A framework for a hierarchical model of cooperation between unmanned airplanes

Pawel Rotter and Wojciech Chmiel

Department of Automatics and Biomedical Engineering, AGH University of Science and Technology, Krakow, Poland

ABSTRACT

In this article we propose a framework for a hierarchical model for cooperation between unmanned airplanes in large groups. We argue that the swarm approach to cooperation between a large number of mobile robots is ineffective when robots can be equipped with efficient communication, precise location hardware and with complex control algorithms. The proposed model, analogous to the organisation of a team of people, is an intermediate solution, between the swarm approach and central control. The bottom layer of the model includes unmanned airplanes that perform tasks related to the mission goals, such as capturing images, looking for objects through image analysis, etc. These airplanes are organised in teams, and each team is controlled by the superior airplane. Two top layers are ground-based: the central computer and the human operator, and their functions are mostly defining the goals, planning and optimisation of the mission. We present two examples of response to events, related to failure or loss of an airplane. We discuss how the proposed model can manage possible interruptions of communication and security issues, and how collision avoidance can be implemented. Finally, we point out that, although the control structure is different than in the swarm approach, swarm intelligence can still be used to optimise missions.

Introduction

Unmanned aerial vehicles (UAVs) are in rapid development, and each year the capabilities of UAVs rise and prices fall. Currently, the cost of the airframe of an electric UAV capable of carrying a 0.5 kg payload and flying for several hours on one battery set does not exceed $500 and even cheaper solutions are available on the market.\(^1\) Even after adding the cost of gear (such as engine, servos and battery) and electronic hardware, the price is low enough to enable development of large groups of UAVs for a variety of applications. In the near future, we may expect to see swarms of hundreds and maybe even thousands of UAVs, supervising or searching large areas. The wide range of possible applications is mostly related to remote sensing, where UAVs are already in use (Baiocchi, Dominici, Milone, & Mormile, 2014; Borgogno & Gajetti, 2017; Rossi, Mancini, Dubbini, Mazzone, & Capra, 2017). However, using a group of small UAVs has a number of advantages when comparing to a single powerful one (Maza, Ollero, Casado, & Scarlatti, 2015): multiple simultaneous observations are possible and tasks such as exploration and search can be executed faster. In a group of UAVs high fault tolerance can be achieved even if the reliability of a single vehicle is relatively low.

On the other side, controlling a large group of UAVs is a challenging task. When looking for a solution to this problem, many researchers proposed the exploitation of a physical analogy between mobile robots and particles that are simulated in swarm intelligence methods (Rubenstein, Cornejo, & Nagpal, 2014), or between robots and nodes of self-organised sensor networks (Daniel, Rohde, Goddemeier, & Wietfeld, 2011; Goddemeier, Daniel, & Wietfeld, 2012). Swarm intelligence is a group of optimisation methods inspired by nature, where the behaviour of swarms of animals is imitated in order to solve optimisation problems. The most popular algorithms are ant colony optimisation (Dorigo, Maniezzo, & Colomi, 1996), the bees algorithm (Pham & Castellani, 2009), and the Particle Swarm Optimisation (PSO) algorithm, inspired by the behaviour of flocks of birds and swarms of fish (Kennedy, Eberhart, & Shi, 2001), although there are also many others (Pham & Castellani, 2014). At first glance, swarm intelligence seems well suited to the problem of controlling large groups of robots.
It has been used in optimisation problems related to UAVs, such as path planning (Roberge, Tarbouchi, & Labonte, 2013; Yangguang, Mingyue, Chengping, & Hanping, 2013), but recent miniaturisation and the decreasing price of mobile robots have made it possible to imitate swarms of animals not only in computer simulations, but to develop robots that imitate swarms of animals species. A single entity – a (micro/nano) robot – may have low processing power but it contributes to the high joint intelligence of the swarm. Such an approach to a robot’s cooperation is attractive, first because it ensures full autonomy of the group of robots, and secondly because it accepts the low capabilities of a single robot. Finally, it is exciting because it follows a vision of physical realisation of collective intelligence developed decades ago by science fiction writers such as Stanislaw Lem (The Invincible, 1974) or Isaac Asimov (Hallucination, 1985).

