Coordination of Multiple UAS for Tracking under Uncertainty

Jesús Capitán[1], Luis Merino[2] and Aníbal Ollero[1,3]

[1] *University of Seville, Camino de los Descubrimientos s/n, 41092, Seville, Spain*

[2] *Pablo de Olavide University, Crta. Utrera, km. 1, 41013, Seville, Spain*

[3] *Center for Advanced Aerospace Technology, C. Wilbur y Orville Wright 17-19-21, 41309, La Rinconada, Spain*

Abstract

This paper presents how *Partially Observable Markov Decision Processes* (POMDPs) can be used for controlling fleets of UAS under uncertainty[1]. POMDPs provide a sound mathematical framework to deal with planning actions when tasks outcomes and perception are uncertain, although their computational complexity have precluded their use for multi-robot applications. However, in this work a scalable approach based on POMDPs is presented. Instead of solving a complex model for the whole team, distributed models in which the UAS have no knowledge about others' actions are considered. Then, a decentralized data fusion algorithm is used to share information from all the sensors and obtain an implicit coordination between the UAS. Moreover, the proposed approach is applied to a tracking application with a fleet of UAS, and some simulations are presented to show its advantages and how the mentioned coordination arises between the members of the team.

keywords: coordination, multiple UAS, decentralized data fusion

1 INTRODUCTION

Many applications with UAS, such as tracking, surveillance or environmental monitoring, require to deal with uncertainties. For instance, in UAS surveillance, where one UAS or a team of UAS should monitor an area and track potential targets of interest, imperfect sensors, incomplete models and other sources of uncertainty require taking into account the uncertainty on the target's position when planning the best actions [5]. In addition, if the application involves several UAS with different sensors, the cooperation between vehicles is also desirable.

Techniques for decision-theoretic planning under uncertainty are becoming more popular in the field of cooperative robotics. For instance, in [15, 17] different probabilistic approaches for mapping and tracking with robotic networks are presented. [16, 2] also propose decentralized Bayesian approaches for multiple-target optimal search with UAS. Usually, these approaches are greedy, in the sense that only the next action is considered, and no planning is involved. Additionally, [4, 8] make a distinction between cooperative and coordinated information-theoretic solutions, and propose the latter to control fleets of

[1]This work has been partially supported by the EU-funded project EC-SAFEMOBIL (FP7, ICT-288082)

Figure 1: Multiple UAS can cooperate in tasks like surveillance and tracking. Left: two autonomous helicopters in a surveillance mission outdoors. Right: three quadrotors following targets in an indoor testbed. Robust mission execution requires to take into account uncertainties about the target's position and motion model.

UAS. In a coordinated approach, the members of the team have no knowledge about the others' models or control actions, but they exchange some information that may influence implicitly other members' subsequent decisions. Hence, sharing fused perception information or the impact of others' control actions over a certain objective function, and acting locally, a coordinated behavior can be obtained.

Within a Bayesian framework, Partially Observable Markov Decision Processes (POMDPs) provide an elegant way to model the interaction of a team of UAS with their environment. Based on prior knowledge of the sensors and actuators models and the environment dynamics, policies which indicate the UAS how to act can be computed. These policies can be extracted by optimizing iteratively a certain value function over the belief space. The main problem is that the information space (the set of potential actions and observations), and hence the computational complexity of POMDPs increases dramatically with the number of UAS in the team.

There exist approximate methods to obtain POMDP policies that can deal with large belief spaces. In this sense, point-based algorithms [9, 14, 12, 13] represent the value function by a set of vectors over the belief space (α-vectors) and restrict the optimization procedure to a bounded set of feasible beliefs. Particularly, [14] propose a point-based solver, called Perseus, where no computation is needed for all the belief points at every iteration, hence improving performance. In [10], an extension of Perseus is presented. It is called *Symbolic* Perseus and uses Algebraic Decision Diagrams (ADDs) [6] in order to optimize the operations in the original Perseus for factored POMDPs. Besides, [7] propose SARSOP, which maintains a tree-shaped set of reachable beliefs that is expanded at every iteration using the best policy so far.

