Autonomous UCAV Coordination in Dynamic Search and Destroy Missions

Jan Tožička, Erika Benvegnu, David Šišlák, Michal Pěchouček, Niranjan Suri

***Abstract*—Military operations of the last decades are no longer conflicts of only two participants. Coalitions of countries are participating on one or both sides of the conflict. These conflicts are composed of a number of missions of different nature. We focus on the search and destroy missions which form an irreplaceable part of most of the conflicts since the Vietnam War. With the introduction of Unmanned Combat Air Vehicles (UCAVs) the importance of these missions increases even more since it allows to further decrease the number of causalities among the allies. In this article we discuss the use of UCAVs in search and destroy missions and compare two approaches: a multi-agent negotiation approach, and Process Integrated Mechanism (PIM). Both approaches allow a high degree of autonomy of the UCAVs, promising to decrease the operator load and the base–UCAV communication. We propose several different quality metrics and use them to evaluate and compare both approaches. We also propose an interesting strategy that uses both approaches to create a coalition of autonomous UCAVs, taking advantage of the strengths of a multi-agent approach and PIM.**

I. Introduction

Nowadays, unmanned aerial vehicles (UAVs) and unmanned combat air vehicles (UCAVs) are commonly used in warfare operations for (armed) reconnaissance and search and destroy missions [1], [2]. While reconnaissance missions are often performed completely autonomously, most of the search and destroy missions are driven by a human pilot remotely. The human pilot seems to be the bottleneck of the system when we try to assign several UCAVs to one mission, for example, the Predator, or the Shadow, require two human operators each [3]. Each operator, or a team of operators, is responsible for one UCAV and controls its actions. These pilots communicate with each other and coordinate their actions in order to achieve a common goal. By improving UCAV autonomy we can decrease the load on the remote pilot which can abstract from routine actions and focus on the most important goals. UCAV then communicate with the operator on demand in mission critical situations. A human is kept in the loop to launch missiles, to intelligently judge the situation [1] and also to take responsibility for the international laws of war [4]. The group of unmanned air vehicles could be composed of several UAVs and several UCAVs. Each type of vehicle focuses on different tasks of the mission, e.g. UAVs focus on surveillance and requests the intervention of a UCAV to perform an attack once a target is spotted. UAVs and UCAVs can also be members of different alliances, in this situation coalitions have to be created to fulfill the tasks. This can happen for example when one country can monitor a third party area but does not have permission to attack on that area – for target elimination a UCAV of another country having permission to attack has to join the coalition. In the rest of the article we will only discuss UCAVs even though many of the claims are also valid for UAVs.

In this article we explore and compare two different approaches for UCAVs cooperation to fulfill a search and destroy mission. A natural step toward higher UCAV autonomy is to replace each pilot with an AI entity – an agent. Such an agent can be placed on the UCAV to reduce base-UCAV communication and to avoid related problems. The first approach, a multi-agent negotiation based system, is a completely distributed peer-to-peer system with independent autonomous entities. All the agents negotiate their goals in order to fulfill a common mission in an effective way. The second approach is based on a Process Integrated Mechanism (PIM). Using PIM a single controlling unit manages all the entities, but the controller is not a specific entity, the process migrates to all the components in the system running locally in each one of them for a short period of time. Both approaches are compared in a realistic like simulation scenario. In this scenario a village with several moving insurgents is explored by a group of UCAVs. The goal is to locate and eliminate all the insurgents in a timely manner.

Section II describes the search and destroy mission in details and presents research focusing on it. Section III gives a brief overview of approaches to the multi-UCAV coordination problem. Two of these approaches, which we want to focus on, are then described in Sections IV and V. Section VI addresses the problem of coalition cooperation within both presented coordination approaches. The evaluation scenarion is presented in Section VII and the results are shown in Section VIII. Section IX provides the conclusion and shows the direction of our future research.

II. Search and Destroy Mission

Search and destroy missions form an integral part of military strategy since the Vietnam War [5]. The main idea is to search for the enemy in its own territory, destroy it and withdraw back to a safe territory.

As the name suggests the search and destroy mission itself is composed of two phases: the search and the destruction. Especially in cases when the location of the targets is not known a priori, or their position is changing, it is necessary to first *search* and identify the targets. This can be accomplished

J. Tožička, D. Šišlák and M. Pěchouček are with Agent Technology Center, FEE, Czech Technical University in Prague, Czech Republic (e-mail: jan.tozicka@agents.fel.cvut.cz).

E. Benvegnu and N. Suri are with Florida Institute for Human & Machine Cognition, Florida, USA.

in many different ways, e.g. visual identification, phone calls tracking, people monitoring, target marking etc. A phase of *destruction* typically follows the search. In the destruction phase targets are verified and a decision whether to eliminate the target or not is taken. It is necessary to observe the target all the time during this phase to be sure that the conditions didn't change, e.g. insurgent leaving targeted car, civilians entering targeted building, etc. Once a decision to *destroy* the target is taken, a missile from the UCAV is launched. It takes some time for the missile to reach the target and thus it is necessary to continue to track the target until it is destroyed and sometimes even longer to verify that the target has been eliminated. The task of keeping the target in the UCAV field of view is known as *tracking*. The destroy command is just a single event which typically does not interrupt the target tracking.

