2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Decentralized Cooperation of Multiple UAS for Multi-target Surveillance under Uncertainties

Jesus Capitan[1], Luis Merino[2] and Anibal Ollero[1]

Abstract— **An interesting applications of Unmanned Aerial Systems is surveillance. Surveillance typically involves the tracking of one or several targets in an area. A key issue for this application is the autonomous decision-making to allocate UAS to targets and determine the actions to be performed by the UAS of the fleet. Since this optimal decision-making needs to deal with uncertainties, Partially Observable Markov Decision Processes (POMDPs) are proposed as models for the surveillance mission. The paper proposes a role-assignment method to alleviate the computational complexity of the application of POMDPs to multi-UAS surveillance. The method is evaluated by simulation and compared with other similar approaches. Furthermore, the system has been implemented in a testbed with real quadcopters to show some preliminary experiments of two UAS tracking two different targets.**

I. INTRODUCTION

A relevant applications of Unmanned Aerial Systems (UAS) is the surveillance of multiple targets [1], [2], [3] (see Fig. 1). This functionality is required in search and rescue missions, monitoring scenarios, traffic control and many others. UAS are particularly useful for monitoring targets, since they have a wider field of view and can access more difficult places than other types of vehicles. In those cases, the cooperation of multiple UAS to monitor the different targets can make a big difference, since the objectives may be far away or too dynamic for a single vehicle.

The problem of target tracking has been extensively considered from the point of view of sensing, and many different filters can be used to estimate the position of the targets from the sensor outputs, and also to maintain an estimation on the associated uncertainties [4]. For the case of multiple vehicles cooperating in the tracking mission, cooperative filters have been also devised to fuse all the information present in the team [5], [6].

Besides estimation, the team of vehicles has to be controlled to track the targets. The objective in a multi-target surveillance operation with multiple UAS is to assign dynamically targets to vehicles in order keep them within the field of view of the team as long as possible. At the same time, many other aspects should be considered, like fuel consumption, communications, etc.

Several approaches have been presented for multi-target tracking with UAS, as for instance [7], [8], [9], [5], [10].

*This work has been supported by the FP7 EC-SAFEMOBIL Project (grant number 288082) and the PAIS-MultiRobot (TIC-7390) Regional Project.

[1]J. Capitan and A. Ollero are with the University of Seville, Seville, Spain [jcapitan,aollero]@us.es

[2]L. Merino is with the Pablo de Olavide University, Seville, Spain lmercab@upo.es

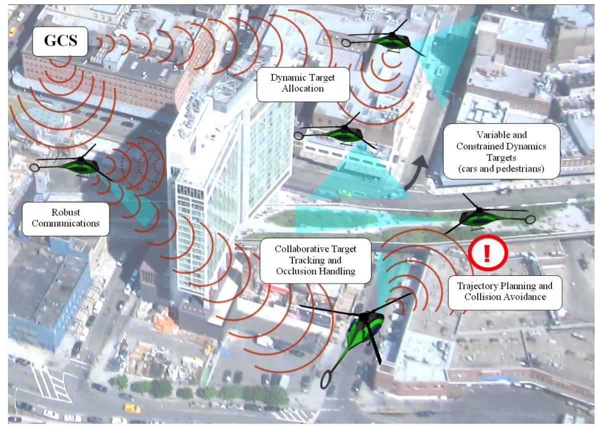

Fig. 1. A depiction of a cooperative surveillance application. In it, a set of cooperating UAS should track and monitor a set of (non-cooperative) targets, like cars, people, etc. Several issues should be handled, like reasoning on uncertainties due to occlusions, imperfect sensors and models on the targets' motion, etc; the dynamic allocation of UAS to the different targets; and the safe flying of the vehicles avoiding obstacles and other teammates (not considered in this paper).

The problem is often casted as an stochastic optimal control problem, in which an utility function related to the problem at hand has to be optimized [11].

In most surveillance applications, there are severe uncertainties involved: targets' positions are unknown, sensors and models are imperfect, occlusions can happen, and so forth. Therefore, the team of UAS needs to make decisions dealing with those uncertainties during the mission.

A very important objective of multi-target tracking is to maximize the information gained on the targets. Several indexes can be used to measure the amount of information, like entropy, mutual information, etc. Most of the works assume Gaussian uncertainties and Kalman (or Information) Filters as the underlying estimation framework, and define utility functions based on these information measures to determine the actuations [12], [13], [14]. However, tracking applications can result in multi-modal distributions when estimating the targets' positions. Hence, other works also consider different representations, like discrete Bayes filters [8], [9] or Particle Filters [5].