However, the above methods still find hard restrictions when applied to multiple UAS, since the space of actions and observations grows exponentially together with the belief space. This leads to very complex POMDPs whose policies are highly difficult to compute but for the simplest problems. Hence, this paper describes a different method that can overcome this issue for certain applications. In particular, instead of solving a large POMDP considering all the UAS, a decentralized approach where each UAS acts without any knowledge about the others' actions is proposed. Then, an implicit coordination is achieved (according to the idea in [4]) by sharing a fused belief of the state among the team members. Moreover, sensors properties and desired objectives could be changed from one UAS to another in order to combine different behaviors within the team.

The application of decision-making under uncertainty that is addressed in this paper is target tracking with a team of UAS. Bearing-only sensors (cameras) on board the UAS are considered to detect the target. In this case, a single sensor does not provide enough information about the distance to the target, what makes tougher a proper tracking. Moreover, the motion of the target is not predictable. Thus, a team of UAS can act together in order to achieve a more efficient performance.

The paper is structured as follows: Section 2 gives some theoretical background about POMDPs; Section 3 presents a Decentralized Data Fusion approach in order to estimate a shared belief; Section 4 details the coordinated tracking approach for multiple UAS; and Sections 5 and 6 explain the experimental results and conclusions, respectively.

2 POMDP MODEL

Formally, a POMDP is defined by the tuple $\langle S, A, Z, T, O, R, h, \gamma \rangle$. The meaning of every component is the following:

- *State space*: All the information available of the environment is encoded in the state, which is not fully observable. The system's state at time step t is defined by $s_t \in S$, where S is the finite set of all the possible states.

- *Action space*: Every UAS can take an action each time step. A is the finite set of possible actions, whereas $a_t \in A$ is the action taken at a certain time step.

- *Observation space*: Given a time step t, after executing an action, a UAS can make a measurement or observation $z_t \in Z$, where Z is the finite set of all the possible observations.

- *Transition function*: After executing an action, the state can vary probabilistically. This probability density function is modeled by the transition function $T : S \times A \times S \longrightarrow [0, 1]$, where $T(s', a, s) = p(s_t = s' | a_{t-1} = a, s_{t-1} = s)$. It represents the probability of ending in the state s' if the UAS performs the action a in the state s.

- *Observation function*: The observations gather information from the current state and are related probabilistically to this state. This probability density function is modeled by the observation function $O : Z \times A \times S \longrightarrow [0, 1]$, where $O(z, a, s') = p(z_t = z|a_{t-1} = a, s_t = s')$. It gives the probability of observing z if the action a is performed and the resulting state is s'.

- *Reward function*: A POMDP selects the best actions so that a reward function is optimized. Thus, the behavior of the system is determined by this function. $R : S \times A \longrightarrow \Re$ is the reward function, where $R(s, a)$ is the reward obtained by executing action a in state s. Here, the reward is assumed to be bounded. Since it allows to model quite complex goals, the reward function is a very powerful tool, and hence, its right design is crucial.

- *Horizon and discount factor*: Then, the goal of a UAS is to maximize the expected reward earned over some time frame. The horizon h defines this time frame by specifying the number of time steps the UAS must plan for. Thus, the objective is to maximize the sum $E\left[\sum_{t=0}^{h} \gamma^t r_t\right]$, where r_t is the reward at time t, $E[\ ]$ is the mathematical expectation, and $\gamma \in [0, 1)$ is a discount factor, which ensures that the sum is finite when $h \to \infty$.

