



Distributed Simulation Environment of Unmanned Aerial Systems for a Search Problem

Stanisław Skrzypecki^(✉), Dariusz Pierzchała, and Zbigniew Tarapata

Cybernetics Faculty, Military University of Technology, Warsaw, Poland
{stanislaw.skrzypecki,dariusz.pierzchala,
zbigniew.tarapata}@wat.edu.pl

Abstract. In the paper the programmable simulation environment for Unmanned Aerial Systems (UAS) and the preliminary research of UAV's swarm applied for a search problem in a large-scale terrain are presented. Proposed approach is based on distributed simulation, multiagent systems and multiresolution modelling in order to perform studies on UAV modelled as a swarm with determined, autonomous, combined behaviours. Some software agents have been simulated in constructive component while others have been controlled by virtual simulator (VBS3) with interoperability provided by DIS or HLA protocols. Furthermore, the biological inspired algorithms (e.g. PSO algorithm and other modifications) have been used to model UAVs' actions. The preliminary results lead to conclusion of usability of the environment in solving search problem and modelling UAV's movements and behaviours.

Keywords: UAS · UAV's swarm · Distributed simulation · Autonomous systems · Biological inspired algorithms · PSO algorithm

1 Introduction

Regarding rapid development of teleinformatics and mechatronics the technical parameters of Unmanned Aerial Vehicle (UAV) are significantly improving. The same process consequence with rising number of opportunities to use them in many others areas, including transport of small-sized goods [6], monitoring and information gathering [16] or entertainment [18]. The wide spectrum of usage also includes military operations: from ISR¹ to combat missions². The development of UAV has led to the concept of Unmanned Aerial Systems (UAS). The definition of UAS means a system consisting of three components: UAV, a control system (operated by human or in autonomous way) and C3 system (communication, command and control) [22]. Such a system, composed of many UAVs, acting jointly to achieve a goal, is often referred to as a drone swarm. The simulations of drone swarm have demonstrated very high effectiveness of the solution [7]. The concepts of using that type of configuration of

¹ Intelligence, surveillance and reconnaissance operations [19].

² E.g. killing of the leader of al-Qaida Anwar al-Awlaki by a U.S. drone attack in 2011.

aerial vehicles is not a most recent idea. Herein, it is worth to mention the Office of the Secretary of Defense's technical report "Unmanned Aircraft Systems Roadmap: 2005–2030". The report assumed continuous development of UAVs - starting from a remotely controlled single aircraft, through those adapting to variable flight conditions and operating in tactical groups (controlled in a distributed manner), ending with fully autonomous swarms of aircrafts [9]. In recent years projects as LOCUST³ and PERDIX⁴ have shown a high level of advancement in the pursuit of these concepts.

The paper presents a programmable simulation environment for modelling Unmanned Aerial Systems and studies confirming the functionality of this environment. As a case study, the problem of using UAVs swarms to search for signal sources in a large-scale terrain is presented. However, development of new algorithms was not the aim of the paper. The simulation environment is based on the original DisSim simulation engine [2, 8], which uses distributed discrete simulation techniques, multi-agent systems and multiresolution modelling. In the current version of the environment, the possibility of collision of aircrafts with each other and real limitations of communication between UAVs have been omitted. Regardless of the above, there is wide range of possibilities to determine a lot of model parameters such as the maximum distance between aircrafts (communication range) or the minimum number of aircrafts remaining in the communication range (communication redundancy). However, the most important assumption of the environment is the ability to prepare models in which individual UAVs are characterized by autonomy in making decisions, deterministic models (predetermined behaviours, movement and arrangement of a whole group) or mixed models. An autonomism is ensured by the use of the biological inspired algorithms, including the Particle Swarm Optimization (PSO) algorithm [3] and its modifications. Likewise, the occurrence of interactions between objects, including those from other simulators, is important in the prepared environment. Interacting simulators may have different command levels (single measures, tactical, operational or strategic) as well as types of simulation (constructive simulation or virtual simulation). For this reason, the environment is required to maintain objects states at various levels and to process states between those levels with interoperability provided with the DIS or HLA protocols.

The paper is organized as follows. In the Sect. 2 the programmable UAS simulation environment was described. It includes discrete event-driven simulation, DisSim as simulation engine, concept of environment and description of particular scenarios implementation. In the Sect. 3 as the case study of environment functionality testing, the problem of using UAVs swarms to search for signal sources in a given area was introduced. Moreover three algorithms solving this problem and results of comparison were presented. The Sect. 4 presents some paper summary and conclusions.

³ Project LOCUST (Low-Cost UAV Swarming Technology) – the group of drones launched in 2016 from the ground launchers (up to 30), each aircraft is capable to execute self-conducting programmed actions [20].

