Aerial Shepherds: Coordination among UAVs and Swarms of Robots

Luiz Chaimowicz and Vijay Kumar

GRASP Laboratory – University of Pennsylvania
3330 Walnut. St. - Levine Hall - Philadelphia, PA, 19014.
{chaimo, kumar}@grasp.cis.upenn.edu

Abstract - We address the problem of deploying groups of tens or hundreds of unmanned ground vehicles (UGVs) in urban environments where a group of aerial vehicles (UAVs) can be used to coordinate the ground vehicles. We envision a hierarchy in which UAVs with aerial cameras can be used to monitor and command a swarm of UGVs, controlling the splitting and merging of the swarm into groups and the shape (distribution) and motion of each group. We call these UAVs *Aerial Shepherds*. We show a probabilistic approach using the EM algorithm for the initial assignment of shepherds to groups and present behaviors that allow an efficient hierarchical decomposition. We illustrate the framework through simulation examples, with applications to deployment in an urban environment.

1 Introduction

The use of Unmanned Aerial Vehicles (UAVs) in concert with Unmanned Ground Vehicles (UGVs) affords a number of synergies. First, UAVs with cameras and other sensors can obtain views of the environment that are complementary to views that can be obtained by cameras on UGVs. Second, UAVs can fly over obstacles while keeping UGVs in their field of view, providing a global perspective and monitoring the positions of UGVs while keeping track of the goal target. This is especially advantageous in two and a half dimensions where UAVs can obtain global maps and the coordination of UAVs and UGVs can enable efficient solutions to the mapping problem. Third, if UAVs can see the UGVs and the UGVs can see UAVs, the resulting three-dimensional sensor network can be used to solve the simultaneous localization and mapping problem, while being robust to failures in sensors like GPS and to errors in dead reckoning. In addition to this, the use of air and ground robotic vehicles working in cooperation has received a lot of attention for defense applications because of the obvious tactical advantages in such military operations as scouting and reconnaissance.

We are interested in deploying teams of heterogeneous vehicles in urban environments to perform tasks such as searching for targets or mapping the environment. We are particularly interested in having several UAVs shepherding large groups of UGVs. We are motivated by our experience with deploying a team of aerial and ground vehicles at a Military Operations on Urban Terrain (MOUT) site at Fort Benning. The MOUT site is a replica of a small city consisting of 17 two and three store buildings, streets and access roads. It is engineered with cameras that allow a multiple view tracking of training missions. It also features a small airfield, being a suitable test ground for air-ground cooperation. Our multi-robot team [9] consists of a 30 foot unmanned blimp and ten ground vehicles. Figure 1 depicts our blimp and two of our ground robots performing a leader-follower task at the MOUT site.

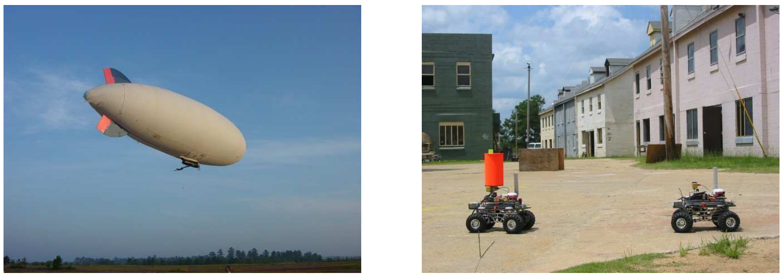

Fig. 1. Blimp and two ground robots during the experiments at the MOUT site.

In this paper, we focus on a scalable approach for deploying tens and hundreds of ground vehicles in order to search the urban terrain for targets or simply to disperse and perform a coverage task. We propose a paradigm in which a group UAVs, equipped with cameras, can act as aerial shepherds, monitoring and commanding a swarm of UGVs. Differently from our previous work [5], where we needed a central planner for controlling the robots, in this paper we present behaviors that allow the UAVs to coordinate the splitting and merging of the swarm into groups and control the shape and motion of each group in a distributed manner. We also propose a probabilistic approach using the EM algorithm for the initial assignment of shepherds to groups and illustrate this architecture with simulation examples.

