where a leader robot frequently builds an *MST* (minimum spanning tree) *control network*. Thibodeau compared his approach to fixed configuration topologies (e.g. chain, fixed tree). The comparison was based on the time to build a complete map. Thibodeau's approach performs better than fixed configurations.

The architectures and strategies in the literature that maintain communication have relied on *a priori* definitions of robot communication relations. In real world environments as the robots explore the environment *a priori* relations limit the mobility of the network. We argue that the relations between team members should be determined dynamically to gain flexibility. Our approach is similar to Thibodeau's approach because we implement an *MST control network*, but in our approach the number of explorers varies dynamically.

3 The BERODE Architecture

Our *BERODE* architecture is based on behavioural roles such as *Explorer* and a communication *Maintainer*. These roles reactively adapt to the dynamic conditions of the *MANET* formed by the robots as they explore an environment. The *MANET* is kept fully connected by creating and updating an *MST control network*, a subnetwork containing only the necessary connections (*control connections*) to keep the *MANET* connected.

The robots select their behavioural role based on their *network status* and their internal state. The *network status* of a robot comprises the safety level and the set of *constraints*. The safety level for a robot is determined by the *signal quality* of its connections on the *MST control network*. The *signal quality* depends on the communication technology. In *RF* technologies the *RSSL* (Received Signal Strenght Level) is typically available to the robots. This value is used as the *signal quality*, whereas in *LOS* technologies the Cartesian distance between the robots is used to estimate the *signal quality*. A robot is on a safety level if all of its connections have at least that safety level. *BERODE* implements three safety levels with the following decreasing order in safety: *safe*, *precautionary* and *unsafe*. Depending on the safety level a set of *constraints* is imposed. In the *safe* level the set of *constraints* is empty because the *signal quality* for the *control connections* is above the *safe* threshold (σ_{safe}). In the *precautionary* level the set of *constraints* is formed by the *control connections*. In the *unsafe* level the set of constraints is formed by the *control connections* for which the *signal quality* is below the *unsafe* threshold (σ_{unsafe}). We argue that a robot could move out of the *unsafe* level faster when it is constrained only by the subset of *unsafe* connections instead of the complete set of *control connections* [10]. It is desirable that $\sigma_{safe} >> \sigma_{unsafe}$ to avoid the risk of disconnecting the *MANET* because of the temporary exclusion of some *control connections*.

According to its behavioural role, a robot exhibits some interest or none at all in the exploration task. As a result, a variable number of robots direct the exploration towards unexplored areas while the rest of the robots keep the network connected.

The behavioural roles generate reactive plans that keep the robot's *constraints* within communication range. The plans are based on the imposition of *virtual forces* by the robot's *constraints*. *Virtual forces* are attractive/ repulsive relations between pairs of robots. These forces are modelled as *virtual springs* where the free spring length is a function of the *signal quality* and behavioural role of the robot and its connections. The reactive plans randomly sample nearby positions to the robot position and generate a plan to move to the position where the energy for the *constraints* is minimized.

Robots are also attracted to unexplored areas; the attractiveness of an unexplored area is a function of its size, the path length and the predicted communication *safety* level at that location. This level is the estimated *signal quality* based on the current positions of the robots.

Within the robot network, information is distributed to the point where all the robots share consistent models of the environment. The information is distributed using the *MST control network*. Each robot shares its positional and sensorial information with the rest of the team periodically. Information is shared at two levels: frequently at the *local* level and less frequently at the *global* level. The robots have to cope with the delays in the reception of information.

The environment is represented by means of a feature based map. Each robot builds and updates it's own map. An Extended Kalman Filter (*EKF*) is used for localization. Robots extract features from their sensors and update their representation. Extracted features are distributed among the team of robots. Robots incorporate the received features with their locally extracted features. The following sections describe the main components of the *BERODE* architecture.

3.1 The Adaptive MST Control Network

In *BERODE* the *MANET* is kept fully connected by creating and updating a *MST control network*. The *MST control network* is calculated at the start of the exploration based on the *MANET* and a signal criterion. The signal criterion depends on the implementation; for instance for typical *RF* technologies the *RSSL* between a pair of robots is the link cost. After the initial calculation the robots retain knowledge of their *K-MST control network* (*local network*). The *local network* for a robot is the network that contains all the robots within a *k-hop* distance.