As commented above, besides maximizing the information available, the system should reason on the fuel available, motion models, control efforts, etc. All the uncertainties involved in the process, as well as the value of information itself, are considered in a principled way by *Partially Observable Markov Decision Processes* (POMDPs). This decision-making method [15] allows for optimal planning

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

by reasoning about uncertainties in terms of sensing and actuation. In particular, they define variables to model the environment states, the UAS' states, the actions and the observations, as well as their probabilistic transitions. Then, designing adequate reward functions, optimal policies can be found for each UAS to performed the assigned tasks.

POMDPs provide a sound mathematical formulation with a high level of flexibility. A broad variety of planning problems under uncertainties can be tackled by varying the models and designing the rewards properly. However, POMDPs have also problems in terms of scalability, since they need to search the state's, action's and observation's spaces when computing optimal policies. In particular, they do not scale well with multiple agents. In our surveillance application, multiple vehicles and targets would make the state's, action's and observation's spaces grow exponentially, what leads to intractable problems for large teams.

In our work, we propose several methods in order to alleviate the complexity of POMDPs for applications with multiple UAS. First, we assume mixed observability, which can reduce dramatically the complexity of some models [16]. In this case, some variables are considered observable, reducing the searching spaces. For instance, in many tracking missions, UAS can assume their positions as known, being the actual uncertainty associated with the targets' locations. This will work when the sensors for local positioning are accurate enough to dismiss their uncertainty against the targets' uncertainties.

Second, we use factored models. POMDPs have also been proposed with these models[17], where the state can be factorized into several factors or variables. The idea is to reduce the complexity of the joint models and exploit conditional independence between the variables. For instance, several factors may be common for different UAS, so we do not need to repeat operations on those factors for each UAS.

Third, we propose to combine different behaviours or roles to achieve the goals of the original problem. We compute optimal policies for single-UAV models that represent the roles. Then, we combine those roles optimally during the execution of the mission. The roles are not fixed, but they can be re-assigned at every time step. A policy for the multi-UAS model is not computed, so the method gains scalability. In addition, cooperation is achieved by the team, since the roles are assigned in an optimal fashion.

We extend our previous work [18] to propose a decentralized auction that assign single-UAS policies for multi-target surveillance. In our previous work, decentralized auctions of POMDP-based behaviours were already used for cooperative tracking. Here, we contribute by extending the work to multiple targets without increasing the computational complexity. For that, factored models are considered and a generalization of the behaviours proposed. Now, each behaviour is defined over a subset of the whole set of factors, which will allow us to re-use policies to emulate different behaviours.

Our approach is totally decentralized and scalable, which is an advantage for multi-UAS applications with hard communication constraints. The UAS employ only local data and local communications (communications with other UAS in the local neighbourhood). Furthermore, the UAS can still take actions when only local information is available. Nonetheless, since cooperation is pursued, UAS share information whenever they are within communication range. This information is used to assign the behaviours and to maintain a common belief of the environment state by means of a *decentralized data fusion* scheme.

The paper is organized as follows: Section II introduces POMDP models for single and multiple agents; Section III describes our decentralized auction for factored behaviours; Section IV details the factored models proposed for multi-target surveillance; Section V presents experimental results and Section VI gives the conclusions and future work.

II. POMDP MODELS

This section explains briefly the models for single and multi-agent POMDPs. It also introduces the idea of factored models with mixed observability.

A. Single Agent

Formally, a discrete POMDP is defined by the tuple $\langle S, A, Z, T, O, R, D, \gamma \rangle$ [15].

- The *state space* is the finite set of possible states $s \in S$, for instance the targets and UAS poses.
- the *action space* is defined as the finite set of possible actions that the UAS can take. $a \in A$;
- and the *observation space* consists of the finite set of possible observations $z \in Z$ from the onboard sensors.
- After performing an action a, the state transition is modeled by the conditional probability function $T(s', a, s) = p(s'|a, s)$, which indicates the probability of reaching state s' if action a is performed at state s
- the observations are modeled by the conditional probability function $O(z, a, s') = p(z|a, s')$, which gives the probability of getting observation z given that the state is s' and action a is performed.
- The reward obtained for performing action a at state s is $R(s, a)$.