However, in this article we argue that methods inspired by nature where entities do not demonstrate any hierarchy are ineffective for those robots that are large enough to be equipped with efficient communication and with complex control algorithms. The lack of hierarchy in colonies of ants is undoubtedly related to the fact that ants are able to communicate only locally. An ant manager, if it existed, could not obtain detailed information about the location (such as coordinates) of its subordinate ants or food resources, and neither could it transmit precise orders to ant subordinates. The swarm approach is adequate for robots with limited capability, such as microrobots or nanorobots, but unmanned airplanes, with possible payload sufficient to include relatively powerful computers, need another control model in order to fully exploit their potential. Moreover, unlike many other types of mobile robots, unmanned airplanes operate outside, so they can be equipped with GPS modules and be aware of their location. In addition, even distant airplanes from the same swarm, unlike ground robots, are usually in line of view, which makes communication much easier.

In the related literature there are reported attempts to introduce a hierarchical structure in a group of UAVs. Maza et al. (2015) classified multi-UAV systems based on the coupling between the UAVs into three main categories: physical coupling, swarms and formations – the intermediate class, where each member of the team must keep user-defined distances with other members. The hierarchical model proposed in the next section can be counted to this class. The first works related to formations date from early 2000s and most authors focused on maintaining the set distance between UAVs of the formation (Giulietti, Pollini, & Innocenti, 2000). Then more complex multi-UAV systems were considered, for example composed of different types of UAVs such as helicopters and airships (Ollero et al., 2005).

In this paper we propose a general scheme for cooperation that can be used as a framework for designing specific systems of UAVs. We argue that the hierarchical approach to the control of a set of unmanned airplanes is more effective than the swarm approach and we propose how the issues that impel the use of the swarm approach can be solved in the hierarchical system. Our hierarchical framework is based on human organisation. Let us notice that no complex task in human history, from building the Egyptian pyramids to contemporary military operations, including group manned flights, could not have been achieved effectively without coordination and without a hierarchy established among individuals. One could say that the reasons for hierarchy among humans, and many animals, are the variety of competences and different levels of knowledge, which is not the case for UAVs, where each item can include the same intelligence. This is true, but even in a team of people with similar competences, hierarchy is needed to perform a complex task effectively. Apart from taking inspiration from human organisation, in the proposed model we use an advantage of robots over people: in the case of UAVs, a superior machine can back up all its knowledge and abilities, and if destroyed, it may be immediately replaced with another UAV, if both are equipped with the same or similar hardware. The difference in structure between the swarm approach and the hierarchical model is outlined in Figure 1.

Figure 1. Information channels in the communication model based on the swarm approach (a) and in the hierarchical model (b).
The hierarchical model

Structure of the model

The proposed hierarchical model is presented in Figure 2. The basic elements are drones that directly work on mission-specific tasks, for example, they capture digital images, analyse them in the quest for objects that they are looking for, deploy nodes of sensor networks, etc. In this article we refer to them as worker UAVs. Worker UAVs are divided into teams, and each team is led by a drone that we call superior UAV. It coordinates the operation of all worker UAVs in its team and sends them the parameters of trajectory that they should follow.

Apart from functionalities listed in Figure 2, elements of the system should be capable of responding to unexpected events, including failures of other elements or loss of communication. In this article we will provide several examples of such emergency procedures.

Exchange of information

In the proposed model, information is basically exchanged between elements at adjacent levels: between the operator and the base, between the base and superior UAVs, and between a superior UAV and its subordinate worker UAVs. In addition, some kind of information can be exchanged directly among elements of the same level. For example, in order to avoid a collision between two worker UAVs, modification of their trajectories should be implemented immediately and direct negotiation between worker UAVs is faster and more efficient than arranging this through the superior UAV. The general rule is analogous to army organisation, where the orders are passed down the hierarchy, reports are transmitted upwards, and some information is exchanged between members of the same team. The details of information flow are presented in Table 1.