⁴ Project PERDIX – micro-drone swarm consisted of 103 Perdix drones launched in 2017 from three F/A-18 Super Hornets fighters and demonstrated advanced swarm behaviours such as collective decision-making, adaptive formation flying, and self-healing [17]; Perdix drone was initially developed at the MIT University in 2010–2011 [21].

2 Programmable UAS Simulation Environment

2.1 Discrete Event-Driven Simulation

The basis of the programmable UAS simulation environment is the discrete event simulation (DES). DES is characterized by discrete timestep changes resulting from sequentially planned moments of event realizations. The event e from finite set of events E is the algorithmically planned state change of the object (system) at a given simulation time t :

$$e = \langle t, f_e^S \rangle, e \in E, t \in T \quad (1)$$

A set of attributes values of the modelled system object at any time t of the simulation time is called the system state S . It is defined as the following ordered four:

$$S = \langle o, a, v, t \rangle, o \in O, a \in A_c, v \in V_a^c, t \in T \quad (2)$$

$O = \{o = \{id, c\}, c \in C^0\}$ – a set of c class objects identified by id with a unique value in the collection of objects,

C^0 – nonempty set of modelled objects classes,

A_c – nonempty set of attributes defined for the object class $c \in C^0$,

V_a^c – set of acceptable values for attribute $a \in A_c$ of the object class $c \in C^0$,

$t \in T$ – simulation time which is variable from countable subset $T \subset R_+ \cup \{0\}$; it means that $(\exists t_i, t_k) (\neg \exists t_j) (t_i < t_j < t_k)$ so there are two such moments that there is no other between them - it indicates a discrete time passage.

The function of state change f_e^S determines the state $s \in S$ in which the system will be located at the moment t after the realisation of event e :

$$f_e^S : T \times S \rightarrow S \quad (3)$$

2.2 Simulation Engine

In the presented simulation environment the discrete event-driven simulation is being executed through the DisSim package. It is a simulation engine implemented in Java that supports sequentially events (*BasicSimStateChange*) realisation planned for many simulation objects (*BasicSimEntity*) and stored in a shared event calendar (*BasicSimCalendar*). The software offers a dynamic configuration of objects system (plug-in system), a number of auxiliary classes for generating pseudo-random numbers, monitoring and gathering variable states or calculation of statistics on accumulated time series [2, 8]. The DisSim package has been enhanced to include: (a) distributed protocols DIS and HLA; (b) multiagent systems; (c) multiresolution modelling [11]. It made it possible to build programmable environment for constructive simulation allowing studies on behavioural, movement and groups models of UAVs.

As a result of layered environmental concept presented in Sect. 2.3, the DisSim has the following packages model classifying interfaces and classes for various purposes:

- **simspace:**
 - **core** – discrete event simulation execution;
 - **process** – continuous processes modelled with the DES;
- **connector:**
 - **agent** – adaptation of a single agent to participation in distributed simulation;
 - **aggregates** – adaptation of simulation objects to the requirements of multiresolution modelling;
 - **dis** – communication via the DIS protocol with other simulators;
 - **hla** – communication via the HLA protocol with other simulators;
 - **statechanges** – includes classes of messages sent between agents and plugins (statechanges), interfaces of statechanges and enumerations of message types;
- **broker** – implementation of broker and observer design patterns;
- **gui** – graphical user interface;
- **monitors** – variables monitoring and statistical analysis;
- **scripts** – enables plugins system;
- **config** – importing configuration files and managing simulation parameters;
- **random** – pseudo-random variables generation.

2.3 Concept of Environment

The concept of the environment (shown in Fig. 1) is based on the layer structure. The following layers are responsible for different functions: (a) GUI layer; (b) the analytical layer; (c) the aggregation layer; (d) agents layer; (e) distribution layer. The components of each layer has been designed based on interfaces and abstract classes that provide the modularity and extensibility of the solution. Each simulation object is a simulation agent and it can be aggregated at any resolution level at the same simulation time. The prepared agent mechanisms allow the exchange of simulation events (so-called statechanges) with DIS or HLA plugins. These plugins are responsible for further messages exchange.

GUI layer (shown in Fig. 2) is intended to manage the simulation execution, to create and manage UAVs and to observe objects from interoperating simulators. All objects are visualised on the two-dimensional map of a terrain and specified in the internal list of objects. For each object, appropriate attributes such as object type, aggregation level and position or level of damage can be checked. GUI also specifies the visibility of objects with regarding to their resolution level and provides the user with information about the messages transmitted between the simulators (communication event log).

The analytics layer is responsible for gathering information and parameters' values of simulated objects. It allows to examine the characteristics of processes and preparing specific measures. Examples of such measures are as follows: task completion status, task's execution time, lost aircrafts' count, total travelled distance, average distance between aircrafts or energy consumption.