This basic mission scheme can differ in reality in many ways. In the simplest case targets are static (e.g. buildings). Then the UCAVs can split the mission area and fly over it using simple zig-zag trajectories [6]. In this basic situation once the UCAV has flown over an area and data from sensors have been processed we can assess if a target was discovered or not and this assessment will not change in the future. When the targets are moving or new targets can appear anytime and anywhere, persistent area surveillance has to be implemented [7], [8]. When tracking moving targets is also required complex algorithms need to be developed. A particularly complicated situation is when the number of tracking UCAVs is less than the number of targets on the ground. When this condition presents persistent surveillance of each target is impossible and it is necessary to switch between tracking multiple targets which can lead to losing the location of one of the targets. Possible solutions to this problem typically include plan recognition or at least some information uncertainty handling [8]. Another problem appears when the targets are moving faster than the tracking UCAVs. The UCAVs then need to cooperate to keep the target in their sensor ranges. Similar situation appears also in the completely opposite case when the target is much slower than the tracking UCAV and the UCAV has wide turning radius, than it's also impossible, for single UCAV, to have the target under the sensor persistently.

During operations in enemy's territory it is important to keep the units as invisible as possible. The UCAV to base communication can help the enemy to locate and to take aim at the UCAV. For this and other more technical reasons UCAVs often operate with limited communication capabilities, e.g. limited radio range. There are two approaches to handle this situation and allow the UCAVs to share their knowledge. We can either limit the movement of the UCAVs to maintain the communication graph connected [9] or plan for the UCAVs to meet at predefined randez-vous points in the future. Both approaches are subject to downsides. The first approach is suitable only for scenarios where there are enough free UCAVs and is subject to the unexpected destruction of an UCAV holding the graph connected. The shortcoming of the second approach is that an UCAV could be considered lost just because it missed one meeting while focusing on some higher priority goal. In the scenario described in Section VII we use the second approach, though the details go beyond the scope of this paper.

Missions are usually made of a dynamic set of UCAVs – new UCAVs can be deployed and join the mission anytime or UCAVs can be lost in action, destroyed, or even be dynamically reassigned to a different mission. Missions can last *forever* while the UCAVs temporarily leave the search and destroy task to return to the base to refuel and recharge and later rejoin the mission [10], [11]. Approaches to autonomous UCAV coordination need to take into account that the set of available agents can change. Additionally, to increase the performance, both when a new UCAV joins the mission or when a UCAV leaves the mission the tasks need to be redistributed among the active UCAVs.

A. Search

The problem of multi-UCAV search is how to deploy a team of cooperating UCAVs in an area to find all the targets on the ground [12]. Some aspects of this problem can influence the difficulty, namely whether the targets are moving or even intentionally avoiding detection and whether they are threatening.

Approaches usually partition the search area into feasible sub-areas that are then searched by single UCAVs. One such approach is described in [13]. The authors divide the problem in three sub-problems: 1. determining relative capabilities of UCAVs, 2. assigning areas to UCAVs, 3. search of assigned areas. All of these sub-problems are solved using low complexity algorithms and the experimental results[1] advocate the feasibility of this approach and its ability to operate in real-time.

One of approaches that does not explicitly divide the searched area into sub-areas is presented in [14]. Here the authors provide a dynamic programming approach to solve a search problem in which a formation of side-by-side flying UCAVs searches an area with regions of opportunity and hazard while the neighboring UCAVs have to stay close enough to be able to communicate and far enough to prevent collisions.

One important extension of the search problem is the search for evading and possibly threatening targets. In [12] the authors present a strategy to construct effective configurations of swarms of UCAVs to search a hostile environment. The strategy is robust with respect to possible loss of UCAVs during the mission. The authors show the dependency of the time needed to fully search the area on the number of UCAVs and present an algorithm for selecting a minimum number of UCAVs to deploy in order to meet a targeted search time within probabilistic guarantees.

Another realistic extension supposes that new targets can appear anywhere and at anytime. In this case we cannot mark any part of the map as *clean* unless it currently is in the field of view of an UCAV.

[1] Implementation done within the COMETS EU project – `http://www.comets-uavs.org/`

B. Tracking

The multi-UCAV tracking problem consists in controlling the UCAVs in such a way that the tracked targets are always in the field of view of the sensors. The problem has more variants depending on the type and number of targets, types and numbers of UCAVs and the environment in which the tracking is performed. In this contribution we focus solely on tracking of ground targets.

A simple and effective strategy for one UCAV to track one moving ground target in an open area is presented in [15]. The circular navigation algorithm presented in this work is a simple reactive policy that always navigates the UCAV on a hypothetical circle around the tracked target. Such policy is easy to implement and the authors show that it is feasible for ground targets of arbitrary motion model moving at varying speed.

An extension of the tracking problem is presented in [16]. The authors present an approach to solving the tracking task in an urban environment where the possibility of losing the target from sensors is augmented by the presence of tall buildings. The approach builds on the circular navigation algorithm, but to ensure the permanent visibility of the target multiple UCAVs are utilized. The contribution of the authors is an algorithm for determining the optimal center of the hypothetical circle and spacing between the UCAVs on the circle. The task is posed as an optimization problem in a discretized space and the algorithm is evaluated using Monte-Carlo simulation.

C. Destroying

When the UCAV is in destroy mode it will launch a missile upon a command from the human operator (in our simulation: after a certain period of time) – the tracking continues until the missile reaches its target to confirm a successful hit.

In the current stage we do not allow missiles to miss their target, so this tasks is always successful and performed within a predefined time period.

