Software package for simulating a group of unmanned aerial vehicles performance

R V Kim, D V Frolova and A S Borovkov
Moscow Aviation Institute (National Research University), 4, Volokolamskoe shosse, Moscow, 125993, Russia
E-mail: kimrv@mai.ru

Abstract. The paper presents the architecture of a software, designed to simulate the functioning of a group of unmanned aerial vehicles. The software consists of two components: software for modeling unmanned aerial vehicles functioning processes, which designed to simulate the controlled movement of a single UAV and to simulate the functioning of its onboard systems and software for the ground control post, which designed to form a flight task, generate and send control commands to the UAV, and receive information from the UAV about its current state. The paper also discusses the unmanned aerial vehicle group application scenario, main functions of each software components, its user interface as well as implemented models and algorithms.

1. Introduction
The tasks of unmanned aerial vehicles (UAVs) group control are significantly different from the tasks of simulating UAVs in their single application. In addition to the tasks like finding the optimal route and control of UAVs on its trajectory, traditional for simulating single UAV, when using UAVs in a group, it is necessary to consider different strategies for using the group, solve the problems of target allocation, UAVs’ control when moving in a group, and design models of information exchange between UAVs, as well as between UAVs and a ground control post (GCP). Thus, the development of unmanned aerial systems (UAS), assuming scenarios for UAVs group use, the emphasis shifts from the single UAV controlling tasks towards tasks of a higher level, typical for group use. Mathematical modeling is an effective tool widely used in the development of complex technical objects, including UAS, allowing to reduce development costs and development time. The development of software systems that allow to simulate the processes of UAVs functioning in their single and group use will ensure an increase in the efficiency of UAS development in general.

2. Typical scenario for UAVs group use
Group control of UAVs can be carried out in accordance with various strategies, the main of which are: a centralized strategy based either on the principle of one-man management, or on the principle of hierarchical control [1–3], and a decentralized strategy [4], a feature of which is the absence of a central control unit, and independent decision-making regarding their actions by each group member. Today, there are various methods of group control within each of these strategies. The simplest to implement and theoretically well-developed methods are the methods of programmed control of UAVs, assuming their movement along pre-planned routes.
In the current version of the software package, exactly this approach to the organization of group management is implemented, since it allows to focus on basic architectural solutions, without concentrating on the complex control algorithms development.

Taking into account the above, the following scenario of group UAVs use is implemented in the software package. It is assumed that during the pre-flight planning stage:

- the number of UAVs available to form a group is fixed and known in advance;
- all UAVs forming a group are of the same type with the same technical characteristics and cost;
- the number of targets is fixed and known in advance;
- coordinates of all target objects are known exactly;
- in the area of the targeted use of UAVs, there may be zones that are closed for flight (overflight zones) due to meteorological conditions or terrain features, the location of which is known exactly.

![Figure 1. Scenario for UAVs group use.](image)

As a result of pre-flight planning flight tasks are formed for each UAV in the group (figure 1). The fulfillment of the tasks provides a reaching the zones where target objects located, taking into account the overflight zones bypass requirement. The return of the UAVs to the base points is not a mandatory requirement, therefore, it is assumed that upon return, the UAVs can follow through the overflight zones. The flight task for each UAV is formed as a set of route points with known coordinates. It is also assumed that UAVs fly at a constant specified altitude, determined at the pre-flight planning stage. A set of routes with the minimum sum of lengths is accepted as the optimal set for the group to reach target objects, and ensures movement to the points of the intended location of the target object in the minimum total time. It is also assumed that each UAV is equipped with an autopilot that ensures its movement along a given route and reaching the target object location point.
3. **Software architecture**

A software architecture is based on presupposes its arrangement of two separate independent components corresponding to individual UAS elements. In accordance with the proposed architecture, the software package includes (figure 2):

- Software for modeling the processes of UAV functioning (UAV software). This software is designed to simulate the controlled movement of a single UAV, as well as to simulate the functioning of its onboard systems.
- Software for simulating the operation of the ground control post (GCP software). This software is designed to form a flight task, generate and send control commands to the UAV, and receive information from the UAV about its current state.

The interaction of the software components is carried out via the TCP/IP network protocol using the upper level MAVLink protocol (Micro Air Vehicle Link) [5].

![Software package architecture.](image)

**Figure 2.** Software package architecture.

As a basis for the UAV software, the open source software system ArduPilot [6] is used, which implements a wide range of functions that provide an opportunity to develop, debug and test control algorithms for various types of UAV and their subsequent loading into the memory of real UAV control systems. One of the significant advantages of the ArduPilot is that it allows...
simulation of the processes of UAV operation and its systems using mathematical models of UAV. At the same time, it is possible to integrate your own mathematical models of the environment, UAV dynamics, navigation devices models and a control system models and algorithms into the simulation circuit. Thus, ArduPilot provides a complete software environment for the development and testing of navigation and control algorithms for UAVs. At the same time, the use of third-party software, such as ArduPilot, to simulate the movement of individual UAVs in a group, allows you to focus more on group control tasks.

GCP software is implemented as a separate application, the main functions of which are:

- forming a simulation scenario, including the calculation of the optimal routes of UAVs included in the group;
- starting the required number of copies of UAV software (ArduPilot);
- sending information about the route to each of the copies of the UAV software;
- receiving information about each of the simulated UAVs state;
- displaying simulation results to the user and saving the results to the data storage.