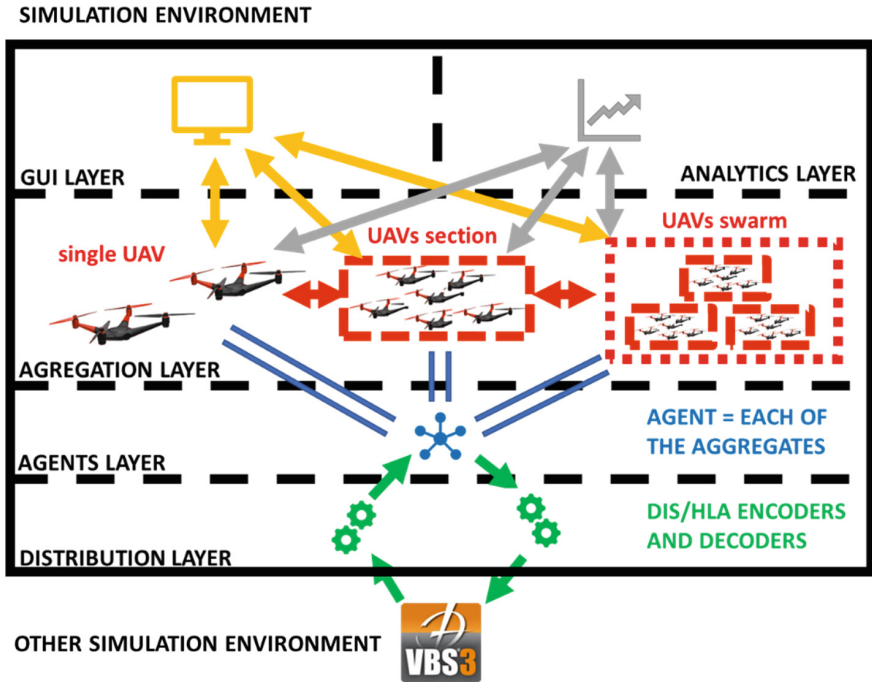


Fig. 1. Concept of UAVs swarm simulation environment

The analytics and the GUI layers interact with the aggregates. The aggregates are the objects immersed in the aggregation layer and characterized by different levels of resolution - from individual UAVs, through sections, to UAV swarms. According to relationships between them, the state change of the object at one level propagates (based on events or intervals) and determines the state of the aggregate that is in the relationship. Processes of transition to the lower or higher resolution using the specified rules are called, respectively, aggregation and disaggregation. They are presented in the Fig. 3. The proposed approach of maintaining different details' levels of simulated objects and defining rules for processing states into an aggregated or disaggregated models is the field of study of multiresolution modelling. That solution enables management of individual UAVs, entire group or as part of the cooperation of this UAV group with other combat measures (as support for ground operations or air assault measures). Moreover, interoperability with simulators on different levels of command will be realized by passing the state of objects at the appropriate level of resolution. High-resolution entity (HRE-type objects) are very detailed – these can be single UAV, and an object with a high aggregation level (LRE-type object which means low-resolution entity) can be UAV swarm [1]. Further, the environment allows the management of aggregates at abstraction levels going beyond the cooperating simulators, which would be represented only in the GUI layer or in the analytical layer for research purposes.