III. Multi-UCAV Coordination Techniques

Many researches have already focused on the multi-UCAV coordination, especially on the surveillance and tracking tasks (for an overview see for example [17]). Nevertheless most of the approaches rely upon a central decision maker and therefore cannot be considered to be fully distributed. For example auction-based algorithms [18] or optimization techniques (POMDP [19], mixed integer linear program [20], [21], etc.) have been successfully used for UCAV coordination. Although these solvers produce efficient results, they have no concept of privacy and allow one central authority to create plans for each UCAV. This can be a very restrictive assumption when the communication is limited or the UCAVs forming a coalition belong to different authorities and cooperate only to achieve a given goal. Another problem is scalability, many UCAVs could join the mission and these algorithms typically do not scale well.

Distributed approaches are either based on a reactive model – the UCAVs form a swarm [22] – or are based on distributed algorithms, e.g. DCOP [23]. The multi-agent based approach evaluated in this paper, Section IV, belongs to the last group. The PIM (Section V) approach instead combines the benefits of a central algorithm with a distributed execution. Both the approaches examined scale well, but only the multi-agent approach is able to protect the privacy of agents' knowledge.

IV. Decentralized Approach – Multi-agent

In this section we describe our multi-agent approach where each UCAV is represented by an agent responsible for its actions and trying to fulfill a common goal. Each agent is switching between the search and destroy tasks. Agents working on the search tasks evenly split the mission area and start to search using zig-zag strategies. Once a target is found, an agent will switch to destroy mode and start to track the target. When an agent switches to destroy mode the area it was previously patrolling remains uncovered, therefore the agent has to inform other agents about its new state. The other agents redistribute the whole area among themselves and continue the search. The agent in destroy mode continues by tracking the target until it receives a command to attack, then it shoots, and continues to track the target until it's destroyed. Agents in destroy mode do not cooperate (in our scenario only 1 UCAV is necessary for the target destruction), they just negotiate what agent will be allocated to which target, this ensures that two UCAVs will not waste their time attacking the same target. After the successful target elimination the UCAV switches back to search mode and informs the other agents. The area is redistributed again among the searching agents. Redistribution happens every time the number of searching UCAVs has changed. It can happen also as a consequence of the UCAV destruction or new UCAV joining the mission.

More technically, we can sketch the agent's algorithm as follows (steps when the agents communicate with each other are marked `[C]`):

Algorithm 1:

```
1. Split the area among all available
       agents [C]
2. Search assigned area and monitor
       incomming messages
3. If a not-tracked target T is spotted:
 3a. Start to track the target
 3b. Inform other agents:
       I'm busy tracking target T [C]
 3c. Once a command to destroy the
       target is received, destroy it
 3d. Inform other agents:
       I'm available [C]
4. If a message about busyness of other
       agents has ben received:
           Continue with step 1.
 4a. Otherwise: Continue with step 2.
```

One of the scenario extensions described in the introduction covers a real world case when the radio-based communication is limited and thus the agents cannot communicate with each other all the time. The agents are not informed about the busyness of other agents nor about the situation when some

UCAV is destroyed or a new one joins the mission. In this case UCAVs need to meet from time to time to redistribute the area among all the active UCAVs. We call this task *Randez-vous*. During the randez-vous task, while they can still communicate with each other, the agents need to agree on the next randez-vous location. In our case the agents can switch to the randez-vous task in two situations. First, after several search loops above the whole assigned area. Since the area is split evenly among the agents they will arrive at roughly the same time unless they are busy with tracking. The other case when an agent can switch to a randez-vous task is after shooting down an enemy.

The figure 1 illustrates possible transitions between the tasks in the case of limited communication.

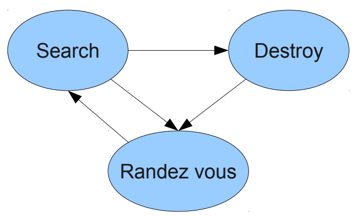


Fig. 1. Agents tasks and their transitions.

V. PIM a Semi-Centralized Approach

PIM is a framework for controlling and coordinating the behavior of distributed systems [24]. What makes PIM different from both a centralized approach and a multi-agent system is that it has a single Coordinating Process (CP), but this process is not located on any single node, it migrates, from one component of the PIM to another. However, the execution of the CP does not restart from the beginning after a migration. Each PIM component stores the code of the CP, the migration only transfers the current content of the memory and the Program Counter. This data enables the receiving entity to restart the execution of the CP where it was interrupted, allowing the programmer to develop algorithms as if the system were running on a single device.

One of the main components of PIM is the Runtime. A Runtime is running on every node of the PIM network and among other tasks it takes care of the migration and network management: the Runtime implements strategies for network discovery and to recognize if components have left the group. It also implements a fault tolerance mechanism to discover if the CP has incurred any errors or has been lost (for example, if a node that was running the CP was destroyed or disconnected) and undertakes appropriate measures to recover from the error. Discovery, detection of dead components, and fault tolerance together represent the core of the framework robustness. Moreover the network created by the PIM components does not have to be fully connected. It can in fact be very sparse. However since all the messages in PIM are sent using UDP (local) broadcast packets, redundant communication paths allow for less hopping when the CP requests access to data on a specific node.

For the purpose of the comparison described in this paper PIM was extended to work with the AgentFly system (Section VII) by adding a new type of PIM part that interacts with the AgentFly server exchanging messages to send new plans for the UAVs and receive information about the surveilled area. The CP holds a unified world view and controls the behavior of the UCAVs. The algorithm loops through the UCAVs extracting the information the UCAV has collected, parsing this information and appropriately instructing the UCAV with the next plan. UCAVs could be instructed to continue patrolling the zone they are assigned to, to flight to a target and start the tracking and shooting phase or they could be reallocated to a new area to patrol due to some other UCAV starting the tracking phase or returning to the search phase (leaving its area uncovered, or needing an area to patrol). The UCAVs in PIM are semi-autonomous. While they depend on the CP to instruct them, since the CP can plan with a centralized world-view, once the CP has created a flight plan they are able to fly and collect data from their sensors without the CP, also they can carry out the tracking and shooting with no interaction with the CP.