It is important to mention that in spite of the growing number of works in cooperative robotics, few tackle specifically the cooperation between UAVs and UGVs. A general application is to use the UAVs as an "extra sensor" for the UGVs. In this context, Stenz et al. [12] present an application where an autonomous helicopter helps the navigation of an UGV by exploring the terrain and informing the UGV about potential hazards (holes, obstacles, etc.) on its way. Other important applications include environmental monitoring, cooperative control, and cooperative localization. Elfes et al. [8] and Lacroix et al. [11], for example, present some preliminary ideas on the integration of blimps and ground robots for environmental surveillance and other tasks. Sukhatme et al. [13] propose an architecture for coordinating an autonomous helicopter and a group of ground vehicles using decentralized behavior based controllers and minimal top down planning. Sukkarieh et al. [14] present a decentralized architecture for data fusion and control of multiple UAVs and UGVs. In our previous work, we report on coordination between aerial and ground vehicles for robust localization in the presence of GPS errors [4].

2 Architecture

We developed a hierarchical architecture to allow a group of UAVs to coordinate and control swarms of ground robots. At the lowest level, there are individual robots i, $i = 1, \ldots, N_r$. They are assumed to be holonomic and the state of each robot is described by its cartesian coordinates:

$$\begin{aligned} X^i &= (x^i, y^i), \\ \dot{X}^i &= u^i = f(X^i, a^j). \end{aligned}$$

Robots are assembled together in a set of groups Γ. Each group $\gamma^j \in \Gamma$, $j = 1, \ldots, N_g$ is modeled by a double (g^j, ρ^j): g is an element of the Lie group $SE(2)$ while ρ is the shape of the group formation [7]. Thus each group γ^j can be represented by an abstraction a^j that comprises the group pose g^j and the group shape ρ^j [1]. More specifically:

$$\begin{aligned} a^j &= (g^j, \rho^j), \\ g^j &= (\mu_x^j, \mu_y^j, \theta^j), \\ \rho^j &= (s_1^j, s_2^j). \end{aligned}$$

This abstraction provides a scalable way of controlling groups of robots. It can be viewed as an *equipotential* or *concentration ellipse* centered in μ with orientation θ and principal axis given by $\sqrt{cs_1}$ and $\sqrt{cs_2}$. The number c is given by $c = -2\ln(1-p)$, where p is the percentage of a large number of normally distributed robots that lies inside the ellipse. The abstraction a^j of a certain group γ^j is computed using only the states of the robots that belong to this group, as explained in [1].

The third level of the hierarchy is composed by medium altitude UAVs (blimps) that hover over the groups. Every group must have a shepherd blimp but one blimp can escort one or more groups simultaneously (see Figure 2). Let's consider that each blimp k, $k = 1, \ldots, N_b$ is shepherding a subset of groups S^k, $S^k \in \mathcal{P}(\Gamma)$, where $\mathcal{P}(\Gamma)$ is the power set of Γ. In this paper, we adopt a simple kinematic model for the blimps that considers its position (τ, η, z) in the 3D space and its yaw angle (ϕ). More complex dynamic models of our blimp are derived in [10].

$$\begin{aligned} Z^k &= (\tau^k, \eta^k, z^k, \phi^k,), \\ \dot{Z}^k &= v^k = f(Z^k, S^k). \end{aligned}$$

Basically, after an initial assignment phase, the blimp controller tracks the shape and pose of the groups in S^k, trying to keep all the robots of these groups inside the blimp's visibility region. For now, we are considering that this visibility region is a circle with radius varying proportionally to the blimps' altitude z and that all the robots inside this circle can be localized by the blimp. The blimps have a superior limit for their altitude (z_{max}) and, consequently, the maximum visibility area is also limited.

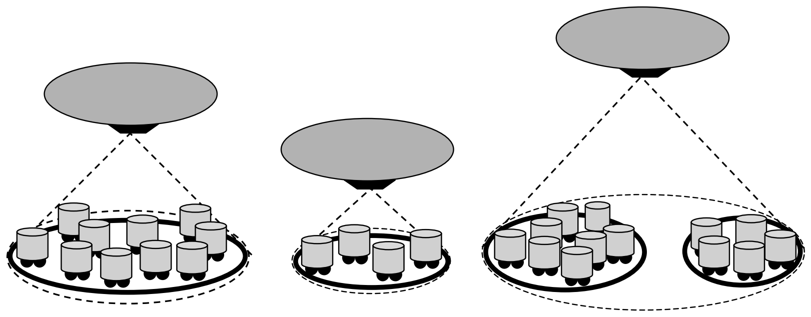

Fig. 2. Hierarchical architecture in which three blimps shepherd groups of UGVs.