Examples of reactions to events

In this section we provide two examples of reactions to events. Since in this paper we present a general model, we do not consider mission-related events, where reactions should be defined precisely for a particular case, such as, for example, responding to specific behaviour of targets during a tracking mission. We focus on two events: failure or loss of a worker UAV, and failure or loss of a superior UAV.

Since we assume a large number of low-budget UAVs, these events are rather common and related procedures should be anticipated. Both examples apply to a wide class of missions called visit known locations, according to UAV route planning classification proposed in Pitre, Li, & Delbalzo, 2012. This includes goals such as taking a series or aerial photos of an area, or deployment of sensor networks. A similar scheme may be applied in search missions.

Failure or loss of a worker UAV

If the superior UAV stops receiving signals (such as current position) from one of the worker UAVs, it interrogates all the other workers to find out if any of them receives the signal, and, if not, it is assumed that the worker UAV is lost. If the reason is only communication failure, the impaired worker UAV should return autonomously to the base, but is considered lost for the further duration of the mission. In reaction to this event, the superior UAV changes the trajectories of its workers, in a way that depends on whether connection with the base is available:

The operator
Functions: defining mission goals, including input data, such as operation area, etc. Re-defining mission goals in response to events.

The base
Functions: planning and optimisation of a mission, coordination of superior UAVs (when connection is available), presentation of status to the operator.

Superior UAVs
Functions: coordination of subordinate worker UAVs, optimisation of the team operation, re-planning team operation in response to events.

Worker UAVs
Functions: mission-specific tasks, e.g. capturing digital images, searching for and recognising objects, etc. Following trajectory imposed by superior UAV. Collision avoidance.

Figure 2. Hierarchy levels of the model proposed in the article.
| Information sent downwards: | Information sent upwards: |
|-----------------------------|-----------------------------|
| **Operator**                | **Data continuously sent:** |
| Before the mission:         | • Mission status: number of UAVs in the air, percentage of mission completed, likelihood of achieving mission goals (if applies), estimated time to mission completion, etc. |
| Goals and parameters of the mission. For example, the parameters for an observation mission (coverage by aerial images) are: area to observe, altitude of observation, etc. | **Statements:** |
| During the mission:         | • Task-specific statements, e.g. coordinates and properties of objects that were found |
| • Re-defining mission goals and parameters | • Additional information sent from UAVs on request, filtered and processed if needed |
| • Mission interruption      | • Mission completed / mission failed |
| • Request for additional information (see right column for examples) | |
| • Request for manual control over a specified superior UAV (then the base computer sends follow trajectory command) | |
| **The base computer**       | **Data continuously sent:** |
| Before the mission:         | • Current position |
| Parameters of the mission divided into superior UAVs. For example, the parameters for an observation mission are areas to be covered by different teams of UAVs. | **Statements:** |
| During the mission:         | • Task completed, the team returns to the base unless new command from the base is received |
| • Re-assignment of tasks (new parameters of the mission divided into superior UAVs) | • Team unable to complete the task – need to re-assign tasks to UAV teams, e.g. assign smaller observation area to incomplete team. The reason can be faster energy consumption than initially forecast (e.g. due to weather conditions) or loss of some worker UAVs |
| • Mission interruption      | • Task-specific statements, e.g. coordinates and properties of objects that were found |
| • Request for additional information (see right column for examples) | • Additional information send on request, such as positions of worker UAVs, real-time images from superior UAV camera or re-transmitted data from any worker UAV from its team |
| • Follow trajectory command (switches off autonomy of team, then superior UAV follows trajectory parameters sent by the base and sets its worker UAVs in a formation with parameters ordered by the base) | |
| **Superior UAVs**           | **Data continuously sent:** |
| During the mission:         | • Current position |
| • Parameters of trajectory to follow | • Estimated energy supply |
| • Task-specific commands, such as start/stop image recording or drop the node of sensor network | **Statements:** |
| • Go to location (x, y, z) and circle around until receiving the next command (or until energy reserve is sufficient for return only – then return autonomously) | • Unable to follow trajectory (difference between desired and actual position exceeds threshold) |
| • Autonomous return to the base command | • Trajectory was changed to avoid collision |
| **Worker UAVs**             | • Task-specific statements, e.g. the query object was found at coordinates (x,y) |
If connection with the base is available, the event is reported and the base decides whether the mission will be rescheduled. If so, the superior UAV modifies the trajectories of its team in order to cover a smaller area without density change (Figure 3(a)) and the base reassigns tasks to teams, possibly re-defining an area to observe.