VI. Coalition cooperation

When several coalition partners each using autonomous UCAVs want to cooperate to carry out a mission, we present a solution that takes advantage of the good aspects of both the approaches described above. The multi-agent approach naturally allows agents forming a coalition to cooperate on a search and destroy mission. The agents share only the knowledge they need to fulfill their tasks while preserving the privacy of the knowledge they do not want to share with other coalition members. On the other side for the PIM process to migrate among the nodes, all the nodes have to run the same algorithm. All the knowledge migrates from one node to the other moving along with the CP. The PIM does not allow for part of the nodes to maintain their knowledge private, therefore it is not suited to work on nodes belonging to different parties of a coalition. We suggest that the UCAVs belonging to the same party of the coalition will run PIM. A different instance of PIM runs on each party, this way the knowledge and technical know-how are only shared among PIM nodes belonging to the same party. Each party also includes an agent of a multi-agent system which represents the party and interacts with agents belonging to the other members of the coalition. These agents can then coordinate their actions, and the actions of the UCAVs they represent. This solution gives us several benefits. The most important one seems to be the simplification of the communication complexity – not all the UCAVs communicate with each other but only agents representing several UCAVs participate in the negotiation. Since the complexity of many coordination algorithms grows steadily (often even exponentially) with the number of participants, this reduction of the number of agents participating in the negotiation can have a significant impact on the performance of the system.

VII. Comparison Scenario

To experimentally evaluate both approaches we implemented a search and destroy scenario within the existing AGENTFLY system [25]. AGENTFLY is a multi-agent based simulation system optimized for both manned and unmanned aeronautical traffic 2. All aerial assets in AGENTFLY are modeled as autonomous entities, each hosting multiple intelligent software agents and fully responsible for its own flight operation. The operation of each UCAV is specified by a plan created to follow the UCAV goals and the negotiation with other agents. The UCAVs continuously build a common operational picture (COP) in a distributed manner and to plan a set of actions (and create appropriate flight plans) upon this COP.

The architecture of the AGENTFLY system is designed to be flexible and open. It allowed us to simply connect the PIM architecture to drive the UCAVs on the planning and coordination level. The architecture is sketched in Figure 3, where you can see different types of Pilot agents (responsible for the coordination and the UCAV control) and also two communication networks. A *Simulation network* is used for the communication with the simulation server. The pilot agents send their commands and plans to the server and receive the sensor inputs through this network. A *Communication network* simulates the UCAV-to-UCAV radio communication channel. This network can simulate unreliable and range- or bandwidth-limited communication environment.

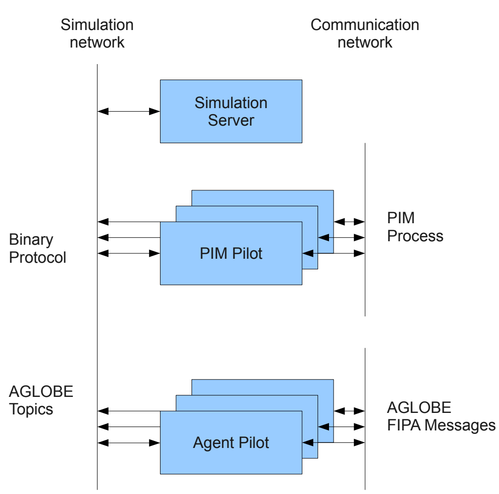


Fig. 3. System architecture. AGENTFLY simulation allows to connect different kinds of pilot agents. Agent-based pilots are naturally connected using AGLOBE system topics. For external PIM pilots, these topics are translated into a platform independent protocol. Similar protocol is used also in other direction, where PIM process controls the UCAVs.

We implemented the following *static* scenario which we then extended to evaluate the behavior of the multi-agent system approach and the PIM approach in a more dynamic environment. The static scenario has the following properties:

1) There are 4 enemies randomly placed in the mission area.
2) 4 UCAVs are instantiated and start to patrol the area.
3) When a UCAV detects an enemy it starts tracking it.
4) When the target has been in the field of view of the UCAV for 30 seconds the UCAV shoots a missile.
5) The missile flies for 5 seconds before it reaches the target.

The static scenario has been extended into following *dynamic* scenarios:

4 UCAVs: The scenario allows for new targets to appear on the ground. A new target appears on the map every minute, 6 new targets are created during the simulation.

2+2 UCAVs: The number of UCAVs collaborating to eliminate the targets changes during the mission. The mission starts with 2 UCAVs. After 4 minutes 2 more UCAVs join the mission. Also, new targets appear as in the case 4 UCAVs scenario.

The demos last until all the targets are eliminated.

A. Metrics

The scenario used for evaluation is configured to have a fixed number of targets and to run until all the targets are eliminated. We compare the agent based system and PIM using the following metrics: duration of the mission (time to eliminate all the targets), average target lifetime and bandwidth required to carry out the mission.

