Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Simulation of the algorithm of group interaction of cyber-physical objects in 3D Space

Simulation of the algorithm of group interaction of cyber-physical objects in 3D Space

S Palavenia, Yu Duinova, M Popova and Imad El Abed

Belarusian State Academy of Telecommunications, ul. F.Skoryny 8/2, Minsk, 220114, Belarus

E-mail: s.polovenia@gmail.com

Abstract. This article discusses the possible types of group management strategies. The rationale for choosing a specific group management strategy is given. This article also provides an algorithm for group movement. Then the phases and results of its simulation will be described. Many models can be built for the same system. These models will differ in the degree of detail and account for certain features and modes of operation of the real object, reflect a certain facet of the essence of the system, focus on the study of a certain property or group of properties of the system. Therefore, it is important to state clearly the purpose of modeling at the initial stage of model construction. It is also taken into account that the model is built to solve a specific research problem. The experience of creating universal models has not justified itself due to the bulkiness of the created models and their unsuitability for practical use. To solve each specific problem, you need to have a model that reflects the most important aspects and connections from the point of view of the study. The importance of a specific task of modeling goals is also dictated by the fact that all subsequent stages of modeling are carried out with a focus on a specific goal of the study.

1. Description of the group interaction algorithm in 3D space

1.1 Description of group interaction

*1.1.1. The advantages of using robot groups***.** This is a greater range, achieved due to the robots dispersal throughout the working area, and an expanded set of functions performed, achieved by installing individual devices on each robot, and, finally, a higher probability of completing the task, achieved due to the possibility of redistribution of tasks between the robots of the group in case of failure of some its member. Therefore, such complex tasks of special robotics as large-scale research and sensing the surface of other planets, assembling complex structures in space and under water, participating in combat, rescue and support operations, etc., can effectively be solved by robots only with their group interaction. At the same time, new problems of group management and communication related to the organization of group interaction of robots arise [1].

Along with the traditional problems of robotics in the using group of robots, a number of new problems arise, and above all this is the problem of organizing the interaction of robots in groups when solving a complex task and the problem of communication related to the organization of interaction of robots.

The first task of group management is to determine the composition and structure of the group, necessary or even the best, to accomplish the stated functional task. If the composition is already set, then remains the task of determining the structure of the group, and also operational reconfiguration, when individual group members fall out or new ones appear.

The second task is to create a self-governing group. Such groups can be localized on a single robot or distributed over a group, i.e., be centralized or decentralized. For a heterogeneous group, it is advisable to allocate one robot (often with the most powerful computing resources) for the center of this type of control, with the distribution of some common functions on other group robots and the reservation of all important functions on other robots.

In general, not all robots of a group can be used to solve a specific main task, the following subtasks should be solved to accomplish the whole main task of group management:

the formation of the active part of the group - the cluster as a set of robots formed to achieve a particular goal;
optimal distribution of subtasks between the robots of the group, as well as the redistribution of these subtasks when the situation changes;
implementation of the functions of the robots in the cluster.

For practical implementation, the following algorithmic tasks are defined:

- development of group operation logic;
- development of the logic of the functioning of the robot in the group;
- development of hierarchy logic in a group, i.e., with which conditions and with which states the robots will play the role of "leaders" and "subordinates";
- development of communication and interaction algorithms in a group of robots;
- development of group navigation algorithms using information from all navigation devices of robots in the group;
- development of unified on-board software for complex processing of group information.

1.1.2. Description of the selected control strategy. In a centralized control strategy there is a central device which has information about the states of all the objects of the group and their environment. It evaluates the current situation and decides on the future behavior of the objects in the group. The central unit can be located outside the group (for example, on the operator’s control panel) or on one of the group members (centralized control with “the master”).

Centralized strategies have good performance when controlling non-large groups of objects. With an increasing the number of objects, the load on the communication channel and control devices grows. One of the solutions is to implement a hierarchical model, in which a group is divided into subgroups, where each one is controlled by its leader. A group of leaders also has its leader, and so on, depending on the number of hierarchy levels [2].

1.1.3. Types of centralized control. Centralized control of objects can be of two types: single-initial and hierarchical [3].

Single-initial control (the presence in the group of the commander or the central control unit of the objects (CCU), which are entrusted with the task of planning and managing the group). The advantage of single-initial control is the simplicity of its organization and algorithmization. The disadvantages include a long decision time due to the task of optimizing all group members to achieve a group goal.