If connection with the base is not available, or the base decided not to reschedule the mission, the superior UAV modifies the trajectories of its team in order to cover the same area but at lower density (Figure 3(b)).

The description of this example is not exhaustive, and the details of the algorithm depend on implementation. For example, if a connection with the base is not available, the superior UAV may check the margin of energy of its team and decide to extend the mission time in order to complete the task with the assumed density.

Failure or loss of the superior UAV

In systems that require high levels of robustness, it is worth considering equipping the entire worker UAVs with hardware identical to that of the superior UAV, and sending the entire content of the superior UAV’s memory to the worker UAVs continuously. Then, if the superior UAV fails, the worker UAV that is closest to the latest location of the superior takes over its role. Before this, it exchanges information with other worker UAVs to confirm that all of them have lost the superior’s signal and that only one worker becomes the superior UAV. This solution is costly, first because workers must be equipped with the same processing power and transmission capabilities as the superior, and second because the entire content of the superior’s memory needs to be copied to the worker UAVs continuously. It is mostly adequate for military missions, where the possibility of some UAVs being shot down is considered, and the overall cost of the system is a minor criterion.

In cost-effective systems, the equipment of worker UAVs is more basic than those of the superior. Workers need less processing power than the superior, which needs to control all workers, optimise their trajectories, and recalculate them in response to events, etc. Worker UAVs may be equipped with communication modules for short range only, to communicate with their superior and other workers from the same team, while in the case of the superior, long-range communication is very desirable, to send data to the base and to receive commands, e.g. when the mission goal or parameters are modified. In such systems, a superior UAV cannot be replaced by a worker UAV. In this case, if workers lose the signal from their superior, they send a request to all other superiors to take them over. Then, superior UAVs negotiate between each other to decide how to divide the unassigned workers based on their proximity, and taking into consideration that a single superior UAV may not be able to control too many workers because of limited processing capabilities. In case some worker UAVs cannot be taken over by any superior, they can automatically return to the base (we thank the anonymous reviewer who suggested this option). If a connection with the base is available, the mission plan is re-optimised by the central computer. Otherwise, it is done using the resources of superior UAVs.

The main issues that impel the use of the swarm approach for groups of mobile robots, and alternative solutions that can be applied by the hierarchical model.

The autonomy of mobile robots is attractive because it eliminates the necessity for human supervision. It is especially desirable for large groups of cooperating
robots, where manual control would require a lot of human resources, even if done remotely. Nevertheless, we argue that the autonomy of a group of robots does not need to be based on the autonomy of single robots. In this section we present the main issues that impel the use of the swarm approach, and we argue that alternative solutions exist, which can tackle with these issues and be more effective than the swarm approach.

Communication issues

When mobile robots are autonomous, they can operate without constant and robust communication. This makes autonomy very attractive especially for robots that work in buildings, underwater or underground, but also for unmanned ground vehicles operating outside, where communication can be disrupted by terrain forms and other obstacles.

For UAVs, especially for unmanned airplanes, which always operate in open space (unlike unmanned helicopters or multicopters), it is usually easier to ensure reliable wireless connection than for other types of mobile robots. However, even in this case communication is not always reliable because:

- for long missions the distance from the base requires long-range transmission, which raises costs and payload of a single drone and consumption of on-board energy
- in mountain areas, line of sight between a drone and the base may be obstructed by terrain

In our hierarchical model, we assume that worker UAVs do not communicate directly with the base but with their superior UAV only, which is relatively close and should remain in line of sight. Nevertheless, worker UAVs should be equipped with an emergency procedure in case communication with their superior UAV is lost, which at least ensures autonomous return to the base area, but it may also include trying alternative channels of communication, such as Global System for Mobile Communications or even satellite transmission in high grade systems.