The search and destroy mission $M = (A, T, UCAV)$ assigned to a group of unmanned vehicles $UCAV$ is to eliminate all the targets T in an area A. We assume that $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of targets t_i. In a real world situation, typically T is not known in advance (e.g. war against terrorism) and the number of targets could dependent on the past actions of UCAVs, e.g., the destruction of a training camp can decrease the rate at which new targets appear or existing targets leave the targeted area A without their elimination. In our evaluation, we omit both these situations for ease of comparison.

a) Mission time: is the time needed to successfully accomplish the mission i.e. to eliminate all the enemies. We define the mission time $MT(T)$ as the time when the last target was eliminated

$$MT(T) = \max_{t_i \in T} d(t_i),$$

where $d(t_i)$ is the time when the target t_i has been eliminated. In our tests the time starts from zero at the beginning of the scenario.

b) Target lifetime: is the average time needed to eliminate each target since its creation, thus

$$LT(T) = \frac{\sum\limits_{t_i \in T} d(t_i) - c(t_i)}{|T|},$$

where $c(t_i)$ is the time when the target t appears in the scenario and $|T|$ is the number of targets destroyed during the simulation.

c) Time to discover target: is the average time needed to spot a target and is defined as

$$DT(T) = \frac{\sum\limits_{t_i \in T} s(t_i) - c(t_i)}{|T|},$$

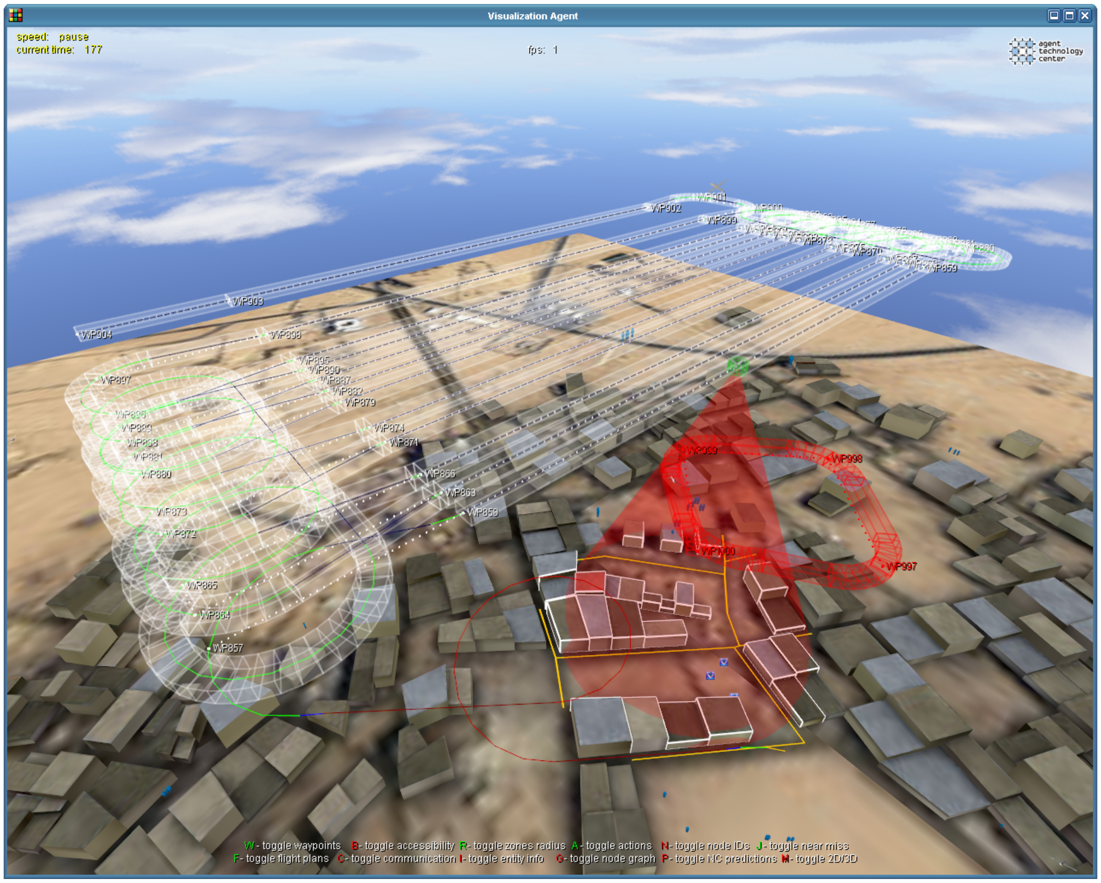


Fig. 2. UCAVs operating over an urban area. The UCAV in green safety area flies by its plan (white cubes with black and green lines) and search ground targets (blue people) within its field of view (red cone). The second UCAV tracks one target – its tracking pattern is illustrated by red cubes.

where the $s(t_i)$ represents the time when t_i was spotted the first time. If several targets are present in the scenario, a target can be spotted while UCAV is tracking other target and thus difference between $DT(T)$ and $LT(T)$ can be greater than the time necessary to destroy target after its identification (in our case we set it to 30 seconds). In such case, newly spotted target can move out of the field of UCAV view and it has to be found again later.

d) Situational awareness: represents how up to date the information the UCAVs have about the whole area A is, regardless of the locations of the targets. More precisely, it describes how often each point in the area A is observed on average. Situational awareness $SA(A)$ is defined as

$$SA(A) = \frac{\int_0^{MT(T)} SA(A,\tau)d\tau}{MT(T)}, \texttt{where}$$

$$SA(A,\tau) = \frac{\int_A a(x,\tau)dx}{|A|}$$

and $a(x,\tau)$ is the function representing the age of the UCAVs' information about point x of area A at time τ. $a(x,\tau) = 0$ if x is in the field of view of a UCAV at time τ. For our evaluation we assume the age of the information is 0 in the entire area when the simulation starts, $a(x,0) = 0, \forall x \in A$.