The *MST control network* can be modified either partially or completely by robots over time. Merging and validation mechanisms are implemented to ensure that the robots maintain the same *MST control network*. Robots periodically re-evaluate their *local network* to improve connectivity; if necessary they modify the *local network* and inform all the robots within the *local network.*

3.2 Control Architecture for the Robots

The robots' control architecture is the same regardless of their behavioural roles. The architecture has two control levels: *social* and *internal*. At the *social level* the robot selects its behavioural role. The modules at this level are the same for all the behavioural roles, while at the *internal level* the behavioural roles have different modules. At the *internal level* the current behavioural role generates reactive plans to meet the *constraints* generated at the *social level*. These plans are adapted if necessary to ensure safety.

Fig. 1 presents the control and information flow diagrams for the robots' control architecture. The modules receive information either periodically or on an event basis. The Communication Manager that handles the information exchanged between the robots in the network is composed of four modules. The *K-MST* Control Module receives the messages related to the status of the network. These messages are received either periodically or on an event basis. This module keeps track of the *constraints* from which the *virtual forces* are derived and generates a *network event* when a change in the set of the *constraints* or the roles of the constraints is detected.

The Behaviour Selection Module is called once a *network* or an *internal event* occurs. *Internal events* occur when a robot achieves its current task, detects a change in the *safety level*, or modifies either the *local network* or the *MST control network*.

The Local and Global World Model modules are temporary storage modules to handle the *local* and *global features* received from the robots in the network. *Global features* are integrated into the local feature map in the map building module. *Local features* are used to aid navigation but they are not integrated in the local feature map.

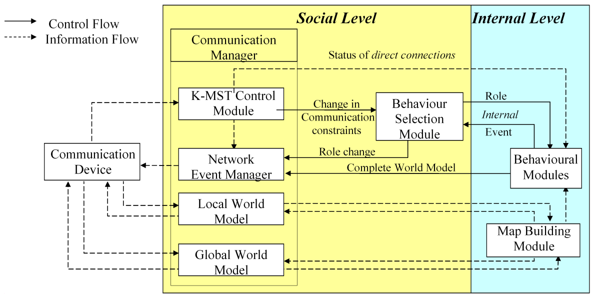
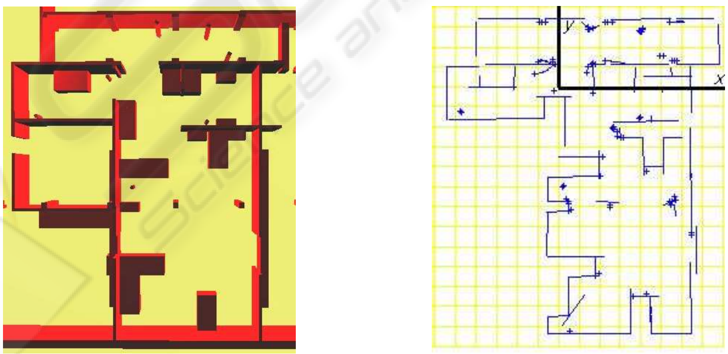


Fig. 1. Control architecture for the robots in $BERODE$.

The observations of these features are integrated to the map as a part of the *global features*. These modules also store and periodically transmit the local and global features observed by the robot.

The map building module builds a map of the environment containing line and point features. Features are extracted from raw sensor data. A low cost platform based on sonar and infrared sensors is used to sense the environment. A *feature management* process extracts, segments and associates the features. The extracted features are then used to update the *EKF* and improve the estimates of the locations of the robots and features. *A priori* structural knowledge (e.g. wall parallelism) is used to improve the quality of the maps. Fig. 2 presents an example of the maps built by the robots. In the experiments the robots start grouped in the upper left area. We assume the robots begin in known locations in a close group. In practice this could be achieved in many ways, such as started them one after the other on the same spot.



a) Environment b) Map built

Fig. 2. An office like environment used in the experiments.

All *non-Explorer* robots make predictive plans to keep the *MANET* connected. They sample positions near the robot, generating a plan to move to a position where the energy from the *spring forces* and the obstacles is minimised. Obstacles generate repulsive potential fields that are a function of the distance. The *spring forces* generated by the *constraints* are a function of the difference between the current *signal quality* and a desired *signal quality*. The desired *signal quality* is a threshold whose value depends on the behavioural roles of the robots [10]. In the *RF* model the *RSSL* is used as the *signal quality*, whereas in *LOS* communication the Cartesian distance is used to emulate the *signal quality* value.