The state is non-observable; at every time instant the agent has only access to observations z which give incomplete information about the state. Thus, a belief function b is maintained by using the Bayes rule. The new belief b' obtained if we apply the action a at belief b and get the observation z is given by:

$$b'(s') = \tau(b, a, z) = \eta O(z, a, s') \sum_{s \in S} T(s', a, s) b(s) \quad (1)$$

The normalization constant:

$$\eta = p(z|b, a) = \sum_{s' \in S} O(z, a, s') \sum_{s \in S} T(s', a, s) b(s) \quad (2)$$

gives the probability of obtaining a certain observation z after executing action a for a belief b.

The POMDP model assumes that, at every step, an action is taken, an observation is made and a reward $R(s, a)$ is given. The objective is to determine the actions that maximize the sum of expected rewards, or *value*, earned during D time steps. This actions will depend on the information available, represented by the belief state b. Thus, the objective is to determine the policy $a = \pi(b)$ that maximizes the cumulative reward in time, or value $V^\pi(b)$:

$$V^\pi(b) = R(b, \pi(b)) + \gamma \sum_{z \in Z} p(z|b, a) V^\pi(\tau(b, \pi(b), z)) \quad (3)$$

where $R(b, a) = \sum_s R(s, a)b(s)$ is the expected immediate reward. To ensure that the sum is finite when $D \to \infty$, rewards are weighted by a discount factor $\gamma \in [0, 1)$. The value of the optimal policy π^* is usually denoted by $V^*(b)$.

The same formulation could be casted using costs instead of rewards. Note that, once the system is correctly modeled through the transition and observation functions, the reward (or cost) function is critical, since it is the way the desired behavior is incorporated into the system.

B. Multiple Agents

When a set of n agents is considered, each agent i can execute an action a^i from a finite set A^i and receives an observation z^i from a finite set Z^i. The transition function $T(s', a^J, s)$ is now defined over the set of joint actions $a_J \in A^1 \times \cdots \times A^n$ (the actions that the fleet as a whole can perform), and the observation function $O(z^J, a^J, s')$ relates the state to the joint action and the joint observation $z^J \in Z^1 \times \cdots \times Z^n$. The common reward signal is defined over the joint set of states and actions $R : S \times A^1 \times \cdots \times A^n \to R$. The goal in the multi-agent case is to compute an optimal joint policy $\pi^* = \{\pi^1, \cdots, \pi^n\}$ that maximizes the expected discounted reward (as in the normal POMDP case)

There are two main options to solve these multi-agent POMDPs: the Decentralized POMDP (Dec-POMDP) model allows for fully decentralized execution [19], while the Multi-agent POMDP (MPOMDP) takes a centralized approach [20]. The key difference between the Dec-POMDP and MPOMDP models is that in the decentralized case, each agent only observes its local observation z^i, while in the centralized case, each agent observes the full observation vector z.

The main issue in the multi-agent case is that the computational complexity of the problem increases exponentially with the number of agents, and in particular the computational complexity of solving a Dec-POMDP is significantly higher than that of a POMDP (NEXP-complete [19] vs. PSPACE-complete [21]). This renders the direct application of these models, useless except for the simplest cases.

C. Factored Models

Factored POMDPs are also widespread [17]. The idea is to exploit conditional independence between variables in order to obtain a compact representation of the transition and observation models. Thus, in a factored POMDP, the state consists of a set of d variables or factors: $s = (s_1, s_2, \ldots, s_d)$ which present some conditional independence relations. The rewards can also be expressed as factored functions depending on a subset of variables.

Additionally, mixed observability can be considered into POMDPs [16]. Assuming some of the factors observable can alleviate the complexity of the model by reducing the state space of the belief. This assumption makes sense in many applications where the uncertainties associated with some variables are not relevant (for instance, for a set of UAS with GPS receivers it might be assumed that the position of the UAS is known with enough certainty).

III. DECENTRALIZED AUCTION FOR FACTORED POMDPS

In our previous work [18] we proposed to emulate multi-agent POMDPs with single-agent policies. The idea was to auction different individual roles to perform a mission in a cooperative way with a team of agents, avoiding the complexity of solving the full MPOMDP. The method was demonstrated in an application where some robots had to track a single target. This section explains the method and its extension to cope with multiple targets without increasing models' complexities.

A. Auctioning Policies

In order to approximate a multi-agent POMDP, a set of single-agent policies are pre-computed offline. Each policy is obtained by solving a single-agent POMDP whose reward function encodes a certain role or behavior. During the execution of the mission, the agents need to select at each moment which is the role that better suits them. Thus, the roles are distributed in an optimal fashion among the agents so that they can carry out their mission cooperatively. Moreover, these roles are dynamic and the agents can switch among them continuously during their mission depending on the circumstances.