e) Tracking reliability: describes how well UCAVs can track the targets. It is an average of fractions of the time when targets $t_i \in T$ are within the sensor field of view. Tracking reliability $TR(T)$ is defined as

$$TR(T) = \frac{\sum_{t_i \in T} \int_{s(t_i)}^{d(t_i)} v(t_i,\tau)d\tau}{|T|},$$

where the target visibility $v(t_i,\tau)$ is defined as the function $v(t_i,\tau) : T \to \{0,1\}$, $v(t_i,\tau) = 1$ if and only if t_i is visible by any UCAV at time τ. In this paper, we do not focus on target tracking. We use a simple circular tracking algorithm and set UCAVs speed and turning radius such that they can keep the target in their field of view all the time while they are circling. Tracking reliability helps us to verify that targets are not lost from the sensor range after tracking has started.

f) Communication bandwidth: among UCAVs is used for comparison of the communication requirements of both approaches. The communication bandwidth $CB(M)$ is defined as the average communication flow in kilobytes during the whole mission.

VIII. Experimental Results

In this section we compare the multi-agent approach and the PIM approach presented in the two dynamic scenarios described above. In the first scenario four UCAVs are started at the same time. In the second scenario, two UCAVs are started when the scenario starts and two additional UCAVs join the mission after four minutes. The graphs on the left present the results of the first scenario while the graph on the right show the results of the second scenario. We used 5 different random configuration files defining the location of the targets. Each test was run with all the configuration files both for the multi-agent system and PIM.

The Figure 4 shows the time to accomplish to whole mission. Both approaches performed almost at the same quality level. In average, multi-agent was 10% faster in the 4 UCAVs scenario but 4% slower in the 2+2 UCAVs scenario. Nevertheless when we remove the extreme values from the measurements, the average mission time of PIM agents is 4% faster than the multi-agent approach in the 4 UCAVs scenario and both approaches perform equally in the 2+2 UCAVs scenario. The reason for that is that the PIM agents are more stable in their performance and their results are more consistent.

Communication bandwidth needed for the coordination of the UCAVs is shown on the Figure 5. Since the agents in the multi-agent based solution communicate only when it's necessary and they also limit the knowledge being communicated, they needed approximately 25% less communication bandwidth in the 4 UCAVs scenario and only half of the bandwidth in the 2+2 UCAVs scenario. The bandwidth used to complete the mission using PIM did not change significantly with a reduced number of PIM nodes. Therefore when a large number of PIM nodes join the mission, we do not expect a significant increase in bandwidth utilization. That is caused by the nature of the communication of the PIM nodes – the controlling process knowledge is being communicated all the time which is more or less independent on the number of nodes.

The target lifetimes are shown on the Figures 6 and 7. Targets are sorted by their lifetimes and then they are averaged over the 5 scenario runs with random targets positions. In average both solutions performed almost equally. We can see that the PIM approach exhibits slightly more stable results than the multi-agent based approach. Both solutions achieved similar results in both 4 UCAVs and 2+2 UCAVs scenarios.

IX. Conclusion

We presented two approaches to autonomous UCAVs coordination specifically targeted to search and destroy missions. We discussed an approach to coalition that takes advantage of the both the approaches: the distributed nature of the multi-agent negotiation and the centralized nature of the PIM process algorithm. We analyzed possibilities how the different coordination techniques can be used for search and destroy missions focusing on different variants of their subtasks. And finally we evaluated and compared the performance of both approaches in different scenarios to corroborate our thesis for a good strategy for autonomous UCAVs coordination in a coalition environment. Future work will include an analysis of the performances of the approaches when more UCAVs are present, as well as longer scenarios that require refueling, i.e., UCAVs temporarily leaving the mission and then returning to the search and destroy effort. We will also implement the proposed coalition strategy and evaluate the performance.

X. Acknowledgment

The AgentFly system has been sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-06-1-3073 and the Czech Ministry of Defense grant OVCVUT2010001. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, or the Czech Government.

References

[1] J. A. Angelo, *Robotics : a reference guide to the new technology*. Westport, Conn., 2007.

[2] U.S. Department of Defense, *Unmanned Aircraft Systems Roadmap 2010–2032*. U.S. Department of Defense, 2010.

[3] M. L. Cummings, S. Bruni, S. Mercier, and P. J. Mitchell, "Automation architecture for single operator – multiple uav command and control," *The International C2 Journal*, pp. 1–24, 2007.

[4] A. J. Lazarski, "Legal implications of the uninhabited combat aerial vehicle." [Online]. Available: http://www.airpower.maxwell.af.mil/airchronicles/cc/lazarski.html

[5] J. H. Hay, *Vietnam Studies, Tactical and Materiel Innovations*. DOA, Washington, D.C., 1989, ch. Chapter XV: Search and Destroy, pp. 169–178.

[6] I. Maza and A. Ollero, *Distributed Autonomous Robotic Systems 6*, ser. Distributed Autonomous Robotic Systems. Springer Verlag, 2007, vol. 6, ch. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, pp. 221–230.

[7] B. Bethke, J. P. How, and J. Vian, "Multi-UAV Persistent Surveillance With Communication Constraints and Health Management," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2009, (AIAA-2009-5654).