The controllers used by each of these hierarchical levels are described in detail in [1] and [5]. Basically, the ground robots only require state feedback and information about the pose and shape of its own group. The group's pose and shape variables are controlled by the blimps based on the robots' positions using simple proportional equations. Finally, the blimp controller tracks the pose and shape of its groups to keep the robots inside its field of view.

One of the main advantages of this architecture is that the amount of information that must be exchanged among the different hierarchical levels is small. Blimps have to broadcast to the ground robots the pose and shape of their groups and the desired group velocities (basically, a^j and $\dot{a}^j$). Other than that, as will be explained in the next section, ground robots broadcast their position once for the initial distribution and respond sporadically to inquires from blimps regarding its shepherd during the merge process.

The communication within the hierarchical levels is also small. As mentioned, the ground robots only require state feedback and information about their own group. No communication is necessary internally among them. Blimps have to communicate with each other to coordinate the split and merge behaviors. But this is not frequent and the number of blimps is much smaller than the number of ground robots. Thus, the amount of information exchanged between the ground vehicles and the blimps increases linearly with the number of robots making the proposed architecture scalable. Moreover, the information being broadcast is sparse: only information about the groups' poses and shapes is being broadcast, regardless of how big the groups are.

3 Coordination

3.1 Initial Distribution

We consider that the initial position of the robots can be described by a *Mixture of Gaussians*. A mixture of gaussians is a distribution Θ composed by m components, each of which is a gaussian distribution in its own right. Each gaussian have its own mean and covariance (μ,Σ), and has a weight w in the mixture Θ. Ideally, each individual gaussian will represent one group γ^j of the hierarchy, and will be shepherded by one blimp. Initially, we do not know which robot belongs to each component and, consequently, the parameters of each gaussian are unknown. In the approach used here, these parameters are determined by the blimps through the *EM – Expectation Maximization* algorithm [6]. The EM algorithm was developed to compute the maximum likelihood estimate of the parameters of a distribution from a given data set when the data is incomplete or has missing values. But one of its main uses is when the optimization of the likelihood function is intractable, but the function can be simplified considering its data incomplete [2]. This is exact the case of mixtures of gaussians.

Thus, the initial distribution of robots into groups and the assignment of blimps to the groups is done as follows. Initially all robots broadcast their position once to the blimps. Each blimp apply the EM algorithm to determine the distribution parameters and allocate itself to one of the computed distributions. Since the blimps have the same data and run the same algorithm they will obtain the same results, and the assignment can be done in a sequential way: blimp b_1 to the first distribution, blimp b_2 to the second, etc. Each distribution determined by the algorithm will become a group in the hierarchy.

The only drawback of this method is that the number of distributions (components of the mixture) in which to divide the robots should be given a priori to the algorithm. We consider that this number is equal to the number of available blimps (N_b). The problem is that this number can be different of the number of "real distributions" (N_d). For example, if we have the robots distributed in three gaussians ($N_d = 3$) but we have four blimps ($N_b = 4$), the algorithm will divide one of the real distributions in two, resulting in four groups instead of three. So, the ideal situation will be when the number of blimps is equal the number of distributions ($N_b = N_d$). If $N_b < N_d$ there will be one blimp allocated to more than one distribution and if $N_b > N_d$ there will be two or more blimps allocated to a single distribution. In these two situations, the groups will probably have to split or merge, as will be explained in the next section. Figure 3 exemplify these three scenarios. In this figure, the small circles represent the robots, the dashed circles represent the blimps and the large solid circles represent the visibility area of each blimp. There are basically 3 robot distributions and 2, 3 and 4 blimps respectively in the pictures.