The single-agent models that represent the different roles are designed in such a way that their joint execution emulates as good as possible the expected behaviour for the original multi-agent POMDP. This is done by a human designer and depends on the application. For example, when tracking a target with bearing-only sensors, roles to follow the target from different angles would help the team to reduce the uncertainty on the target position [18].

Once the roles are designed and their corresponding policies computed, we have a value function for each single-agent POMDP. This value function $V(b)$ provides the expected average reward of the policy for a given belief of the state. During the mission, each agent maintains a belief over the state, which is used to evaluate the value functions of the different roles. In order to foster optimality, roles with higher value functions are preferable for each agent at each moment. The assignment is solved by means of an auction algorithm where the bids are the negated values obtained by evaluating the policies of the different roles. Therefore, the roles are re-distributed at each time step in such a way that agents are assigned behaviours with higher expected rewards (as long as the local beliefs change, the role distribution could do it too).

B. *Decentralized Data Fusion*

Apart from the auction, a decentralized data fusion scheme is used to obtain a common belief among the agents. This also help the system to foster cooperation, since the agents have a fused belief with a consensus information from the team. The same situation would happen in a multi-agent POMDP, but here we used a decentralized filter for data fusion [6].

Basically, the joint belief is estimated using just local information and exchanging information with the neighbors. Each agent employs only its local sensor data and then *shares* its belief with its neighbors at certain time instants. The received information from other teammates is also locally fused in order to improve the local perception of the world.

C. *Factored Generalized Roles*

In a first approach, each role is represented by a different POMDP model with a reward function that encodes that behaviour. Therefore, as many policies as roles will be pre-computed offline and later evaluated online over the evolving beliefs. In this case, roles and policies match. However, the use of factored models can allow us to generalize the concept of roles and exploit redundancies in the models.

A role can be represented by a given policy over a certain subset of factors of the whole state. Thus, the same policy but evaluated for different subsets of factors, could produce different roles. This means to re-use the same policy but with different factors to emulate different behaviours. Now, we can have a set of pre-computed policies that are used to generate a different set of roles, depending on the part of the factored belief that is plugged into each policy. The number of policies to solve and roles is no longer the same, so advantages in terms of computational complexity can be obtained.

In the next section, it will be shown how the same POMDP model (with its policy associated) can produce different roles depending on the belief factors plugged. This is done by exploiting redundancies in the model and will allow us to tackle a multi-target surveillance mission by using only one-to-one POMDP models, in which a single UAS tracks a single target.

IV. POMDP MODELS FOR MULTI-UAS SURVEILLANCE

In this section we will present how the previous models can be applied to survey multiple targets with a team of multiple UAS. We recall that the objective is to allow a fleet of UAS to monitor a set of targets. The main objective is to be able to maintain an estimation of the position of all the targets. The system will work at a planning level, giving waypoints to the UAS that are later followed by the low-level navigation control algorithms.

Firstly, we present the multi-UAS case and later we describe the single-UAS model used to generate different roles. Moreover, it will be shown how these roles are combined with our method in order to carry out multi-target surveillance with the team of UAS.

A. *Multi-UAS POMDP*

We want to monitor a set of m targets with a team of n UAS. That problem can be modeled as a multi-UAS factored POMDP. The scenario is discretized into a cell grid and the state at each iteration is a vector with the following discrete factors: $s^{multi} = (t_1, \ldots, t_m, l_1, \ldots, l_n)$. The position in the grid of each target j is represented by factor t_j, while the position of UAS i is represented by factor l_i.

Each UAS is equipped with a camera sensor pointing downwards that provides a (noisy) binary observation about the presence of a target. Thus, the observation vector for each UAS i is $z^i = (o_1^i, \ldots, o_m^i)$, where o_j^i is a binary factor indicating whether UAS i has detected target j. At each iteration, each UAS can also take an action $a^i \in \{\textit{stay}, \textit{north}, \textit{west}, \textit{east}, \textit{south}\}$, in order to hover on the same cell or move to a neighbouring cell.