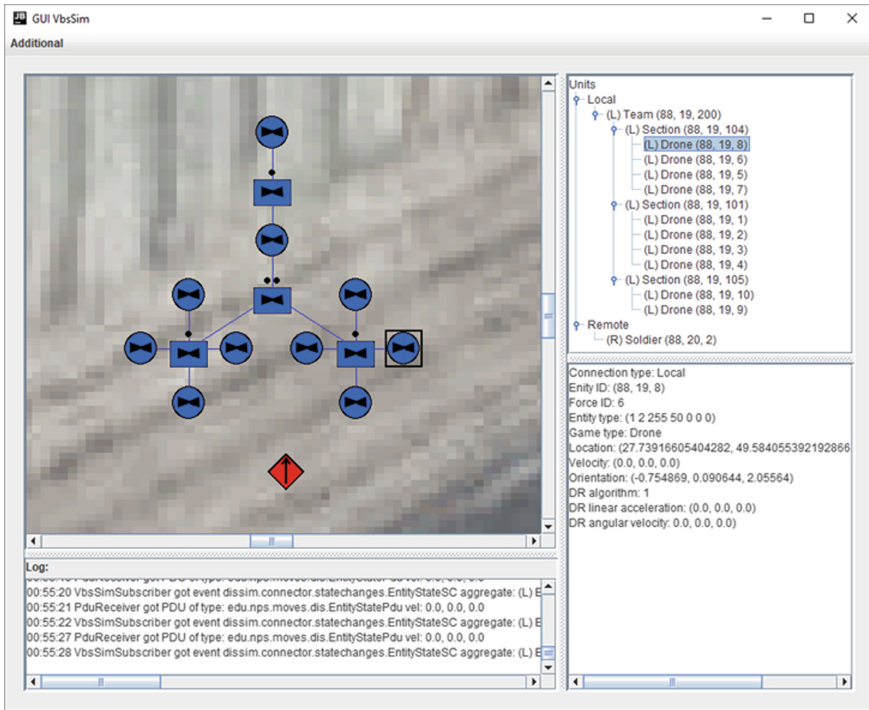


Fig. 2. GUI Layer showing the UAV swarm divided into sections and individual aircrafts

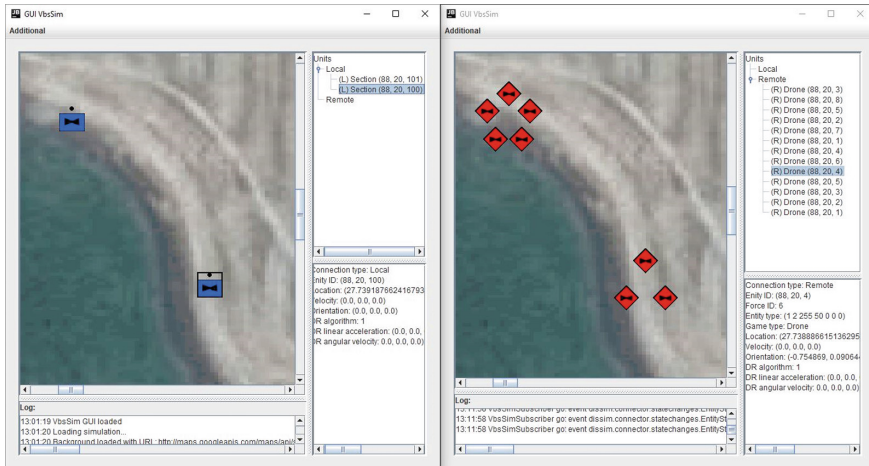


Fig. 3. The process of disaggregation in the aggregation layer – the movement of two sections of the BSP results with movement of individual aircrafts

Each of aggregates from the aggregation layer can be a simulation agent in the agent layer simultaneously. The agent, which is the basis of a multi-agent approach, represents every object participating in the simulation. It has appropriate data structures to store its state and knowledge both about the environment and other agents. In order of using the self-preservation rules and the rules of reacting to external stimuli, there are determined the actions to be taken by the given object. Presented in Fig. 4 reaction to the event of firing is one example of these rules appliance. The multi-agent approach allows for simulation of objects autonomy and modelling the behaviour of individual aggregates from single aircrafts to whole groups. The agent layer interoperates with other simulators by sending events via the event bus (and receiving respectively).

The distribution layer is responsible for the cooperation of the presented environment with other simulators using distributed simulation protocols. These can be simulators controlling different scopes of simulation, however working in one simulation experiment. The most commonly used distributed simulation protocols are *Distributed Interactive Simulation* (DIS) and *High Level Architecture* (HLA)⁵. The main difference between them is the way of exchanging messages: in the case of DIS – directly between simulators using structured PDUs (Packet Data Unit), in the case of HLA – using a central component called RTI (run-time infrastructure), which manages their transmission only to specific simulators.



Fig. 4. Reaction to simulation events in the agent layer – group of 5 UAV is fling in compact order, after firing the aircrafts are being dispersed.

Distributed simulation gives possibility to model UAV behaviours in a wide range of scenarios. Moreover, the approach avoids the necessity of implementation full behavioural models of objects which the UAVs would interact. In the tests of environment, the Virtual Battlespace 3 system (VBS3) has been used for this purpose. Due to the possibility of performing high-resolution virtual simulation characterized by high quality of graphical imaging, VBS3 allows visualization of the UAV group cooperation model. On the other hand, advanced simulation algorithms and interactivity allows to perform UAV interactions with other objects [14]. However, this would not be possible without the

⁵ Commonly used distributed simulation protocols: DIS (35%), HLA (35%), TENA (16%), CTIA (3%), other (7%) [4].

software encoders and decoders located inside the distribution layer of the environment. At first, events sent from the agent layer are received from the event bus, and then the encoders are responsible for sending them to another simulator in the format corresponding to the chosen protocol - the decoders are responsible for the reverse process. If the DIS protocol is used, the layer is responsible for sending the PDU to the IP addresses specified in the options and receiving incoming packets to the port which the listener is being set on. In the case of the HLA protocol, the layer is responsible for connection with RTI, as well as for subscribing and publishing selected methods and objects of the federation in accordance with the FOM model. In the presented solution, to ensure compatibility with the DIS protocol, we have used FOM model RPR 2.0. Figure 5 shows the state of two cooperating simulators: VBS3 and DisSim working in a common simulation experiment. The right side of the graphic presents the section of five objects held locally in DisSim (blue colour) and two objects from another simulator (red colour). On the left is VBS3 system with the same objects - five UAVs are in the air, while the two UAVs maintained by the VBS3 simulation are on the surface.