The *Explorer* robots plan movements towards unexplored areas by estimating the *signal quality* in the unexplored areas, and selecting the most attractive area. This is the *frontier* with the largest utility in the safest hierarchic level. A *frontier* is a portion of free space that is adjacent to unknown space in the projected grid map. The utility of a *frontier* is the information gain value minus the cost of idealized travel to the *frontier* from the robot position. The *frontiers* have a hierarchy level based on the communication coverage at the *frontier* position. The hierarchy levels minimize the number of exploration failures. An exploration failure occurs when a robot generates a plan to a *frontier* which it aborts because of possible loss of communication.

3.3 The Behavioural Roles

BERODE implements four behavioural roles: *Recoverer*, *Explorer*, *Maintainer* and *Pusher*. These roles are manifested under the following conditions:

1. *Recoverer*: robot is in the *unsafe* level.
2. *Explorer*: The robot is in the *safe* level, has one connection in the *MST control network* and there is at least one unexplored area predicted as *communication safe.*
3. *Maintainer*: The robot is in the *safe* or *precautionary* level, and has more than one connection in the *MST control network*.
4. *Pusher*: The robot is in the *safe* or *precautionary* level, has one connection in the *MST control network* and no unexplored areas predicted as *communication safe.*

The first three roles are based on previous research on the areas of network maintenance [7] and recovery [9]. In this previous research there is only one robot that explores (*Explorer*) the space while the rest of the team (*Maintainers*) keep the network connected. *BERODE*, in contrast with previous approaches, allows several robots to explore the space at the same time. This speeds up the exploration process but under certain circumstances conflicts arise between explorations pulling in diferent directions. The *Pusher* role resolves these conflicts on the network in a decentralized fashion. One of the pulling *Explorers* becomes an active tail: a *Pusher*. A *Pusher* robot is the result of the lack of unexplored areas that are communication *safe*. As the *Pushers* traverse the environment they may discover *safe* unexplored areas, and can then transition to the *Explorer* behavioural role.

The goal for *non-Exploring* robots is essentially the same: Move towards the location that maximizes the predicted *OSQ* (*overall signal quality*) for the *constraints*; the difference is in the parameters of the *virtual spring* model. These parameters are a function of the behavioural roles of the robots. This type of parameterization generates local interactions that induce *leader–follower* motions in the robot network where the *Explorers* direct the exploration and the *Pushers* accelerate the movement in the exploration directions [10]. The goal for the *Explorer* behavioural role is to move towards unexplored areas while maintaining a *safe* connection.

The effect of an *unsafe* connection on the global behaviour is the contraction of the network until the *unsafe* connection recovers a higher quality level and is detected as *precautionary* or *safe* at which point the robots continue the exploration. When a pair of robots detects an *unsafe* connection they transition to the *Recoverer* role and back track their most recent movements. Afterwards if the connection is still *unsafe* they move towards each other according to the last received positions.

3.4 Hierarchical Information Distribution

To achieve coordination and build an environmental model efficiently the robots have to exchange information periodically. In *BERODE* each robot is in charge of distributing its information. It is not surprising, as Winfield [13] has shown, that the delay in the propagation of information distribution between a pair of nodes is proportional to the number of retransmissions. It is expected that for typical indoor environments pairs of robots with close positions are in either direct contact or within a small hop distance on the *MST control network*. Moreover, these close robots require a higher degree of coordination to guarantee that the *MANET* is kept connected. The robots transmit *beacon signals* that contain their position. They determine their *network status* based on the *beacon signals*.

To maximize the coordination and minimize communication costs, two levels of communication are proposed: local and global. The local level is composed of the robots in the *local network* (within *k*-hops) while the global level is composed of all the robots.

The robots transmit their current goal and recent *local features* to all the robots in their *local network* with a frequency less than that of beacon signals (typically 10%). The recent *local features* are the features that have been extracted since the last transmission at the local level. These features are used for planning purposes as temporary aids, but are not incorporated into the feature map. The *global features* are transmitted to all robots less frequently than local features to local robots, typically 25% less frequently. The *global features* are the features that have been extracted since the last transmission at the global level. These features are integrated to the robots' maps using the same process as for locally extracted features.

local robots maintain consistent environment models most of the time while distant robots have *weakly-consistent* maps. The exploration process is not impaired by the *weakly-consistent* maps of distant robots because they do not directly coordinate.