Given that it is not directly observable, the actual state cannot be known by the UAS. Instead, the information about the environment must be encoded in a *belief distribution*, which indicates a probably density function over the state space. Moreover, under the Markov assumption, this belief can be updated (Bayesian filter) recursively using the previous belief $b_{t-1}(s)$, the most recent action a and the most recent observation z:

$$b_t(s') = \eta O(z, a, s') \sum_{s \in S} T(s', a, s) b_{t-1}(s) \tag{1}$$

where η acts as a normalizing constant such that b_t remains a probability distribution.

Once the UAS has computed the belief state at a certain time, the best action is determined by a policy $\pi(b)$, which maps all the possible beliefs to their optimal actions. The policy is computed in such a way that it maximizes the value function:

$$V^\pi(b) = E\left[\sum_{t=0}^{h} \gamma^t r(b_t, \pi(b_t))|b_0 = b\right] \tag{2}$$

where $r(b_t, \pi(b_t)) = \sum_{s \in S} R(s, \pi(b_t)) b_t(s)$.

3 DECENTRALIZED DATA FUSION

In the multi-UAS case, maintaining a belief over the state space according to (1) is not trivial. Given a team with N UAS, a centralized node with access to all the information would update the belief as follows:

$$b_t^{cen}(s') = \eta O(z^J, a^J, s') \sum_{s \in S} T(s', a^J, s) b_{t-1}^{cen}(s) \tag{3}$$

where $a^J = \langle a^1, \cdots, a^N \rangle$ is the joint action and $z^J = \langle z^1, \cdots, z^N \rangle$ the joint measurement. However, if the belief estimation is decentralized and each UAS i uses only its local information (action a^i and observation z^i), some communication must be allowed among the UAS so that they can recover this centralized belief locally [11].

The *conditional independence* assumption of the measurements (given the state at that instant s') is typical in Bayesian data fusion. This is reasonable when the measurements obtained by each agent do not depend on the state of the other agents. Therefore, assuming this independence ($O(z^J, a^J, s') = \prod_i O(z^i, a^i, s')$) and assuming that actions are known when predicting, it is possible to combine locally received beliefs from other UAS with the one from UAS i, $b_t^i(s')$, to recover the centralized belief:

$$b_t^{cen}(s') \propto b_t^i(s') \prod_{j \neq i} \frac{b_t^j(s')}{b_t^{ij}(s')} \tag{4}$$

Equation 4 fuses the belief in UAS i with the one received from j by multiplying them. The common information previously exchanged by the UAS $b_t^{ij}(s')$ must be removed not to count it twice. This common information can be maintained by a separate filter called channel filter. Further details about the channel filter and the equation above can be seen in [1].

It has been shown [3] that if the state is dynamic, it is possible to obtain locally the same belief as in a centralized node with access to all the information available, by including delayed states in the belief. This represents a common belief signal for all the team members. If only the current state is considered, some information is lost with respect to an ideal centralized fusion unless the UAS communicate every time they gather new information [1].

4 COORDINATED TRACKING WITH MULTIPLE UAS

An application for tracking a target by means of multiple UAS is considered here. In this problem there is a moving target and a team of N hovering-capable UAS (like an helicopter or a quadcopter) which are the pursuers (see Fig. 1). Each of these UAS carries a camera in its nose which determines whether the target is visible or not (see Fig. 2, right). The UAS can communicate among themselves by wireless links. Then, the objective is to find the target in the environment and localize it as well as possible.

The state is composed of the position of the target (t_l) and the position of the pursuers ($p_l^1, \cdots, p_l^N$). Since bearing-only sensors are used, the headings of every UAS ($p_h^1, \cdots, p_h^N$) are also considered and included in the state. Moreover, the space is discretized into a cell grid, and a map of the scenario is assumed to be known. Thus, an occupancy grid can be obtained indicating which cells are attainable and which are non-flying zones. Then, the locations of the target and every pursuer are specified by cells, and just non-occupied cells are possible values. There are also four possible headings for every robot: *north*, *west*, *south* or *east*.

Every sensor provides a boolean measurement: *detected* or *non-detected*. These sensors proceed as it follows, if the target is out of its field of view, the sensor produces a *non-detected* measurement. However, when the target is within its field of view, it can be *detected* with a probability p_D.