In each team, only the superior UAV communicates directly with the base. The superior has a certain degree of autonomy, so when connection with the base is lost, the team may continue the mission (although cooperation with other teams is limited, especially when other superior UAVs report unexpected events). Disruptions of communication between superior UAVs and the base that result from long distance and terrain obstructions in mountain areas can be solved by delegating some airplanes as transmitters. They can dynamically change their position based on an elevation map and the current positions of superior UAVs.

Let us note that introducing hierarchy allows global communication among UAVs because it radically reduces the amount of information exchanged (worker UAVs from different teams do not communicate directly with each other, except for collision avoidance). In large systems with no hierarchy, limitation of communication range is necessary in order to avoid jamming, but it impedes the online exchange of data with remote robots.

Security issues

If the operation of a drone is based on a signal transmitted wirelessly, there is a risk that attackers can take control. A swarm of autonomous UAVs may seem robust in the face of this risk, since it operates without external control, but in fact autonomous operation can be even more risky. If a drone is controlled wirelessly, transmission of data can be encrypted and moreover, if asymmetric cryptography is used, hijacking one drone and its key does not allow to the control signals for other drones to be encoded. In contrast, GPS-based flights are more susceptible to spoofing when they are autonomous, because of the lack of external control. GPS spoofing is a technique of overriding satellite GPS signals with an attacker’s transmission in order to spoof the receiver. This risk, and countermeasures, has been well known and discussed for more than 10 years (Warner & Johnston, 2003), but it received more attention after the incident with US drone RQ-170 Sentinel, seized by Iran on 4 December 2011. Reportedly, it was hijacked by jamming the control signals in order to switch the drone into autonomous mode, and then using GPS spoofing techniques to make the drone land on hostile territory instead of at its home base (Peterson & Faramarzi, 2011). Although this version was never officially confirmed, GPS spoofing is certainly a real risk, and using it to take control of UAVs has been demonstrated (Shepard, Bhatti, & Humphreys, 2012). Countermeasures based on analysis of location changes in time, such as that proposed in (Warner & Johnston, 2003), can be outsmarted by sophisticated spoofing (Tippenhauer, Pöpper, Rasmussen, & Capku, 2011). Let us underline that even full autonomy of a UAV does not protect against hijacking.

In applications that demand strong security, where a group of UAVs should be robust in the face of GPS jamming and spoofing, we propose that superior UAVs are equipped with alternative location

---

The requirement for encryption seems obvious in military applications, but in the past many intelligence drones transmitted unencrypted signals, which were intercepted using simple software such as SkyGrabber.
techniques. Some of these techniques, such as Inertial Navigation Systems (Woodman, 2007) and Terrain Contour Matching (Golden, 1980), which was an early version of Digital Terrain Systems (Fountain, 1997), were introduced a long time before GPS, and for security reasons they are still in use alongside satellite navigation in many military airplanes, helicopters and cruise missiles. The basic alternative technique is Inertial Navigation System (INS), which includes accelerometers and gyroscopes. It can provide exact and robust location data but requires frequent calibration. Otherwise, it produces an increasing error, which even in high quality military INS increases by as much as 1.5 km per hour of flight (Wilkinson, Brookes, Price, & Godfrey, 2009). Therefore it should be accompanied with a location system appropriate for calibrating it, such as Digital Terrain System (DTS), which compares a digital elevation map in the computer’s memory with data from on-board radar (Jun, Weisong, & Yirong, 2011) and is used as a part of the widespread TERPROM® system (Cowie, Wilkinson, & Powlesland, 2008). Computer vision methods are another important technique that can be used to calibrate inertial navigation. Such methods either match the map stored in the memory with an image from the camera (Sim, Park, Kim, Lee, & Kim, 2002), or they calculate vehicle motion based on tracking some features in the image (Angelopoulou & Bouganis, 2014). An advantage of computer vision methods is the possibility of using the same system for location and for tasks such as object finding or tracking, as proposed in (Ludington, Johnson, & Vachtsevanos, 2006). The best way to achieve robustness of the system is multimodal location, i.e. simultaneous use of several different systems and combining the output data. For example, DTS does not provide location data over flat terrain, while vision-based navigation can be blinded by bad weather conditions (Jun et al., 2011).