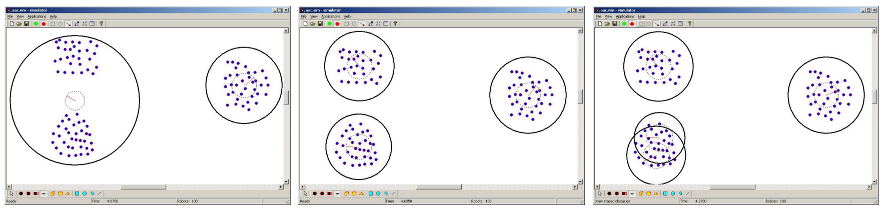

Fig. 3. Initial distribution when: a) $N_b < N_d$, b) $N_b = N_d$, c) $N_b > N_d$.

3.2 Merge and Split

The ability of splitting and merging groups is a desired behavior in several different scenarios. Basically, one group may divide into two or more groups and different groups can merge into a single one. For example, a robot group may have to split to avoid obstacles or to cover different regions as shown in Figure 4. Also, split and merge behaviors can be used to fix an unbalanced initial distribution as mentioned in the previous section.

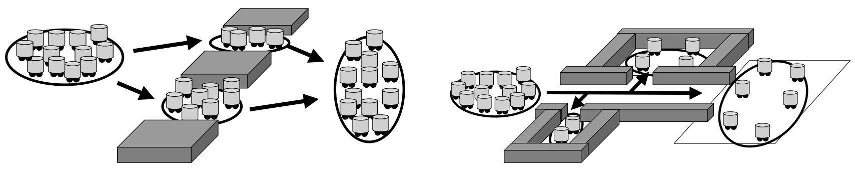

Fig. 4. Scenarios where groups of robots perform split and merge behaviors for avoiding obstacles and exploring environments.

In this paper, we use a deliberative approach where each blimp has a plan for its group. This plan is represented by a directed graph in which each node contains a desired pose and shape for the group (g_{des}, ρ_{des}) and the edges indicate the ordering of the goals. Here, these plans are specified manually, but variations of automated techniques such as grid based approaches or voronoi diagrams could be used.

Merge

The merge process occurs when two groups that are heading to the same goal are close enough to be shepherded by a single blimp. In this case, one of the blimps will escort all the robots while the other blimp will be dismissed and will return to its base. Let's consider two groups γ_1 and γ_2 moving to a common goal and being shepherded by two blimps b_1 and b_2 respectively. When a robot $i \in \gamma_2$ is detected inside the visibility region of b_1, b_1 will create and start to track a *virtual group* $\gamma_{1,2}$ composed by the robots of both groups. When all robots from γ_2 are inside the visibility region of b_1, the two groups are merged into γ_1, γ_2 is removed from the hierarchy, and the blimp b_2 is dismissed. The blimps must coordinate themselves to do the merging process, but, as mentioned, the amount of communication necessary for this is very small. Basically, b_1 inquires robot i to discover who is its shepherd blimp. Then b_1 and b_2 communicate to check if they are going to the same goal, and if this is the case, b_1 creates the virtual group and start tracking it, receiving information from b_2 regarding the pose and shape of its group. To track both groups, b_1 will probably have to fly at a higher altitude in order to increase its visibility area. This will be demonstrated in the simulations of Section 4. When all robots are inside its visibility region, b_1 finishes the merging processes: it broadcasts a message to the robots of γ_2 informing them that they now belong to a new group and have a new shepherd, and also sends a message to b_2 releasing it from its group.

Split

The split processes is initiated by a blimp when its group reaches some predetermined goals according to its plan. If the out-degree (q) of a certain node (goal) in its plan is greater than 1, the group will split in q subgroups. The shepherd blimp is responsible for deciding how many and which robots should be assigned to each of the new groups, and for requesting help from other blimps to escort the new groups. Let's consider that a group, γ_1, shepherded by blimp b_1 has reached a goal and must be split in two groups γ_2 and γ_3, that will move to different goals. The blimp will compute the number of robots (n_j) that should be assigned to each group γ^j based on the shape of the next desired goals on the graph. Then, the n_j robots that are closer to one of the next goals are assigned to the group that will move towards that goal. After this division, b_1 will create a virtual group $\gamma_{2,3}$ and start escorting both groups. At the same time, b_1 will broadcast a message to the other blimps to see if there is an available blimp that can shepherd one of the new groups. If there is such blimp, say b_2, it communicates with b_1, receives information about the pose and shape of the new group, and starts moving towards it. If there is no other blimp available or if b_2 takes too much time to reach the group, probably b_1 will reach its maximum altitude if the groups are moving apart. In this situation, b_1 will abandon one of the groups and shepherd the other to its goal. The abandoned group will stop and wait for the arrival of b_2 or for the return of b_1, in the case that no other blimps were available.