The UAS need to maintain a belief over the state $b(s)$, which can be updated by applying (1) with probabilistic transition and observation functions: $T(s', a^J, s)$ and $O(z^J, a^J, s')$. Noisy transition functions are considered for the UAS actions and the target movements; and a target can be detected with a certain probability p_D if it is in one of the 9-connected cells of a UAS. Additionally, mixed observability is used in order to reduce the complexity of the model. In this case, UAS' positions (l_i) are considered observable. Due to their on-board sensors, UAS can localize themselves with high precision in our application, so we can assume that this uncertainty is insignificant compared to the uncertainty on the targets' positions.

The mission could be modeled by designing a joint reward function over the whole state space, encoding the preferences of the system. In this application, a high reward is given for each UAS located in the same cell as a target. If the target is already being monitored by another UAS, the reward is lower. Therefore, we foster a cooperative surveillance, since the UAS will try to catch all the targets as many times as possible and they will distribute their sensing capabilities efficiently. Note that the complexity of this model increases exponentially with the number of UAS and targets, since the state's, observation's and action's spaces do.

B. *Single-UAS POMDP*

Instead of solving the previous multi-UAS model, which does not scale with the number of UAS and targets, we propose to find a policy for a simpler model and then emulate different behaviors with it. The model we solve consists of a POMDP for tracking a single target with a single UAS. The state is now $s^{local} = (t, c, l)$, where t and l are the positions of the only target and the UAS, respectively. Another binary factor c is included. This factor specifies whether the target has been visited or not. Thus, when the UAS is on the same cell as the target, this variable is set to 1. Otherwise, if the target was already visited, c can switch back to 0 (not-caught) with a probability p_c at each time step. This is used by the UAS to forget that the target was detected after some time, and go after it again. As it will be seen, this will allow a single UAS to switch between different targets in the multi-target case.

Observations and actions are the same as in the multi-UAS model, but for a single target and UAS. The reward model changes slightly in order to take into account the new factor c. If the UAS is at the same position as the target and $c = 0$, a *high* reward is earned. If the UAS is at the same position as the target but $c = 1$, a *low* reward is earned. Otherwise, no reward is earned.

C. Roles to Monitor Multiple Targets

A policy for the single-UAS model shown above is pre-computed offline. This policy is used to represent several factored behaviors at the same time. In this application, we have designed as many behaviors as targets. Each behavior represents tracking a specific target, so the UAS will try to distribute the behaviors among them by tracking different targets.

With this approach, cooperation between the UAS is fostered and the targets get distributed optimally among them. A UAS gets a reward for monitoring a target, but the reward is even higher if the target has not been seen for a while (not-caught). Thus, the UAS monitor all the targets in turns, and they will choose first targets that are not being followed by others, since their c factors will have lower probabilities.

The relevant thing here is that the m behaviors can be modeled using the same policy mentioned. The idea is to plug into the policy a factored belief containing only the factors corresponding to a certain target and UAS. For instance, if we use the belief over target j of a UAS i to evaluate the value function of the single-UAS policy, we will obtain the benefit of executing behavior j in UAS i. Therefore, in the auction, the bid value for each behavior j is computed by evaluating the value function of the single-UAS policy for the belief over target j. Thanks to the factorization of the models, we can exploit the probabilistic independence of the targets and tackle the problem without solving policies for multi-target models.

In order to evaluate the different behaviors and bid for them in a decentralized fashion, all the UAS need to maintain a multi-target belief with the following factors: $(t_1, \ldots, t_m, c_1, \ldots, c_m)$. This is achieved by executing the decentralized data fusion scheme explained in Section III. Then, during the execution of the mission, the decentralized auction is performed online among the UAS in order to switch continuously between different behaviors (between targets in this case). For instance, if UAS i wants to evaluate the behaviour j, it needs to plug into the value function of the single-UAS policy the following factored belief: (t_j, c_j, l_i). Once a behaviour has been selected, the optimal action is obtained in a similar manner, by plugging the reduced factored belief into the single-UAS policy.