[8] "Commander's handbook for persistent surveillance," Joint Warfighting Center, Suffolk, Virginia. [Online]. Available: http://www.dtic.mil/doctrine/doctrine/jwfc/surveillance_hbk.pdf

[9] R. W. Beard, "Multiple uav cooperative search under collision avoidance and limited range communication constraints," in *In IEEE CDC*, 2003, pp. 25–30.

[10] C. L. Evans, "Multi-mission maritime aircraft acquisition planning: Requirements development and maturation," in *In Jons Hopkins Apl Technical Digest*, vol. 24, no. 3, 2003, pp. 25–30.

[11] P. River, "Multi-mission maritime aircraft acquisition strategy," Maritime Surveillance Aircraft Program Office, MD, 2001.

[12] P. Vincent and I. Rubin, "A framework and analysis for cooperative search using UAV swarms," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM New York, NY, USA, 2004, pp. 79–86.

[13] I. Maza and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *7th International Symposium on Distributed Autonomous Robotic Systems*. Springer, 2004.

[14] R. Beard and T. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *42nd IEEE Conference on Decision and Control, 2003. Proceedings*, vol. 1, 2003.

[15] F. Rafi, S. Khan, K. Shafiq, and M. Shah, "Autonomous target following by unmanned aerial vehicles," in *SPIE Defence and Security Symposium*, 2006.

[16] J. Kim and Y. Kim, "Moving ground target tracking in dense obstacle areas using UAVs."

[17] R. A. Wise and R. T. Rysdyk, "UAV coordination for autonomous target tracking," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, M. Likhachev, B. Marthi, C. McGann, and D. E. Smith, Eds., Keystone, Colo, USA, August 2006, pp. 3210–3231.

[18] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-uav context," in *ICRA*, 2004, pp. 3622–3627.

[19] S. A. Miller, Z. A. Harris, and E. K. P. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 2:1–2:17, January 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/724597

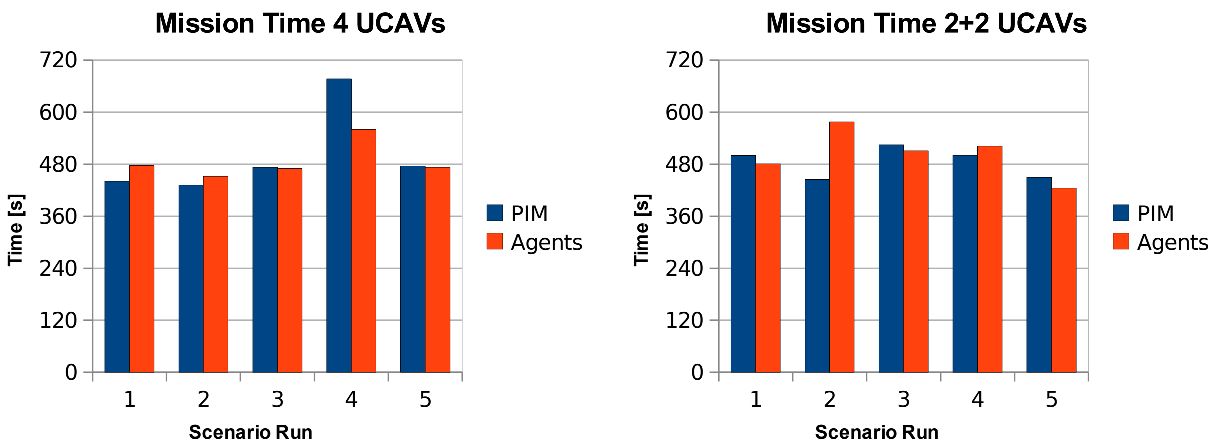


Fig. 4. The overall mission time MT was in average 486s for the mutli-agent based system and 542s for the PIM agents in the 4 UCAVs scenario. In the 2+2 UCAVs scenario the average mission time of the multi-agent solution was 503s while the PIM agents had the average time of 484s.

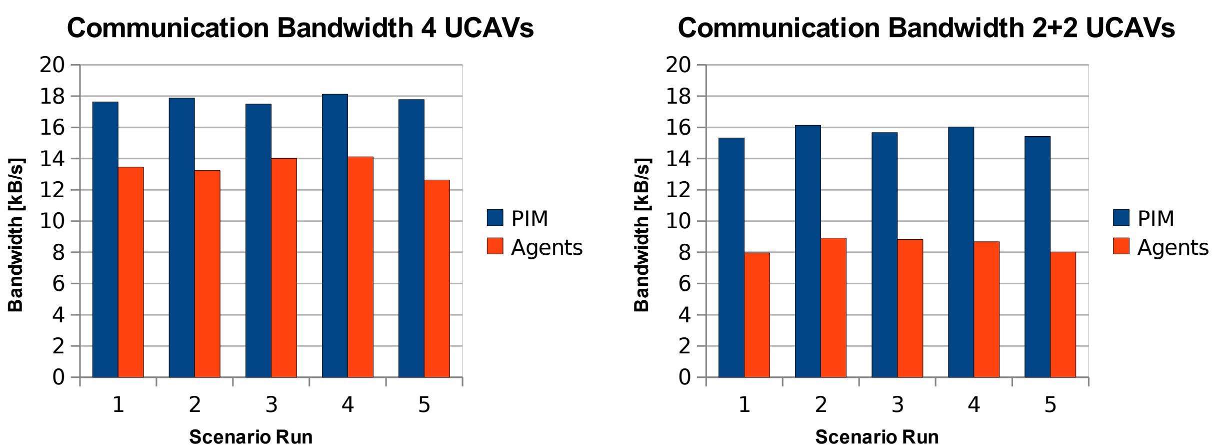


Fig. 5. Communication bandwidth. Average values for the 4 UCAVs scenario are 13.5 kB/s for multi-agent approach and 17.7 kB/s for PIM agents. In the 2+2 UCAVs scenario the multi-agent solution used in average approximately 8.6 kB/s and PIM agents used the bandwidth of 15.7 kB/s.