Fig. 5. The state of two simulators cooperating with the use of the distribution layer (Color figure online)

2.4 Particular Scenarios Implementation

The DisSim package prepared for performing a distributed event-driven simulation with multiresolution and multiagent modelling approach results with possibility of implementing appropriate scenarios in order to conduct research on UAS. Such a scenario requires: the preparation of objects extending *SimObject* class, defining aggregation rules by implementing the interfaces *IAggregate* and *IAggregateRule*, using encoders and decoders for the proper PDUs (in the case of DIS Protocol) or interaction and objects handles (in the case of the HLA protocol). It is also necessary to extend the abstract classes for the subscriber and event broker as well as the abstract *EntityManager* class by mapping the prepared *SimObject* classes to the *EntityType* attribute compliant with the DIS protocol. Section 3 describes the test scenarios implemented in simulation environment we have prepared in such a way. The presented research problem is served as a case study in order to present functionality of the concept. A class diagram of prepared scenarios is presented in Fig. 6.

The presented distributed discrete event-driven simulation environment provides agent-based modelling of UAVs activity as well as modelling at different resolution levels. It is also prepared to achieve interoperability with the VBS3 simulator. In order to demonstrate the solutions ability to carry out researches of UAVs movement and swarming models and techniques, the algorithms solving search problem variant was implemented. However, It should be emphasized that presented algorithms are for demonstration purpose only and they are not compared with listed before solutions from literature. The problem in this case was formulated as follows:

In the given area there are K sources of terrestrial signal. The strength of the signal decreases with the distance from source d according to the function $G(d)$. Using a group of L unmanned aircraft vehicles, find as many signal sources as possible in the shortest possible time. Sensitivity of the aircraft sensors allows the signal to be detected at a distance c from it, however, the moment the source is detected is to be at this distance c directly from the source. Aircrafts start a flight from one point in space with a specific initial speed and the maximum flight speed is V_{max} . The duration of search should last no more than maximum exploration time T_{max} . The problem of aircrafts collisions is negligible.

As part of the above problem, a multicriteria optimization task was formulated. It is based on two criteria: the number of found sources Z and the exploration time T . The first criterion is maximized and the second one is minimized. To determine one solution in the objective function a weighted sum method with weights of 0.5 for each of two summands is used. The first summand is the quotient of the number of found sources Z and number of signal sources K . The second summand is the ratio of the time remaining to maximum exploration time, which is $T_{max} - T$, and maximum exploration time T_{max} . In this manner the objective function should be maximized in the value range $[0, 1]$. It should be emphasized that the signal source is found when the UAV is located at a distance d not larger than the range of the aircraft sensors c rather than when the sensors detect only the signal itself in the area defined by the maximum range of the signal source f (assumed that $c < f$). It means that the maximum value of the function determining the signal strength $G(d)$ is within the range of the aircraft sensors.

In order to solve the problem, three algorithms has been implemented:

- Algorithm I – exact algorithm,
- Algorithm II – PSO algorithm with maximum speed and search area restrictions,
- Algorithm III – PSO algorithm with maximum speed and search area restrictions modified with maximal remembered signal value resetting for given aircraft and for all aircrafts in case of source being detected.

Algorithm I – Exact Algorithm. The exact algorithm is a full overview of the search area by all aircraft. The area is divided into the number of aircraft. Each area segment is checked by one aircraft to check the signal level entirely. Algorithm I has been illustrated in Fig. 7.

c_1, c_2, c_3 – coefficients defining the influence of individual summands, respectively, the weight of inertia, cognitive parameter and collective (social) parameter,
 r_1, r_2, r_3 – random values from a uniform distribution $U(0, 1)$.
 $\|vector\|$ – norm of the vector.

Assessment the aircraft position in the simulation step is conducted with usage of the signal strength function $G(d)$ according to the distance from the source. The function has a great influence on formula for the new velocity of the i -th agent in the next step $t + 1$. Signal strength function is used to determine the value of $pbest$ (the highest value of $G(d)$ found by the i -th aircraft and its best-known position) and value of $gbest$ (the highest value of the $G(d)$ function found by a set of neighbours and the best position found by them). Moreover, it determines when a given signal source is considered as found (function value equal to 1). Function $G(d)$ is formulated as follows:

$$G(d) = \begin{cases} 1 & , d \leq c \\ 0 & , d > f + c \\ 1 - \frac{d-c}{f} & , c < d \leq f + c \end{cases} \quad (6)$$

d – aircraft distance from the signal source,
 c – aircraft sensors range,
 f – maximum source signal range.