In addition, each UAS can choose at each time step between four possible actions: *hover*, *turn right*, *turn left* or *go forward*. *hover* explains itself; when *turning*, the robot changes its heading 90^o degrees;

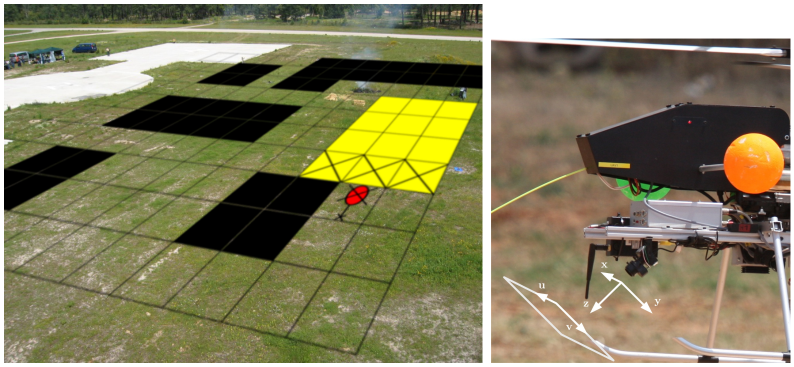


Figure 2: Simulated environment (left). The black cells are non-flying zones, whereas the yellow ones represent the field of view of the camera on board the UAS (right). If the target is in one of the cells with crosses, a high reward is obtained.

and when *going forward*, it moves to the cell ahead.

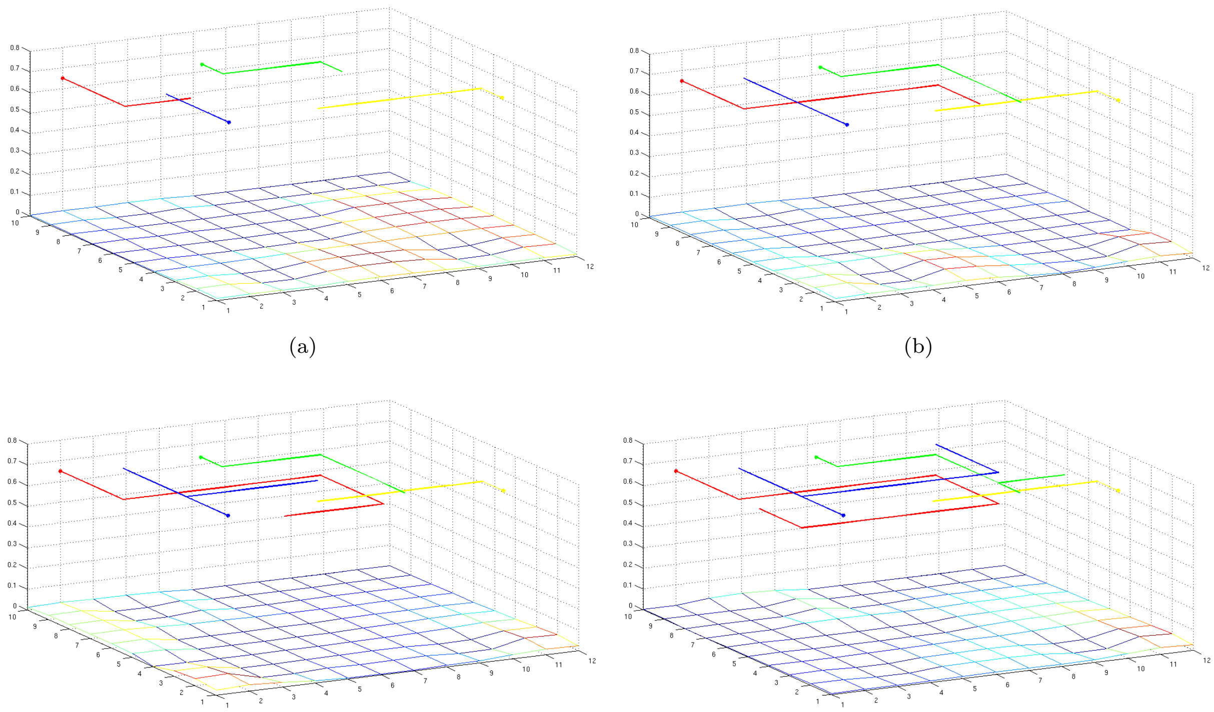
Finally, the key point is how to design a reward function so that the target is localized and tracked by the team. All the UAS are supposed to provide additional information about the target, so their individual reward functions $\{R^1(s,a), R^2(s,a), \cdots, R^N(s,a)\}$ encode a high reward when the target is within their fields of view and no reward otherwise. For all the members of the team, a cost of 1 is associated with the actions.