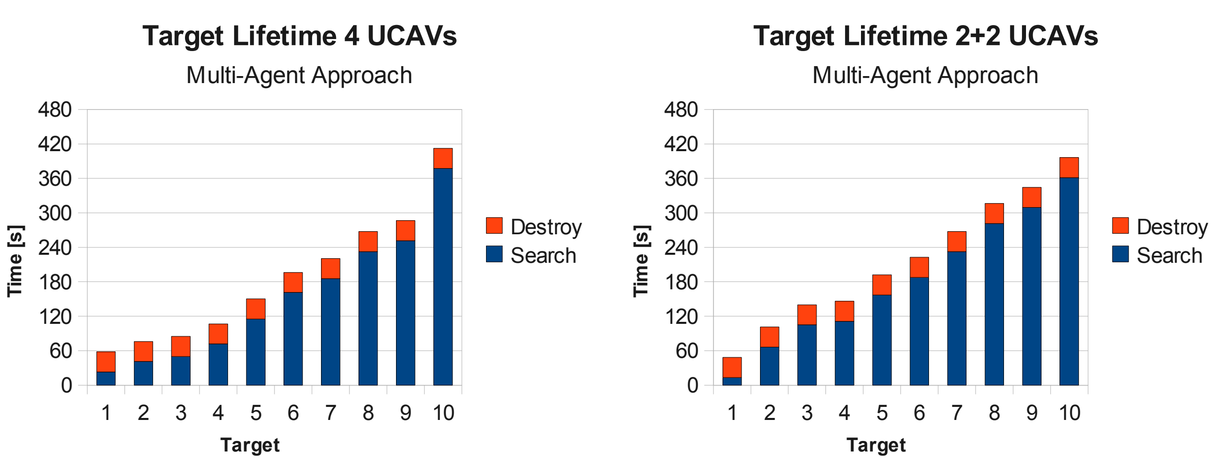
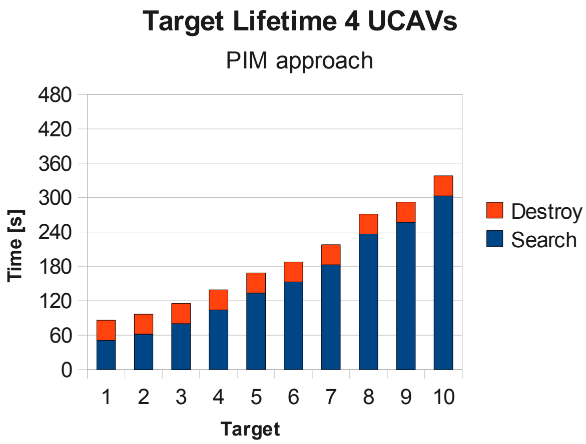


Fig. 6. Target lifetime distribution in multi-agent based solution. Average values are for 4 UCAVs: 151s and for 2+2 UCAVs: 182.5s.



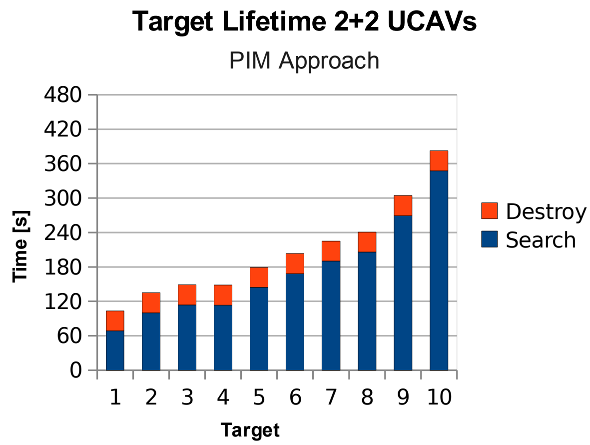


Fig. 7. Target lifetime distribution in PIM based solution. Average values are for 4 UCAVs: 156.3s and for 2+2 UCAVs: 172.2s.

[20] A. Richards, J. Bellingham, M. Tillerson, and J. How, “Coordination and control of multiple UAVs,” in *AIAA Conf. on Guidance, Navigation and Control*, 2002.

[21] M. Alighanbari, Y. Kuwata, and J. How, “Coordination and control of multiple uavs with timing constraints and loitering,” in *American Control Conference, 2003. Proceedings of the 2003*, vol. 6, june 2003, pp. 5311 – 5316 vol.6.

[22] H. V. D. Parunak and et al., “Digital pheromones for autonomous coordination of swarming uav’s,” 2002.

[23] M. Jain, M. E. Taylor, M. Yokoo, and M. Tambe, “DCOPs meet the real world: Exploring unknown reward matrices with applications to mobile sensor networks,” in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

[24] K. M. Ford, J. Allen, N. Suri, P. Hayes, and R. Morris, “PIM – a novel architecture for coordinating behavior of distributed systems,” *AI Magazine*, vol. 2010, June 2010.

[25] M. Pěchouček and D. Šišlák, “Agent-based approach to free-flight planning, control, and simulation,” *IEEE Intelligent Systems*, vol. 24, no. 1, January-February 2009.