Figure 8 presents an interpretation of the function $G(d)$ conditions:

- (1) Function value is equal to 1, when the source of the signal S is directly in the range of the sensors of the aircraft D (source is found);
- (2) Function value is equal to 0, when the range of the aircraft sensors D is beyond the coverage area of the S signal source;
- (3) Function values are in the range (0, 1), when the range of sensors of the aircraft D is in the coverage area of the signal source S , but not directly over this source.

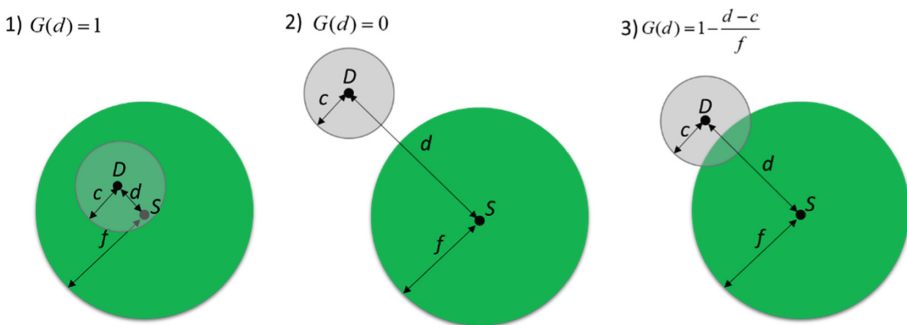


Fig. 8. Interpretation of the function $G(d)$ conditions for PSO algorithms

Algorithm III – Modified PSO Algorithm with Maximal Signal Value Resetting.

As a third solution, the PSO algorithm based on the Algorithm II modifications was reused. Modification includes limitations on the maximum velocity and the search area, moreover additionally variable for each signal source determining whether it has already been found has been applied. This variable determines new conditions, namely, in each iteration:

- if the new signal source was found by any aircraft, the value of g_{best} and its location is zeroed for each aircraft,
- if any aircraft is above the already found source, the value of p_{best} and its location is zeroed only for that aircraft.

Owing to such a solution, if any radar is found, the UAV swarm focuses on finding another one. Figure 9 presents the environment during experiments for a scenario implementing the modified PSO algorithm.

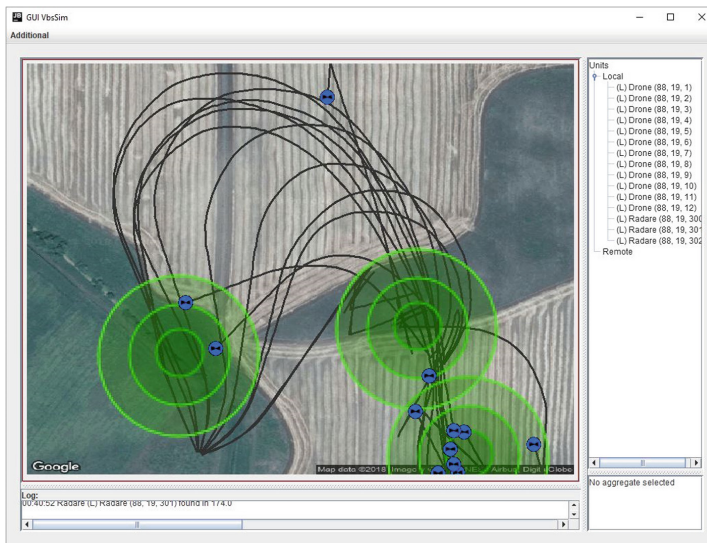


Fig. 9. Twelve aircrafts and three signal sources during experiments for a scenario implementing the modified PSO algorithm.

Algorithms Comparison. In order to compare quality of algorithms, basing on established model parameters, the criteria values and the value of the target function for the experiments were calculated. Table 1 presents the results obtained in a series of 10 experiments. It can be noticed that the heuristic algorithm II does not show better results than the exact Algorithm I, what could have been expected. However, the applying of further modifications to the heuristic algorithm has conducted in Algorithm III to best results. Nevertheless, it should be noted that such experiments ought to be carried out for a much larger number of parameters in order to be able to draw more

precise conclusions. Taking into account the results of the PSO algorithm, once in the case of applying only limitations to the search area and maximum speed, and the second time with the above restrictions by adding resetting the values of the local target functions after finding the signal source, the heuristic algorithm have to be well adjusted to the problem model. Otherwise, it can product much worse results than the exact algorithm.