4 Experiments

The proposed architecture was implemented and tested using MuRoS, a multi-robot simulator that we have developed for cooperative robotics [3]. Implemented for the MS Windows environment, MuRoS has a friendly user interface and can be easily extended with the development of new inherited classes defining new robots, controllers and sensors.

The simulation presented here demonstrates the initial distribution and the merge and split behaviors described in the previous section. Figure 5 shows some snapshots of this simulation where 50 robots are shepherded by two blimps. Again, the small circles represent the robots, the dashed circles represent the blimps and the large solid circles represent the visibility area of each blimp, which varies with its altitude z. The two rectangles are obstacles and the (barely visible) crosses represent the group's current and desired poses and shapes. The robots are initially divided in two groups (Figure 5(a)), that are allocated to two blimps after the application of the EM algorithm. The blimps start to shepherd the groups towards the first goal (b) and, when the groups are sufficiently close, they merge and become escorted by a single blimp, while the other return to its base (c). After moving to the second goal (Figure 5(d)), the group have to split to move to different goals. In this case, as mentioned, the blimp creates a virtual group and tries to shepherd both groups until the arrival of a second blimp (e). The last snapshot (f) shows the two blimps shepherding their groups to their final destinations.

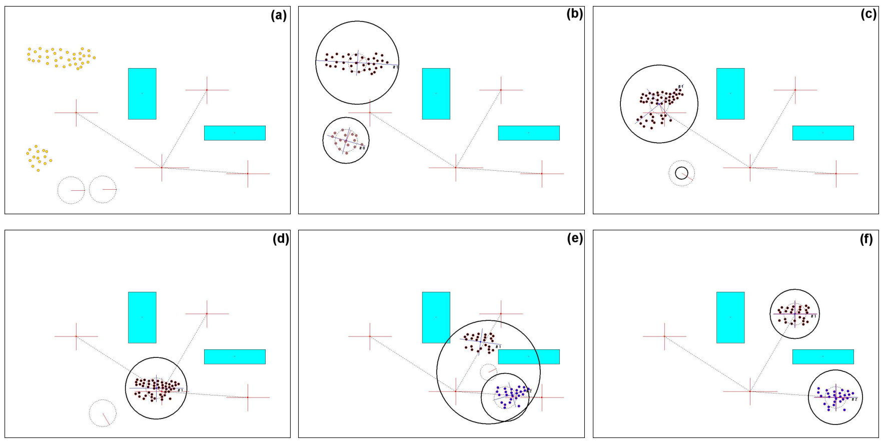

Fig. 5. Simulations where two UAVs coordinate a group of 50 UGVs.

The graph of Figure 6 shows the altitude (z) of both blimps (b_0 and b_1) during the simulation. The labels T_m and T_s mark the times in which the groups merged and split. Basically, the blimps initially fly from the base to their initial groups and start to shepherd them to the first goal. As shown in Figure 5(b), the group on the top, that is shepherd by b_0, is larger and has to compress its shape en route to the first goal while the group on the bottom has to expand. Consequently, b_0 can decrease its altitude while b_1 must fly higher in order to escort its group. This can be observed in the graph of Figure 6, just after time 5. Just before time T_m, b_0 creates a virtual group and increases its altitude to track both groups. After the merging, b_0 lowers its altitude again while shepherding the merged group to the first goal and b_1 returns to its base and lands. The merged group shepherd by b_0 then moves to the second goal where it should split. After the split (T_s), b_0 has to increase its altitude again to shepherd both groups while b_1 takes off and goes after one of the groups to help b_0, as can be seen on Figure 5(e). Finally, b_1 takes over one of the groups, b_0 decreases its altitude, and both blimps stabilize and escort their groups to their final goal.

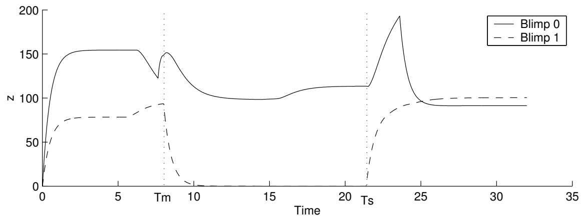

Fig. 6. Blimps' altitude (z) during the execution of the task.