Because of high price and additional weight, it is reasonable to implement alternative location technology in a selected subgroup of UAVs rather than in every UAV in a numerous group. Remaining UAVs can be equipped with GPS-based localisation, along with additional verification of position based on the distance from those UAVs that use alternative, robust technologies.

**Collision avoidance**

In large systems, collision avoidance is an example of a task for which global control is ineffective. One reason is limited capacity and speed of communication channels. Since collision avoidance algorithms must run continuously for all UAVs, sending the position data from UAVs to the central computer, then performing calculations and sending back the control signal is ineffective. Moreover, global calculation would be time-consuming, while local correction of trajectory is relatively simple. In swarms of autonomous UAVs, collisions are avoided locally, based on information from on-board sensors, such as laser scanners, radar or by using vision-based methods, as proposed in (Chung, Oh, Shim, & Sastry, 2011), where the cost function of the linear-quadratic regulator controller is extended with penalty when the predicted trajectory is close to other objects.

In our model, the first measure is assignment of UAVs from different teams to disjoint areas. It may happen that the optimal course of the mission requires intersecting of two teams during manoeuvres, and then their superior UAVs negotiate different altitudes for both teams. The trajectories of UAVs from the same team do not intersect, because they are scheduled at the level of their superior UAV, taking into consideration collision avoidance. These rules should, in theory, exclude collisions, but in some situations they may fail due to temporary loss of communication, GPS errors or sudden gusts of wind. For this reason, additional measures may be optionally implemented at the level of worker UAVs, to modify trajectory based on data from on-board sensors in the event of possible collision with another UAV or any other obstacle.

**Robustness to failure or loss of individual robots**

In swarms of mobile robots, work is divided locally among the closest individuals and not pre-assigned to them. As a consequence, a swarm is robust to loss of individuals, and this feature is crucial for many applications. A decimated swarm may be less effective, but execution of all tasks continues. In contrast, if a group of robots follows a schedule and tasks are assigned to individuals in advance, failure of or damage to a single robot means that its task remains unassigned.

In our hierarchical approach, the loss of a worker UAV leads to its work being re-scheduled and assigned to other worker UAVs by the superior UAV. If one of the superior UAVs is lost or damaged, its tasks are assigned to a worker UAV from its team (in robust systems, where all worker UAVs have the potential capability of superior UAVs) or to another superior UAV (in cost-effective systems, where the equipment of worker UAVs is more basic than that of superior UAVs).

**Collective intelligence for decision making**

A swarm of intelligent robots can perform complex tasks using the collective intelligence of its robots, even if they communicate only locally, without any
central control, as in the recent spectacular example of Kilobots, developed at Harvard University (Rubenstein et al., 2014), where a swarm of cooperating 1024 minirobots was able to self-organise in defined shapes, although a single robot communicates with its neighbours only and follows just a few simple rules. Although such self-organisation is impressive, clearly the result is far from the optimal performance of a task, which could be completed much more efficiently if central control were possible. For example, it takes Kilobots more than 10 h to form a shape.

Let us notice that, when it comes to collective intelligence, there is no deep analogy between swarms of unmanned airplanes and flocks of birds or swarms of insects: animals are looking for food in a physical space that is identical to the optimisation space, and it has attributes that can be interpreted as the goal function (smells, amount of food, and concentration of pheromones). In contrast, in the case of unmanned airplanes, even in a basic search task there are no attributes that could define a goal function in the physical space, such as for example an object’s proximity, which remains unknown.

More centralised approaches, including our hierarchical model, do not exclude the use of nature-inspired algorithms for optimisation of the mission. The use of these methods is even more flexible and effective when the structure of a virtual swarm used for the optimisation procedure does not need to correspond with physical robots. For example Pitre et al. (2012) used a modified Particle Swarm Optimisation to optimise joint missions of two unmanned airplanes, assuming centralised fusion of information.