Table 1. Comparison of algorithm quality for the following model parameters: 10 measurements, 4 signal sources, 15 UAVs, maximum exploration time 1000 s, signal range 80 m, range of aircraft sensors 5 m, maximum speed 8 m/s, search area 400 × 600 m, PSO coefficient weight: $c_1 = 10$, $c_2 = 4$, $c_3 = 1$.

	Algorithm I	Algorithm II	Algorithm III
Found signal sources average	4,00	3,80	4,00
Found signal sources standard deviation	0,00	0,42	0,00
Exploration time average	455,98	534,70	204,80
Exploration time standard deviation	136,33	290,80	79,71
Optimisation function average	0,77	0,71	0,90
Optimisation function standard deviation	0,07	0,19	0,04

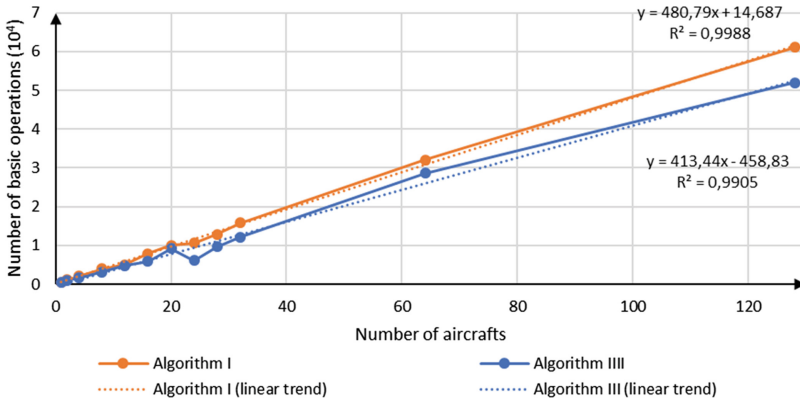


Fig. 10. Graph of the computational complexity of Algorithm I (exact) and Algorithm III (with maximal signal value resetting) at a constant number of signal sources equal to 4 and increasing number of aircrafts.

During the examination of the presented algorithms in the simulation environment, in addition to the comparison of the quality of the algorithms, the computational and memory complexity has been also analysed. Complexity has been calculated in depend of increasing number of UAV, signal sources or maximum exploration time. Considering the concept of iterative calculation of subsequent aircraft positions, the differences in complexity (in general) are not significant and relative to certain dependencies

asymptotically aspire to the linear functions such as shown in Fig. 10. However, in each case, biologically inspired Algorithm III has showed better coefficients of these functions than Algorithm I.

4 Conclusion

In the paper the simulation environment was presented as a platform for UAS testing. Moreover, it can be used to study autonomous vehicles, boats (including submarines) or other robots used in an automation of production processes in factories or even processes in agriculture. The environment is a practical consequence of both agent-based modelling of UAV behaviours and modelling them at diverse resolution levels. It also managed to achieve interoperability with the VBS3 simulator with support of distributed simulation protocols. The proposed solution has been used to conduct research on the use of the UAV group to search for sources of terrestrial signal in a given area. The described problem was a case study in order to present the functionality and usefulness of environment. We defined a multi-criteria optimization task based on two criteria: number of found sources, time of exploration. Thereafter, there were proposed three algorithms to solve the problem: one exact algorithm and two heuristic algorithms basing on the PSO algorithm. In order the algorithms were used for demonstration purposes, there were not compared to other solutions. Considering the results of experiments, Algorithm III (involving maximum speed and search area restrictions and modified with maximal remembered signal value resetting for given aircraft and for all aircrafts in case of source being detected) seems to be the best. In case of this algorithm, the lowest value of the average exploration time were obtained as well as the highest value of the optimization function. Moreover, each of experiments ended with finding all signal sources.

The present state of work is actually a promising beginning to further wider research. The presented results show the usability and functionality of the prepared environment for simulation of UAS and problem solving. Further steps and directions of development envisage extension of models and methods (both for the problem presented and for the new problems), strike a balance with maintaining the autonomy of individual objects, and controlling and hierarchizing a group of objects. As part of considering centralized and hierarchical solutions, it should be possible to define a partially determined model (e.g. specifying only the preferred number of aircrafts in the sections) or strongly determined (e.g. a strict determination of the UAVs sections as a part of larger group with provided autonomy of individuals aircrafts or with autonomy of sections with strict determination of position arrangement of UAVs within this section) [13]. In addition, it also seems important to take into account collisions between aircrafts as well as between aircrafts and other objects, so as to explore more heuristic algorithms for UAS appliance (mainly in the field of biologically inspired algorithms). These algorithms shall be examined in a larger range of test cases and parameters. Especially, the interactions with external objects, e.g. air defense measures maintained in VBS3 simulation, should be provided.