4 The Communication Models

The experiments were conducted in simulation using the Webots simulator [3]. To improve the realism of the simulation we derived sensor models from a real robot [11], and derived radio communication propagation characteristics from manufacturers data

[2]. Two types of communication have been modelled: *LOS* and *RF*. The following assumptions have been made in the implementation of these models: If two robots are within range of communication (*direct connection*) there is no loss of information; the communication bandwidth for the robot connections is large enough to cope with the exchange of information regardless of the robot positions; the delays in communication are proportional to the number of retransmissions required based on the *MST control network* regardless of the distance between the robots.

The effect of interference is modelled by delaying the messages a random time with a certain probability. In the *LOS* model any obstacle in the direct path of the signal blocks the entire signal. The *RF* model is based in Rappaport's model [5]. This model calculates the strength of a signal based on the *path loss* in decibels (dB). The *path loss* is the amount of power lost by a signal due to the transmission distance, the number of obstacles in the direct path of the signal and the properties of these obstacles (e.g. material and density). The experimental results for the attenuation of the *RF* signals for several materials (in dB/m) for a frequency λ=2.4 GHz are used in the simulations [2]. The multi path effects are modelled by adding Gaussian Noise (with mean zero) when there is no *LOS* between transmitter and receiver.

5 Comparison with Fixed Robot Networks

In *BERODE* the robots recalculate the *MST control network* either partially or globally. The recalculation of the network aids the exploration process because it enhances the connectivity of the network and adapts it to local geography. To determine the effectiveness of the adaptability of *BERODE* we compared its performance with several fixed networks. In fixed networks the *MST control network* is created at the start of the exploration and remains the same through all the process. The fixed networks use the same *BERODE* architecture.

Three types of fixed networks are proposed for the comparison: maximum, minimum and $k-$connections connectivity. The maximum connectivity tries to create star like topologies whereas the minimum connectivity tries to create column like topologies. The k-connections connectivity tries to create a network in which all the robots have k connections.

In the experiments the robots transmit their beacon signals every second. The process of sensing a location takes 1.8 seconds (sensing step). The robots share their local and global features every 10 and 40 sensing steps respectively. The size of the team of robots was n=4, $\ldots$, 16 for the environment of Fig. 1. The *local network* sizes were k= 2, 4, $\ldots$, k/2 for each team size. In our research we have found a trade-off between the exploration time and the size of the *local network* [10]. For small values of k the trade-off is optimal because there is a linear increase in the communication bandwidth as k increases. For larger values of k the increase tends to be quadratic.

We ran 10 trials for each combination of robot and *local network* size. Several environments were tested to validate the results. For the fixed networks the information is always transmitted at the global level.

Two metrics were used in the comparison: the speedup factor and the percentage of time fully connected. The speedup factor is the ratio between the exploration times for a single robot and a robot network of a certain size. The speedup factor describes the scalability of the control approach with respect to the number of robots.

Fig. 3 shows the results in one of the tested environments comparing *BERODE* using three *local network* sizes (*BRD k*) against four fixed networks: maximum connectivity (*MAX*), minimum connectivity (*MIN*) and $k-$connections for branching factors k=3,5 (*KC k*=3 and *KC*=5 respectively). Similar trends were observed when using the *LOS* communication model. From Fig. 3(a) it is observed that regardless of its *local network* size *BERODE* has significantly better speedup factors than the fixed networks. The closest speedup factor for a fixed network was 8.39% worse on average compared to *BERODE* with k=3. It is also observed that fixed networks with smaller branching factors (*MIN* and *KC k*=3) have better speedup factors than those with larger branching factors (*KC k*=5 and *MAX*). These networks have a linear increase in the speedup factor with respect to the number of robots up to a certain number of robots. Afterwards there is only a slight increase if not zero increase in the worst case.

From Fig. 3(b) it is observed that regardless of the *local network* size in *BERODE* the percentage time slowly decreases in a linear fashion as the number of robots increases until a certain number of robots is reached. Afterwards the percent of time stabilises at a certain percentage. It is also observed that the minimum connectivity type maintains similar if not better percentages than *BERODE*. 0.78±0.2% better than *BERODE* with k=8.

BERODE has a better performance than fixed networks. *BERODE* has significantly better speedup factors than the fixed networks and keeps the *MANET* connected more time than the fixed networks. Although not as efficient as *BERODE's* adaptive networks, fixed networks with column like control formations are a good solution for robot networks of medium sizes (n <12) using *RF* technologies. These networks are suitable for indoor environments with little clutter and might be preferred over *BERODE* because of their simplicity.