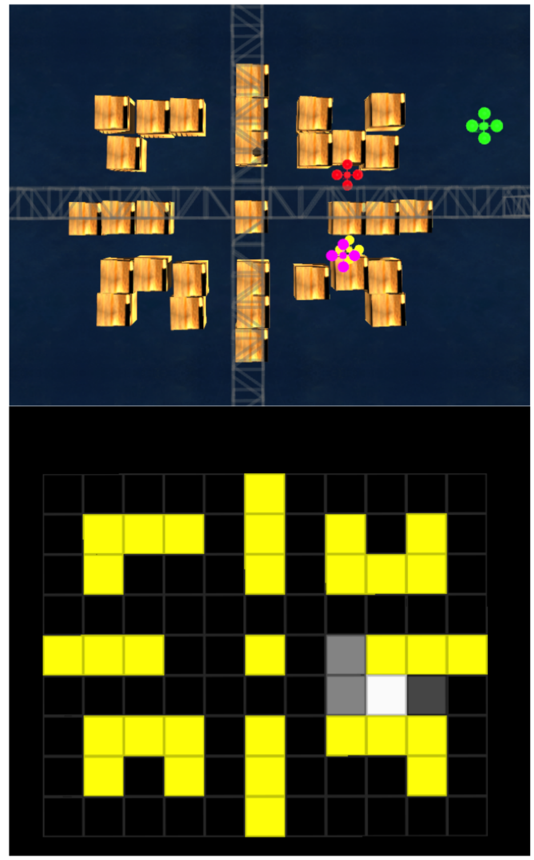

Fig. 2. Top: Simulated scenario. Two quadcopters act as targets (red and yellow), while two others act as the cooperative trackers (green and magenta). Bottom: the two trackers maintain a belief over the position of the two targets by using the decentralized data fusion filter. The figure shows the fused belief for one of the targets (the yellow one in this case). The scenario is discretized into a grid of cells (yellow tiles indicate non-flyable zones). The belief state is used to determine which tracker should be assigned to which target.

V. EXPERIMENTS

The techniques described in this paper have been validated by means of simulations and some preliminary experiments in the facilities of the Centre for Advanced Aerospace Technologies (CATEC) in Seville, Spain. CATEC has a testbed of 256 m^3, in which several quadcopters can fly at the same time (see Fig. 5) and track moving targets.

A. Simulations

In order to evaluate quantitatively the approach, a set of simulations has been performed. In the simulations, two UAS survey a zone where two targets are present. Although it is not a requirement of the technique, in this particular case the two targets were assumed not to be evading, and they followed a predefined trajectory, not known by the trackers. Figure 2 depicts the simulated scenario, which is adapted from [22].

Three different configurations have been tested to solve our auction:

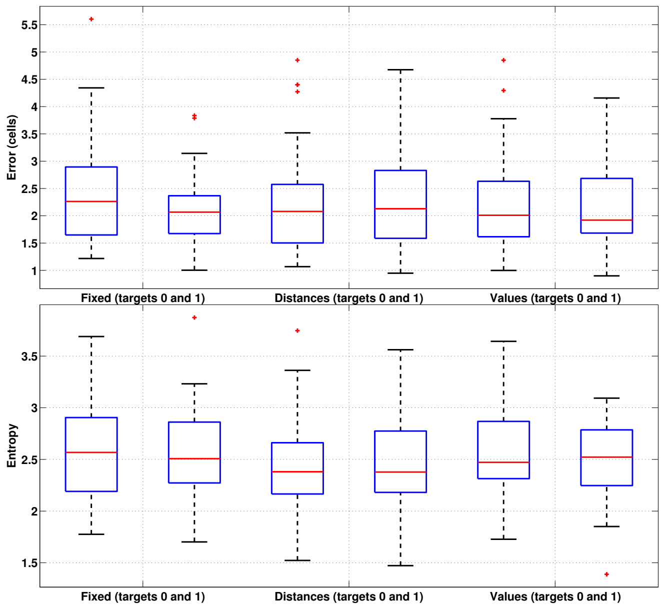

Fig. 3. Top: average error on the two targets' positions (expressed in cells) for all the simulations. Bottom: average entropy of the beliefs for both targets. The plot shows the median (red), the extreme values (whiskers) and the 25th and 75th percentiles (blue boxes)

- Fixed allocation: the UAS are assigned to the two targets in a fixed configuration (denoted as *fixed* in the figures).
- Distance-based allocation: each UAS is allocated, at each iteration, to the closest target. As the position of the targets is not known with certainty, the position with the highest probability in the belief state is selected as the estimated position of each target. This option is denoted as *distance* in the plots.
- Value-based allocation: this is the approach presented in this paper. At each iteration, the target selected to be tracked (behaviour) is the one that maximizes the value function (the expected cumulated reward) of the single-UAS POMDP. This option is denoted as *values* in the plots.

For each configuration, 50 simulations were performed, with random initial positions of the targets and UAS in the scenario[1]. In the simulations, $p_D = 0.9$ and $p_c = 0.04$.