Hierarchical control (the presence of the control center unit or the commander, who control a small number of objects, each of them has its own group of objects). Comparing with a single-initial control, this type of control significantly reduces the complexity of the task, but the complexity of the management structure can lead to strong delays or failures in the transfer of commands from the upper to the lower level.

1.2 Description of the connection establishment between the subordinate and the leader

1.2.1. The organization of communication between group members. The most popular solutions are based on radio signals (Wi-Fi networks, GPRS, Ad hoc and wireless sensor networks), as well as methods based on sound and IR signals. All these solutions are characterized by encoding information into discrete or analog signals based on a command dictionary, which is predefined and well-known by control system [4].

The main advantages of the radio signal are:

the quality of the connection is almost independent of weather and environmental conditions;
the ability to exchange between devices in any position relative to the source of the signal - there are no "dead zones" associated with full overlap or reflection of the signal.

1.2.2. Algorithm for establishing a connection between the subordinate and the leader. The subordinate-object searches for the leader within the coverage area of its own communication module. When an accessible leader is detected, the subordinate-object initializes the establishment of a connection with it (see figure 1).

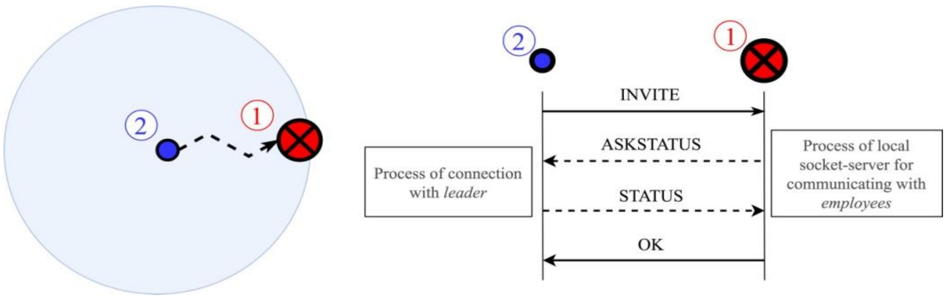

Figure 1. The process of initializing the connection between the subordinate and the leader: 1– leader, 2 - subordinate.

The first step of establishing a connection is sending an INVITE message to the leader. If a connection establishment is possible, the subordinate receives ASKSTATUS message from the leader, which requests information about the status of the subordinate. Next, the subordinate-object sends a message of the STATUS type, which stores information about the status of its movement and its current location. Having received a complete information about the status of the subordinate, the leader includes it in the list of controlled objects and sends him an OK type confirmation, indicating the end of the connection establishment.

1.3 Group movement in 3D space

1.3.1. Tasks of group management. To achieve a specific goal for a group of objects, in the case of a deterministic environment, each object can perform a predetermined sequence of actions [5]. In the case of a non-deterministic dynamic environment, this sequence must be found by the group control system in the process of achieving the goal. Moreover, the actions of the group objects should obviously be coordinated and agreed in a certain way. Thus, the task of managing a group of robots arises. This task is performing the previously found sequence of actions by all objects in the group with the group control system, or in finding such a sequence and its implementation in the process of achieving the set goal. The task of controlling a group of objects whose actions are aimed at achieving a common group goal will be called the task of group management. The essence of the task of group control is to search for and implement such actions of each individual robot group, which would lead to the optimal achievement of a common group goal.

The algorithm of group movement in 3D space. Control of group movement and the development of managing commands for group members is the process of managing the movement of the entire group, which is performed by the leader. This process incrementally calculates the movement of the leader along a given route in 3D space and changing the location of the members from the control group relative to the movement of the leader.

The general scheme of the group movement algorithm is shown in figure 2.

Commands for moving the leader are transmitted directly to his process of managing his own movement in space, and commands to the subordinate with destination points are transferred to the local socket server of the leader, where they are sent to the subordinate-objects in the message format of MOVETO. In a subordinate's program, the process of connection with the leader accepts a move command, and then it is transferred to its own movement control process. Upon reaching the destination specified by the leader, the subordinate sends a STATUS message as confirmation of the performing task, which was received previously. The order of sending the command to the subordinate and receiving the response is shown in **f**igure 3.