These results, in spite of not providing rigorous measurements of performance, demonstrated that this hierarchical architecture can be used for coordinating UAVs and swarms of UGVs in a scalable manner. It is important to mention that other simulations were performed on different scenarios, using a different number of robots, blimps, and groups, also leading to similar results.

5 Conclusion

In this paper, we presented a hierarchical architecture in which a few number of UAVs is used to command, control and monitor swarms of UGVs. Basically, the individual UGVs are assembled together into groups that are shepherded by medium altitude UAVs (blimps). The initial assignment of blimps to groups is done in a distributed manner using the EM algorithm and the blimps are responsible for controlling the groups' shape, pose and motion. An important feature of this architecture is that the communication requirements grow linearly with the number of vehicles, making it scalable to tens and hundreds of robots. We described behaviors that allow blimps to coordinate the splitting and merging of groups during the task execution and demonstrated these concepts through simulations.

Future work is directed towards improving this architecture to deal with some specific situations. For example, if one group is too large to be shepherded by a single blimp, we would like to have two blimps coordinating themselves in order to escort this group. We are also interested in executing simulations to obtain more detailed performance measurements of this architecture, mainly in regard to communication and control.

References

1. C. Belta, G. A. S. Pereira, and V. Kumar. Abstraction and control for swarms of robots. In *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, 2003.
2. J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation and for gaussian mixture and hidden markov models. Technical Report TR-97-021, Dept. of Computer Science – UC Berkeley, 1998.
3. L. Chaimowicz, M. Campos, and V. Kumar. Simulating loosely and tightly coupled multi-robot cooperation. In *Proceedings of V SBAI – Brazilian Symposium on Intelligent Automation*, November 2001.
4. L. Chaimowicz, B. Grocholsky, J. F. Keller, V. Kumar, and C. J. Taylor. Experiments in multirobot air-ground coordination. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 4053–4058, 2004.
5. L. Chaimowicz and V. Kumar. A framework for the scalable control of swarms of unmanned ground vehicles with unmanned aerial vehicles. In *Proceedings of the 10th International Conference on Robotics and Remote Systems for Hazardous Environments*, 2004.
6. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.
7. J. Desai, J. Ostrowski, and V. Kumar. Modelling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.
8. A. Elfes, M. Bergerman, J.R.H. Carvalho, E.C. Paiva, J.J.G. Ramos, and S.S. Bueno. Air-ground robotic ensembles for cooperative applications: concepts and preliminary results. In *Proceedings of the International Conference on Field and Service Robotics*, pages 75–80, 1999.
9. D. G. Ibanez, B. Grocholsky, F. Hager, J. F. Keller, J. W. Kim, V. Kumar, P. Srivastava, and C. J. Taylor. An experimental test-bed for air ground coordination. Submitted to the 2004 American Control Conference, 2004.
10. J. W. Kim, J. F. Keller, and V. Kumar. Design and verification of controllers for airships. In *Proceedings of the 2003 IEEE International Conference on Intelligent Robots and Systems*, pages 54–60, 2003.
11. S. Lacroix, I-K. Jung, A. Mallet, and R. Chatila. Towards cooperative air/ground robotics: issues related to environment modeling. In *10th International Conference on Advanced Robotics*, Budapest (Hungary), 2001.
12. T. Stentz, A. Kelly, H. Herman, P. Rander, and R. Mandelbaum. Integrated air/ground vehicle system for semi-autonomous off-road navigation. In *Proceedings of AUVSI Symposium on Unmanned Systems*, 2002.
13. G. S. Sukhatme, J. Montgomery, and R. T. Vaughan. Experiments with aerial-ground robots. In T. Balch and L.E. Parker, editors, *Robot Teams: From Diversity to Polymorphism.* AK Peters, 2001.
14. S. Sukkarieh, B. Grocholsky, E. Nettleton, and H. F. Durrant-Whyte. Information fusion and control for multiple UAVs. In *NRL Workshop on Multi-Robot Systems*, volume 2. Kluwer, March 2003.