Solving the above POMDP for the whole team is a highly non-scalable approach regarding the number of UAS. Hence, a different scheme is proposed where each UAS i solves its own POMDP without considering the other UAS. That POMDP has state $s^i = (p_l^i, p_h^i, t_l)$, and reward $R^i(s,a)$. Then, once the policies have been calculated, the coordination is achieved during the execution phase by sharing a fused belief state $b^{cen}(s)$ that considers information from all the robots. This belief can be computed in a decentralized manner with the method in Section 3, what makes the approach scalable for multiple UAS.

With this approach based on POMDPs, coordination arises implicitly due to the fused belief, and there is no need to solve the original POMDP, which was far more complex. Thus, the approach can scale with the number of UAS in the team, since this does not influence the complexity of the independent models to solve. The drawback is that the policies are sub-optimal, as the UAS do not reason about the others' actions, but through proper design of the rewards and the shared information that is provided by the decentralized data fusion, a coordinated behavior useful for many applications can be obtained.

5 EXPERIMENTS

In order to test the proposed methods for target tracking, several simulations are presented here. A simulated environment was created in Matlab and discretized into the 12x10 grid shown in Fig. 2. For these simulations, a team with four homogeneous UAS is considered. All of them carry bearing-only sensors with the same field of view (see Fig. 2) and detection probability ($p_D = 0.9$). The role of the



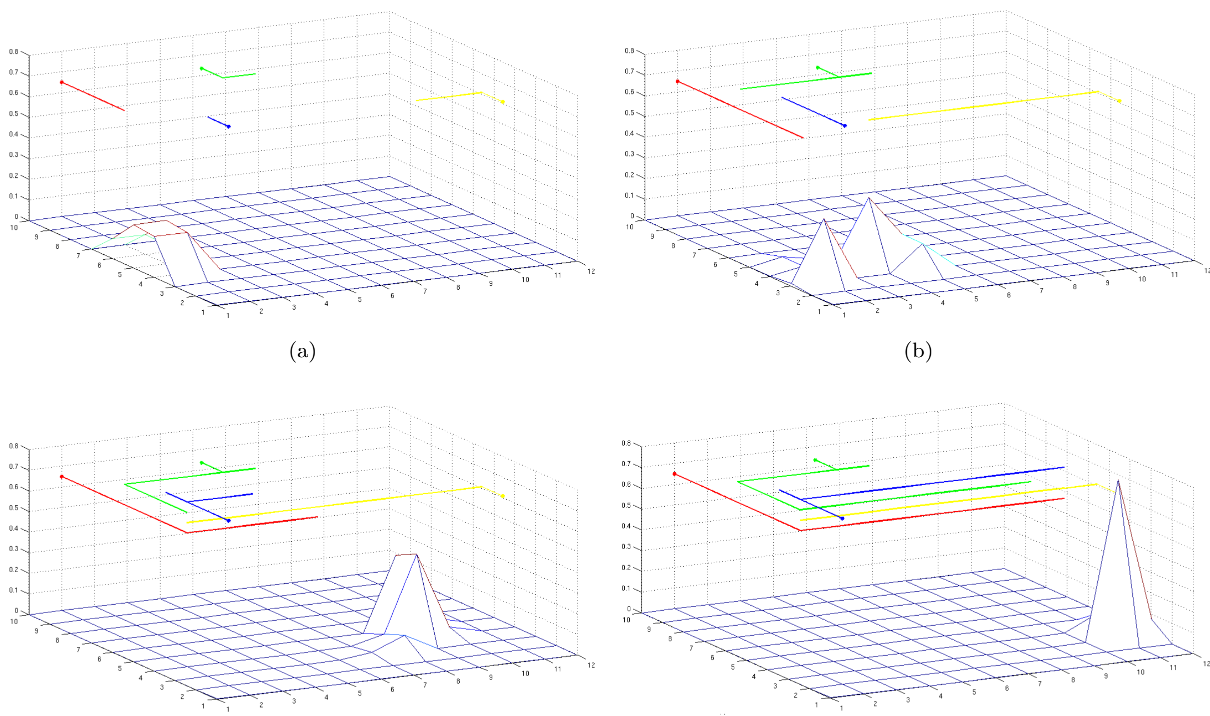
(a) (b) (c) (d)