Conclusions

In this article we proposed a framework for a hierarchical model for cooperation between unmanned airplanes. It is based on the observation that swarms of autonomic agents inspired by nature are ineffective for those robots that are large enough to be equipped with efficient long-range communication and complex control algorithms, and where robots can be located based on, for example, satellite navigation, with sufficient precision, compared to their operation range. Undoubtedly, the swarm approach can be effective for groups of simple, especially miniature, robots, with very limited processing and communication capabilities and that are not aware of their location. Nevertheless, unmanned fixed-wing airplanes are of a size sufficient to carry medium-range wireless communication hardware, and operating outside makes it easy to obtain global coordinates from a satellite navigation system. In order to make use of these features, a model that is more complex than a flat structure is needed.

We argue that the autonomy of the group of UAVs does not need to be based on the autonomy of single UAVs. Instead of the swarm approach, where autonomy is achieved by methods inspired by flocks of animals, we proposed a hierarchical model, similar rather to the organisation of a team of people, which is an intermediate solution between the swarm approach and central control. We introduced four layers, where unmanned airplanes are organised in a two-layer structure: the bottom layer consists of the worker UAVs, which perform mission tasks, and the top layer includes superior UAVs, which coordinate and control the worker UAVs. The ground-based part includes two layers: the base computer and the human operator, and their functions are mission defining, planning and optimisation, as well as re-planning the mission in response to events that occur during the operation. Superior UAVs have a certain degree of autonomy, which allows them to lead their teams and complete the mission if, for example, communication with the ground base fails. Worker UAVs are also equipped with emergency procedures that at least allow their autonomous return to the base if they stop receiving instructions from their superior UAV.

We discussed two examples of reaction to events in the hierarchical model, both related to failure or loss of an airplane. We chose these events as examples because they are common in large groups of low-budget UAVs and, unlike the swarm approach; the hierarchical model requires an elaborated scheme of reaction. We pointed out issues that could, in common opinion, be considered in favour of the swarm approach. We argued that they can be handled in our hierarchical model as well. We discussed how the hierarchical model can manage possible interruptions of communication, security issues, and implementation of collision avoidance, and how it can make a group of UAVs robust to failure or loss of single airplanes. Finally, we pointed out that swarm intelligence can be used for optimisation of a mission regardless of the structure of control in the group of airplanes.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was partially supported by the European Union within European Social Fund (and coordinated by the Malopolska Regional Development Agency), grant number MARR/3251/2014/DZPP and by the National Centre for Research and Development, grant number DZP/RID-I-68/14/NCBIR/2016.
**References**

Angelopoulou, M.E., & Bouganis, C.S. (2014). Vision-based egomotion estimation on FPGA for unmanned aerial vehicle navigation. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6), 1070–1083.

Baiocchi, V., Dominici, D., Milone, M.V., & Mornmle, M. (2014). Development of a software to optimize and plan the acquisitions from UAV and a first application in a post-seismic environment. *European Journal of Remote Sensing*, 47(1), 477–496.

Borgogno, M.E., & Gajetti, M. (2017). Preliminary considerations about costs and potential market of remote sensing from UAV in the Italian viticulture context. *European Journal of Remote Sensing*, 50(1), 310–319.

Chung, H., Oh, S., Shim, D.H., & Sastry, S.S. (2011). Toward robotic sensor webs: Algorithms, systems, and experiments. *Proceedings of the IEEE*, 99(9), 1562–1586.

Cowie, M., Wilkinson, N., & Powlesland, R. (2008). *Latest development of the TERPROM* digital terrain system (DTS). Paper presented at the Position, Location and Navigation Symposium, Monterey, CA.

Daniel, K., Rohde, S., Goddemeier, N., & Wietfeld, C. (2011). Cognitive agent mobility for aerial sensor networks. *Sensors Journal, IEEE*, 11(11), 2671–2682.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1), 29–41.