Figure 3 shows the results on the average error position of the targets with respect to the ground truth position, as well as the average entropy of the beliefs. It can be seen how the allocation using the value function achieves the lowest mean error on the position of the targets, while the entropy is slightly lower in the case of the distance-based allocation.

Figure 4 shows the average accumulated reward for the different options. In this case, the value and distance-based allocation schemes again outperform a simple fixed allocation. Moreover, the distance-based allocation achieves the highest reward, although similar values are obtained by the value-based allocation. The value-based approach is more general, as it can be applied to any application using POMDPs. In this particular application, the distance-based approach makes sense and implies incorporating additional information on the problem at hand.

[1]The policy for the single-UAS was computed offline with Symbolic Perseus.

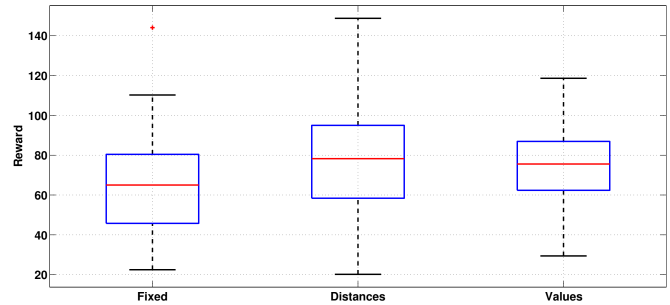

Fig. 4. Average accumulated reward for all the simulations and the different allocation schemes.

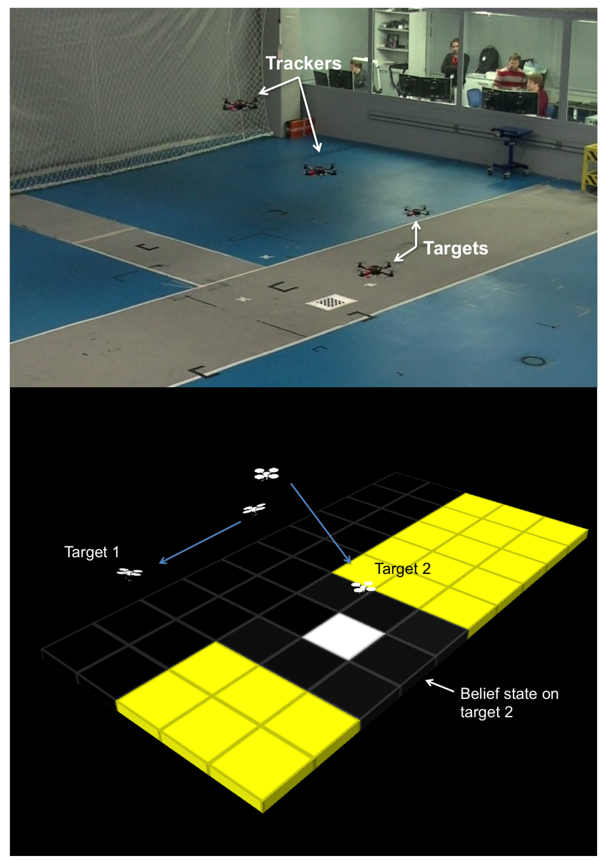

Fig. 5. Top: a picture of the real experiments carried out. Two quadcopters act as targets, while two others act as the cooperative trackers. Bottom: the figure shows the fused belief for one of the targets. The scenario is discretized into a grid of cells (yellow tiles indicate non-flyable zones).

B. Testbed Experiments

Some preliminary results were obtained in the real testbed of CATEC with an experiment similar to the simulations described above: two cooperative UAS tracking two targets. All of them were quadcopters, two acting as targets and two as trackers. The scenario is a bit simpler than the one used in the simulations due to space limitations (see Fig. 5).

The two tracker UAS were executing in real time our proposed technique. First, the decentralized data fusion filter was used to maintain a multi-target belief over the two targets. In this particular experiment, the UAS were within communication range all the time. Second, the POMDP policy was computed for the single-UAS model described in Section IV and used to define two different behaviours[2]. UAS selected a behaviour to monitor one of the targets, and then selected the next cell to move depending on the target (behaviour) assigned.

In the testbed, a VICON system provides the position of all the vehicles (targets and trackers) with millimetre accuracy. However, in the experiments only the positions of the trackers are assumed to be known. The positions of the targets provided by the VICON are only used to emulate the binary sensors described above. These preliminary results[3] show how the system is able to control the team of two UAS, fostering cooperation under uncertainties in a decentralized fashion.