Figure 3: Sequence of an experiment using totally independent UAS (steps 8, 16, 26 and 34). The surface represents the belief of the red UAS on the target position. The solid lines are the trajectories of the UAS, representing the dots their initial positions. The UAS fly at different heights to avoid collisions.

UAS is to get as close as possible to the target to reduce its uncertainty. Thus, the UAS get a high reward (+100) when the target is in one of the closest cells of their field of view, otherwise the reward is zero. Fig. 2 shows an example of the field of view of a UAS. Besides, the cells that would get high rewards are marked with crosses.

In these experiments, no uncertainty is considered for the UAS' control. Thus, every time a UAS takes one of the actions described in the previous section, it is supposed to be executed perfectly. Nevertheless, there is some uncertainty associated with the observations and the target model. Actually, the target is assumed to move randomly. Therefore, the transition function for its position t_l indicates that, from one time step to the next, the target can move to any of its 8-connected cells with the same probability (only non-obstacle cells are considered in order to calculate that probability).

Two main experiments are presented here. In the first one, each UAS calculates its policy independently and run it without considering the other. In the second experiment, the approach proposed in this paper is tested. Even though the policies are calculated separately, the UAS share a belief during the execution. In both experiments, the initial position of the UAS as well as the target trajectory were equal. Nonetheless, in order to include some uncertainty in its behavior, at every time step, the target could (with equal probability) either stay in the same cell or follow the prefixed path. Moreover, all the



(a) (b) (c) (d)

Figure 4: Sequence of an experiment using the coordinated approach (steps 5, 10, 15 and 22). The surface represents the fused belief for the red UAS.

policies were calculated with the algorithm SARSOP.

Results of the first experiment are shown in Fig. 3. Since each UAS looks for the target on its own, they encounter some difficulties due to the lack of range measurements. In fact, the belief of the UAS is quite flat during the whole experiment. Of course, when one of them detects the target the others do not notice it and keep searching around, so the UAS trajectories are not coordinated.

The results of the second experiment are summarized in Fig. 4. In this case, the localization of the target is improved and beliefs with high peaks are achieved. This is because all the UAS know about the target position once it is detected (fused belief), so they coordinate their trajectories toward the target to take more measurements.

Both approaches, considering communication between the robots (a shared belief) and no communication, were also compared running 500 simulations of 20 time steps each one. In all the runs, the four UAS started at the same positions as above and the target followed the same path (also with the randomness explained before). The average expected reward and the average entropy ($-\sum_i p(s_t = s_i)\log(p(s_t = s_i))$) were computed for all the simulations. The results are shown in Table 1. It can be seen how the implicit coordination achieved through decentralized data fusion leads to higher rewards and lower entropy, i.e. the proposed approach reduces the uncertainty in the target position.