Fountain, J.R. (1997). *Digital terrain systems*. Paper presented at the IEE Colloquium on Airborne Navigation Systems Workshop, London.

Giulietti, F., Pollini, L., & Innocenti, M. (2000). Autonomous formation flight. *IEEE Control Systems Magazine*, 20(6), 34–44.

Goddemeier, N., Daniel, K., & Wietfeld, C. (2012). Role-based connectivity management with realistic air-ground channels for cooperative UAVs. *IEEE Journal on Selected Areas in Communications*, 30(5), 951–963.

Golden, J.P. (1980). *Terrain contour matching (TERCOM): A cruise missile guidance aid*. Paper presented at the SPIE Image Processing for Missile Guidance.

Jun, Z., Weisong, L., & Yirong, W. (2011). Novel technique for vision-based UAV navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 47(4), 2731–2741.

Kennedy, J., Eberhart, R., & Shi, Y. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Ludington, B., Johnson, E., & Vachtsevanos, G. (2006). Augmenting UAV autonomy. *IEEE Robotics & Automation Magazine*, 13(3), 63–71.

Maza, I., Ollero, A., Casado, E., & Scarlatti, D. (2015). Classification of multi-UAV architectures. In K.P. Valavanis & G.J. Vachtsevanos (Eds.), *Handbook of unmanned aerial vehicles* (pp. 953–975). Springer, Dordrecht.

Ollero, A., Lacroix, S., Merino, L., Gancet, J.,Wiklund, J., Remuss, V.,…. Caballero, F. (2005). Multiple eyes in the skies. Architecture and perception issues in the COMETS unmanned air vehicles project. *IEEE Robotics and Automation Magazine*, 12(2), 46–57.

Peterson, S., & Faramarzi, P. (2011). Exclusive: Iran hijacked US drone, says Iranian engineer. *Christian Science Monitor*, 15 December 2011.

Pham, D.T., & Castellani, M. (2009). The Bees algorithm - modelling foraging behaviour to solve continuous optimisation problems. *Journal of Mechanical Engineering Science*, 223(12), 2919–2938.

Pham, D.T., & Castellani, M. (2014). Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms. *Soft Computing*, 18(5), 871–903.

Pitre, R.R., Li, X.R., & Delbalzo, R. (2012). UAV route planning for joint search and track missions - an information-value approach. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3), 2551–2565.

Roberge, V., Tarbouchi, M., & Labonte, G. (2013). Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, 9(1), 132–141.

Rossi, P., Mancini, F., Dubbini, M., Mazzone, F., & Capra, A. (2017). Combining nadir and oblique UAV imagery to reconstruct quarry topography: Methodology and feasibility analysis. *European Journal of Remote Sensing*, 50(1), 211–221.

Rubenstein, M., Cornejo, A., & Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198), 795–799.

Shepard, D.P., Bhatti, J.A., & Humphreys, T.E. (2012). Drone hack. Spoofing attack demonstration on a civilian unmanned aerial vehicle. *GPS World*, August 2012.

Sim, D.G., Park, R.H., Kim, R.C., Lee, S.U., & Kim, I.C. (2002). Integrated position estimation using aerial image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 1–18.

Tippenhauer, N.O., Pöpper, C., Rasmussen, K.B., & Capku, S. (2011). On the requirements for successful GPS spoofing attacks. Paper presented at the 18th ACM conference on Computer and communications security, New York.

Warner, J.S., & Johnston, R.G. (2003). GPS spoofing countermeasures. Homeland Security Journal, 25(2): 19-27, 2003.

Wilkinson, N., Brookes, T., Price, A., & Godfrey, M. (2009). Latest developments of the TERPROM* Digital terrain system*. Paper presented at the Joint Navigation Conference, Orlando, Florida, USA.

Woodman, O.J. (2007). *An introduction to inertial navigation* (UCAM-CL-TR-696). Retrieved from [http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf](http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf)

Yangguang, F., Mingyue, D., Chengping, Z., & Hanping, H. (2013). Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization. *Systems, Man, and Cybernetics: Systems, IEEE Transactions On*, 43(6), 1451–1465.