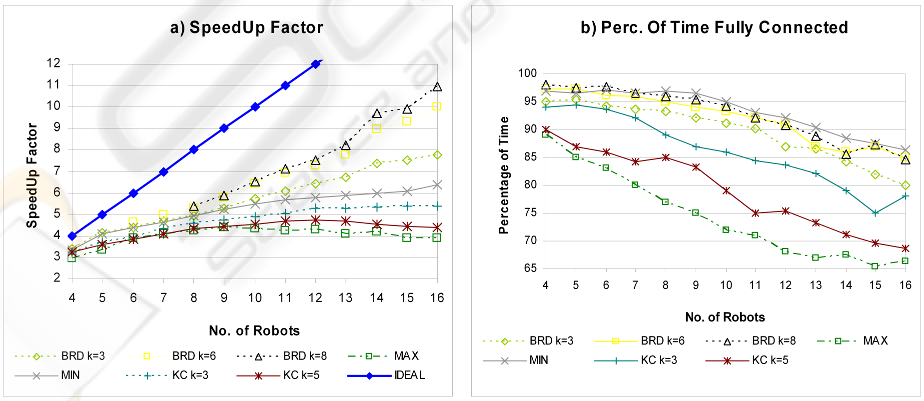


Fig. 3. Comparison of $BERODE$ with $local\ network$ sizes $k = 3, 6, 8$ against fixed networks using the RF model.

6 Conclusions

In this paper, we presented the *BERODE* architecture to explore and map an initially unknown environment using a group of robots with local communication capabilities. The robots are kept as a single connected and adaptable communication network to guarantee the coordination between the robots. *BERODE* is scalable with respect to communication because it implements a hierarchical approach to distributing information. Note that this is hierarchical broadcasting within an adaptive decentralised system, and involves no loss of robustness. We presented experimental simulations that assumed two types of communication: *LOS* and *RF*. In the *LOS* communication any obstacle in the path of the signal blocked the signal while in the *RF* model a part of the signal is absorbed by the obstacles.

BERODE maintains the communication network by creating and updating an *MST control network*. This network is updated to improve the *signal quality* of its connections. Experiments showed that *BERODE* explored the environments more efficiently than robot teams that implement a fixed control network. *BERODE* maintained the network fully connected for more time than the fixed networks. In future research we plan to validate these encouraging results in real environments.

References

1. George A Kantor, Sanjiv Singh, and R. Peterson. Distributed search and rescue with robot and sensor teams. In *The 4th Int'l Conf. on Field and Service Robotics*, July 2003.
2. MaxStream. *Indoor Path Loss: Application Note*. MaxStream, 2003.
3. O. Michel. Webots: Professional Mobile Robot Simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.
4. Matthew Powers and Tucker Balch. Value-based communication preservation for mobile robots. In *International Conference on Robotics and Automation*, 2004.
5. T. S. Rappaport. *Wireless Communication: Principles and Practice*. Prentice Hall, 2nd edition, 2001.
6. R. Simmons, W. Burgard, D. Fox, and S. Thrun. Coordination for multi-robot exploration and mapping. In *Proc. of the Nat'l Conf. on Artificial Intelligence*, pages 852–858, 2000.
7. B.J. Thibodeau, A.H. Fagg, and B.N. Levine. Signal strength coordination for cooperative mapping. Technical Report 04-64, University of Massachusetts Amherst, July 2004.
8. S. Thrun. Robotic Mapping: A Survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*, pages 1–35. Morgan Kaufmann, 2002.
9. P. Ulam and R. Arkin. When good communication go bad: communications recovery for multi-robot teams. In *Proc. of the Int'l. Conf. on Robotics and Automation*, pages 3727–3734, 2004.
10. Jose Vazquez. *The BERODE architecture*. PhD thesis, University of Edinburgh, 2006.
11. Jose Vazquez and Chris Malcolm. Fusion of triangulated sonar plus infrared sensing for localization and mapping. In *Proc. of the 2nd Int'l IEEE Conf. on Int. Systems*, June 2005.
12. Alan Wagner and Ronald Arkin. Multi-robot communication-sensitive reconnaissance. In *Proc. of the IEEE Int'l Conf. on Intelligent Robotics and Systems*, pages 4674–4681, 2004.
13. Alan FT Winfield. Distributed sensing and data collection via broken ad hoc wireless networks. In *Distributed Autonomous Robotic Systems*, volume 4, pages 273–282, 2000.