	Coordinated	**Non-coordinated**
Reward	976.91 ± 13.81	540.14 ± 8.6
Entropy	1.85 ± 0.01	3.35 ± 0.01

Table 1: Average reward and entropy for 500 simulations with coordination and without it. Standard deviations are also shown.

6 CONCLUSIONS

This paper has presented preliminary simulations that show how POMDPs can be used for multi-UAS planning under uncertainty. Even though the intractability of some models is still a drawback for multi-UAS teams, the paper has shown how solving the POMDPs in a decentralized manner and sharing information can lead to implicit coordination within the team, without having to solve the full POMDP model. In this sense, results with non-coordinated and coordinated UAS were compared and a better performance was achieved with the latter.

Since each UAS does not reason about the others' actions, no intentional cooperation is considered. However, the proposed approach is rather scalable and the whole scheme (including the belief estimation) is totally decentralized.

Next steps include the implementation of the proposed method in a fleet of UAS; particularly, in the indoor CATEC's testbed with up to 10 quadrotors (see Fig. 1, right). Furthermore, future works will address techniques with more explicit cooperation between the UAS. And, although one particular application has been presented, due to their generality and adaptability the same techniques can be applied to other scenarios, like environmental monitoring, as well as to the cooperation between static sensors and UAS for cooperative navigation when the UAS lack of accurate sensors for localization.

REFERENCES

[1] F. Bourgault and H.F. Durrant-Whyte. Communication in general decentralized filters and the coordinated search strategy. In *Proc. of The 7th Int. Conf. on Information Fusion*, 2004.

[2] F. Bourgault, T. Furukawa, and H.F. Durrant-Whyte. Decentralized bayesian negotiation for cooperative search. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2681–2686, 2004.

[3] J. Capitan, L. Merino, F. Caballero, and A. Ollero. Delayed-State Information Filter for Cooperative Decentralized Tracking. In *Proceedings of the International Conference on Robotics and Automation*, 2009.

[4] Ben Grocholsky, Alexei Makarenko, Tobias Kaupp, and Hugh F. Durrant-Whyte. *Lecture notes in Computer Science*, volume 2634, chapter Scalable Control of Decentralised Sensor Platforms. Springer, 2003.

[5] Ruijie He, Abraham Bachrach, and Nicholas Roy. Efficient Planning under Uncertainty for a Target-Tracking Micro-Aerial Vehicle. In *IEEE International Conference on Robotics and Automation*, 2010.

[6] J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. SPUDD: Stochastic planning using decision diagrams. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

[7] H. Kurniawati, D. Hsu, and W.S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*, 2008.

[8] George Mathews, Hugh Durrant-Whyte, and Mikhail Prokopenko. *Advances in Applied Self-Organizing Systems*, chapter Decentralized Decision Making for Multiagent Systems. Springer-Verlag, 2008.

[9] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1025–1032, 2003.

[10] Pascal Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. PhD thesis, University of Toronto, 2005.

[11] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.

[12] T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 520–527, 2004.

[13] T. Smith and R. Simmons. Point-based POMDP algorithms: improved analysis and implementation. In *Proceedings of the 21th Conference on Uncertainty in Artificial Intelligence*, pages 542–547, 2005.

[14] Matthijs T. J. Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[15] A.W. Stroupe and T. Balch. Value-based action selection for observation with robot teams using probabilistic techniques. *Robotics and Autonomous Systems*, 50:85–97, February 2005.

[16] E.-M. Wong, F. Bourgault, and T. Furukawa. Multi-vehicle bayesian search for multiple lost targets. In *Proceedings of the International Conference on Robotics and Automation*, pages 3169–3174, 2005.

[17] Feng Zhao, Jaewon Shin, and J. Reich. Information-driven dynamic sensor collaboration. *Signal Processing Magazine, IEEE*, 19(2):61–72, March 2002.