References

1. Davis, P.K., Bigelow, J.H.: Experiment in Multiresolution Modeling (MRM). RAND (1998)
2. Dyk, M., Najgebauer, A., Pierzchała, D.: SenseSim: an agent-based and discrete event simulator for wireless sensor networks and the internet of things. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT) Proceedings (2015). ISBN 9781509003655
3. Eberhart, R., Kennedy, K.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995, Nagoya, pp. 39–43 (1995)
4. Gustavsson, P.M., Björkman, U., Wemmergard, J.: LVC aspects and integration of live simulation 09F-SIW-090. In: SISO Fall Simulation Interoperability Workshop (2009)
5. Hutchison, M.G.: A method for estimating range requirements of tactical reconnaissance UAVs. In: AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Portsmouth, Virginia, pp. 120–124 (2002). <https://doi.org/10.2514/6.2002-3438>
6. Kückelhaus, M.: Unmanned aerial vehicle in logistics. A DHL perspective on implications and Use cases for the logistics industry, DHL customer solutions & innovation (2014)
7. Munoz, M.F.: Agent-based simulation and analysis of a defensive UAV swarm against an enemy UAV swarm, Naval Postgraduate School, Monterey, California (2011)
8. Najgebauer, A., et al.: The qualitative and quantitative support method for capability based planning of armed forces development. In: Nguyen, N.T., Trawiński, B., Kosala, R. (eds.) ACIIDS 2015. LNCS (LNAI), vol. 9012, pp. 212–223. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15705-4_21
9. Office of the Secretary of Defense, Unmanned Aircraft Systems Roadmap: 2005–2030, Technical report, Department of Defense (2005)
10. Ousingawat, J., Campbell, M.E.: Establishing trajectories for multi-vehicle reconnaissance. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island (2004)
11. Pierzchała, D., Skrzypecki, S.: Multi-agent and multi-resolution distributed simulation DisSim – VBS, Simulation in research and development (in Polish: Symulacja w badaniach i rozwoju), vol. 7 no. 1-2/2016, pp. 25–34 (2016)
12. Ryan, J.L., Bailey, T.G., Moore, J.T., Carlton, W.B.: Reactive tabu search in unmanned aerial reconnaissance simulations. In: Proceedings of the 1998 Winter Simulation Conference, Grand Hyatt, Washington D.C., pp. 873–879 (1998)
13. Tarapata, Z.: Movement simulation and management of cooperating objects in CGF systems: a case study. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010. LNCS (LNAI), vol. 6070, pp. 293–304. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13480-7_31
14. Wantoch-Rekowski, R. (Red.): Technologies for designing live and virtual simulators in a programmable virtual simulation environment VBS3 (in Polish: Technologie projektowania trenerów i symulatorów w programowalnym środowisku symulacji wirtualnej VBS3). Wydawnictwo Naukowe PWN, Warszawa (2016)
15. Xie, R., Wang, X., Wang, X., Wei, H.: UAV flight performance optimization based on improved particle swarm algorithm. In: Su, C.-Y., Rakheja, S., Liu, H. (eds.) ICIRA 2012. LNCS (LNAI), vol. 7506, pp. 396–403. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33509-9_39
16. Aerial Surveillance & Monitoring, observation with UAV/drone. <https://www.microdrones.com/en/industry-experts/security/monitoring/>

17. Department of Defense Announces Successful Micro-Drone Demonstration > U.S. DEPARTMENT OF DEFENSE > News Release View. <https://www.defense.gov/News/News-Releases/News-Release-View/Article/1044811/departement-of-defense-announces-successful-micro-drone-demonstration/>
18. DroneBase - Drones and the Future of Entertainment. <https://blog.dronebase.com/2017/07/10/drones-and-the-future-of-entertainment/>
19. ISR Unmanned Aerial Vehicles: Intelligence, Surveillance, Reconnaissance (ISR). <https://www.militaryfactory.com/aircraft/uav-intelligence-surveillance-reconnaissance.asp>
20. LOCUST: Autonomous, Swarming UAVs Fly into the Future. http://www.navy.mil/submit/display.asp?story_id=86558
21. Project Perdix | Beaver Works. <https://beaverworks.ll.mit.edu/CMS/bw/projectperdixcapstone/>
22. Unmanned Aerial Systems (UAS) - SKYbrary Aviation Safety. [https://www.skybrary.aero/index.php/Unmanned_Aerial_Systems_\(UAS\)](https://www.skybrary.aero/index.php/Unmanned_Aerial_Systems_(UAS))