VI. CONCLUSIONS

The paper has presented a method for multi-target surveillance with multiple UAS. The method is able to consider and reason about the uncertainties present in the application, both in the prediction models and sensors, by using POMDPs as the underlying model. In order to scale these systems to real-time applications for fleets of multiple UAS, the paper proposes a role-based auctioning method. The methods have been evaluated using simulations, both from a quantitative and qualitative points of view. Furthermore, preliminary experiments controlling a small fleet of UAS in a testbed are described as well.

The main contribution of the paper is showing that POMDPs are an adequate model for multi-UAS planning. Furthermore, the system described is scalable, and its computational complexity only depends on the size of the local communication neighborhood of each UAS.

As future work, instead of predefining them, the prediction and observation models could be learned from data from previous missions. Moreover, a key issue in this approaches is a proper definition of the reward (cost) functions. An interesting line of research is learning these cost functions from observing human experts determining the actions to be carried out by the fleet. Also, experiments with larger fleets will be performed to further illustrate the scalable nature of the approach.

REFERENCES

[1] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnstrom, and P. M. Olson, "Relay positioning for unmanned aerial vehicle surveillance," *International Journal of Robotics Research*, vol. 29, no. 8, pp. 1069–1087, 2010.

[2] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. G. V. K. andCamillo J. Taylor, Y. End, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie, "Adaptive teams of autonomous aerial and ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, pp. 991–1014, 2007.

[2]The parameters for the model and the policy are the same described in the simulations.

[3]A short video of the experiments can be found at http://personal.us.es/jcapitan/icuas-14.mov

[3] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, July 2006.

[4] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, 2001.

[5] L. Ong, B. Upcroft, T. Bailey, M. Ridley, S. Sukkarieh, and H. Durrant-Whyte, "A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles," in *IROS*, 2006, pp. 4539–4544.

[6] J. Capitan, L. Merino, F. Caballero, and A. Ollero, "Decentralized delayed-state information filter (DDSIF): A new approach for cooperative decentralized tracking," *Robotics and Autonomous Systems*, vol. 59, pp. 376–388, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/B6V16-526DWNX-1/2/dd823dbe262732b7d71fecdeb8cd1354

[7] R. He, A. Bachrach, and N. Roy, "Efficient Planning under Uncertainty for a Target-Tracking Micro-Aerial Vehicle," in *IEEE International Conference on Robotics and Automation, 2010*, 2010.

[8] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Decentralized bayesian negotiation for cooperative search," in *Proceedings of International Conference on Intelligent Robots and Systems*, 2004, pp. 2681–2686.

[9] E.-M. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 3169–3174.

[10] K. Cook, E. Bryan, H. Yu, H. Bai, K. Seppi, and R. Beard, "Intelligent cooperative control for urban tracking with unmanned air vehicles," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. IEEE, 2013, pp. 1–7.

[11] R. Anderson and D. Milutinovic, "Anticipating stochastic observation loss during optimal target tracking by a small aerial vehicle," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. IEEE, 2013, pp. 278–287.

[12] B. Grocholsky, A. Makarenko, T. Kaupp, and H. Durrant-Whyte, "Scalable control of decentralised sensor platforms," in *Information Processing in Sensor Networks*, ser. Lecture Notes in Computer Science, F. Zhao and L. Guibas, Eds. Springer Berlin / Heidelberg, 2003, vol. 2634, pp. 551–551.

[13] P. Skoglar, U. Orguner, and F. Gustafsson, "On information measures based on particle mixture for optimal bearings-only tracking," in *Aerospace conference, 2009 IEEE*. IEEE, 2009, pp. 1–14.

[14] F. Morbidi and G. L. Mariottini, "On active target tracking and cooperative localization for multiple aerial vehicles," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 2229–2234.

[15] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[16] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "POMDPs for Robotic Tasks with Mixed Observability," in *Proceedings of the Robotics: Science and Systems Conference*, Seattle, USA, 2009.

[17] P. Poupart, "Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes," Ph.D. dissertation, University of Toronto, 2005.

[18] J. Capitan, M. Spaan, L. Merino, and A. Ollero, "Decentralized multi-robot cooperation with auctioned POMDPs," *International Journal of Robotics Research*, vol. 32, no. 6, pp. 650–671, 2013.

[19] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, 2002.

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

[21] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of operations research*, vol. 12, no. 3, pp. 441–450, 1987.

[22] D. Hsu, W. Lee, and N. Rong, "A point-based POMDP planner for target tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 2644–2650.