The formation of the simulation scenario is carried out by the user by setting the parameters of the UAV group (the number of UAVs, the type of UAVs, the initial location of each UAV), the coordinates of the target objects and overflight zones. Based on this information, the optimal routes of each UAV of the group are calculated.

Before starting the simulation the GCP software starts the required number of copies of the UAV software, each of which simulates the flight of single UAV. Also at this stage, each of the UAV software copies opens the required number of network ports through which interaction with the GCP software is carried out. For each instance of the UAV software, at least two ports are opened, the first of which is intended for sending commands from the GCP software to the UAV software, the second for receiving information about the UAV’s state in the opposite direction.

Each of the instances of the UAV software interacts with the GCP software through the available ports via the TCP/IP protocol using the upper level MAVLink protocol. The MAVLink protocol defines the message formats for the exchange both between the GCP and the UAV, and between the UAVs. The messages themselves are text files in XML format. In the current version of the software such messages contains the information about flight tasks (coordinates of intermediate points of the route) which is sent from the GCP to the UAV, and telemetry (parameters of the UAV state) which is sent from the UAV to the GCP. Based on telemetry received from the UAV software, the GCP software user interface displays charts of UAVs trajectory parameters evolution. These parameters are also saved to .csv and .xml files.

The GCP software is based on the open source Dronekit library [7], implemented in the Python programming language and containing a set of classes that provide interaction with the ArduPilot software system.

4. Models and algorithms

ArduPilot includes its own open source simulator that allows to supplement and modify existing mathematical models. Within the framework of the ArduPilot simulator, the developed models of the external environment, models of spatial controlled movement of UAVs, and models of typical UAS elements were implemented.

The ArduPilot software package is implemented in C++ using an object-oriented approach. The main components are:

- UAV dynamics model (aircraft and multi-rotor types);
- models of the external environment;
- model of the navigation system;
- model of the control system.
The object-oriented architecture is based on the Singleton pattern [8], which ensures the interaction of an instance of software for modeling the processes of UAV operation with other software, guarantees that a single instance of a UAV will exist in a single-threaded application of modeling and provides a global access point to this instance.

As mentioned above, the value of the total route length of all UAVs is used as an optimality criterion for solving the target allocation task. At the same time, the shortest possible route is used as the length of the UAV route from the launch point to the designated target, taking into account the requirement for the need to fly around the overflight zones. In the current version of the software package, the target allocation task is solved by enumerating all possible sets of shortest routes of individual UAVs to their assigned targets. At the same time, since the return of the UAV to the basing point is not a mandatory requirement, when solving the task of target allocation and constructing optimal routes, only the route lengths from the basing point to the allocated target location point are taken into account. The initial data for solving the problem are the target objects location coordinates, the UAV basing points coordinates and the coordinates of the overflight zones. It is assumed that the overflight zone, regardless of its geometric parameters, can be inscribed in a rectangle with sides parallel to the north and east directions. Taking into account this assumption, in the current version of the software package, the overflight zones are described in the form of quadrangles set by the coordinates (longitude and latitude) of their peaks.

The problem of calculating the optimal (shortest) route of a single UAV from the basing point to the target object location point is solved in the GCP software using the A* (AStar) algorithm. Note that due to the peculiarity of the A* algorithm, which assumes the movement of the UAV through the centers of the regions into which the space is divided, the calculated routes are quasi-optimal in general.

5. User Interface
The user interface of the GCP software is implemented on the basis of the Tkinter library and is designed to set initial data (figure 3) and schematically display UAVs routes in the horizontal plane (figure 4).

```
Figure 3. GCP software User Interface.
```

The “UAVs” tab is designed to set the coordinates of the UAVs basic location points. Tab “Targets” is intended for setting coordinates of target objects location points. The “Zones” tab
is used to set the coordinates of the overflight zones. The “Simulation settings” tab is used to set the settings for the simulation process and control the simulation process (start and stop simulation).

6. Conclusion
The architecture of the software package for modeling the functioning of a UAV group considered in the paper presupposes the implementation of the software package in the form of two separate interacting components: software for modeling the processes of UAV functioning; software for modeling the operation of a ground control post. The main advantages of the proposed architecture are the possibility of independent revision of individual components, as well as the replacement of software components with hardware—real UAVs, their components and ground control stations, which provides a low-cost transition from simulation to semi-natural simulation and testing of real UAVs.

References
[1] Murphey R A 2000 An approximate algorithm for a weapon target assignment stochastic program (Boston: Springer Science+Business Media) pp 406–421
[2] Puterman M L 2005 Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Statistics (New York: John Wiley & Sons Inc)
[3] Ross M S 1983 Introduction to Stochastic Dynamic Programming (New York: Academic Press)
[4] Rasmussen S J, Chandler P, and Schumacher C 2002 Investigation of Single vs Multiple Task Tour Assignments for UAV Cooperative Control AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA 2002 (Monterey, California)
[5] MAVLink Developer Guide URL https://mavlink.io/en/
[6] ArduPilot Documentation URL https://ardupilot.org/ardupilot/
[7] Dronekit URL https://dronekit.io/
[8] Larman C 2004 Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (Boston: Addison Wesley Professional)