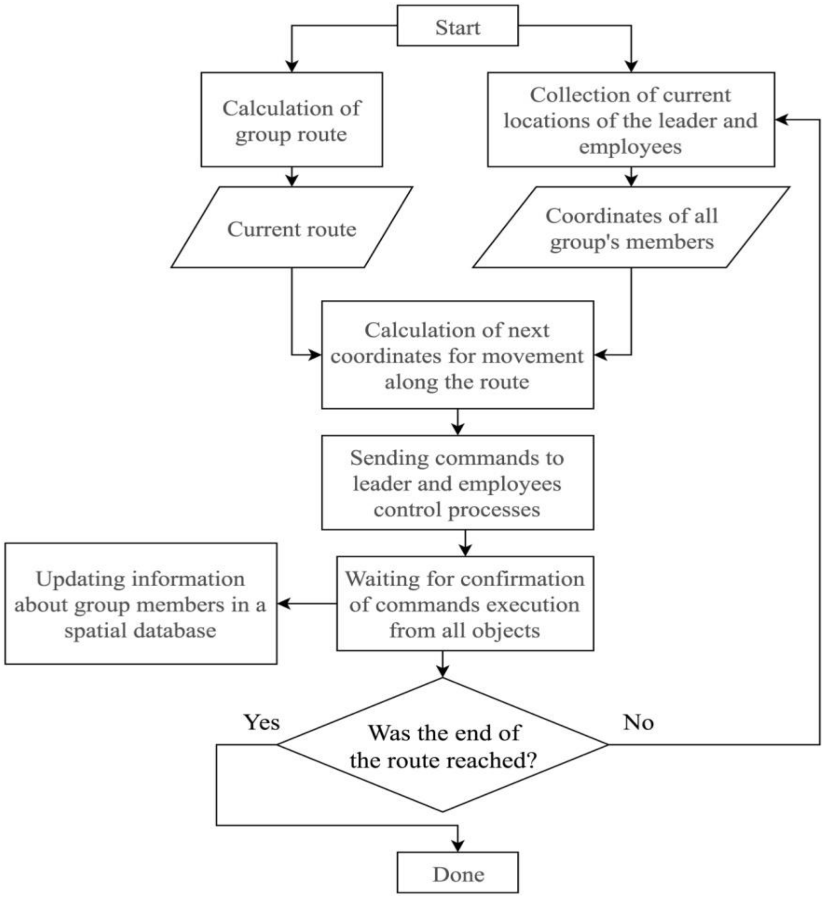

Figure 2. The general scheme of the group movement in 3D space.

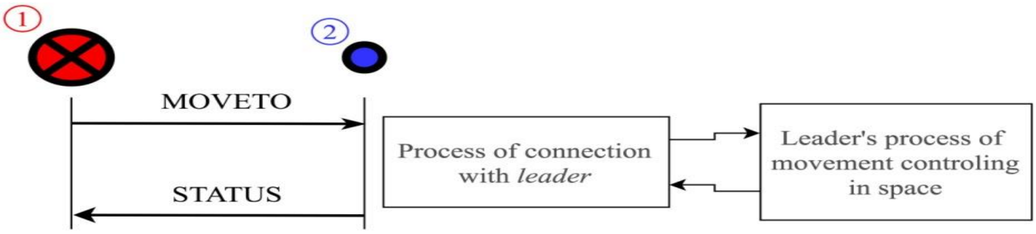

Figure 3. The order of transferring the command to the subordinate and receiving confirmation.

2. Results of group interaction's simulation

2.1 Database Initialization

In order to test the efficiency of the developed group interaction algorithm, we initiate a simulation, during which we will visually demonstrate the work of the algorithm program.

To run the simulation, it is sufficient to have one personal computer. Minimum system requirements are:

operating system (Windows 7+ or UNIX);
interpreter of Python version 3 programming language with installed libraries and modules used in the algorithm;
distribution kit of PostgreSQL DBMS with the PostGIS extension.

The first step is launching the server version of the PostgreSQL DBMS. The structure of the used database is shown in figure 4.

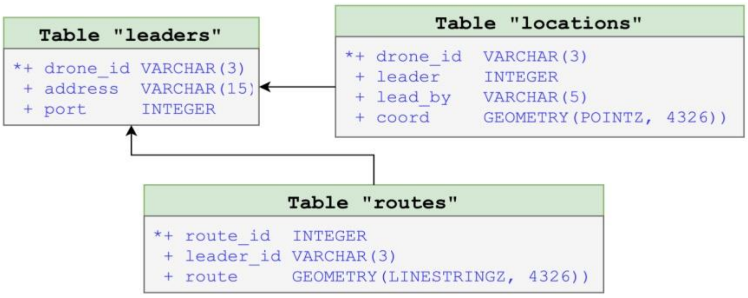

Figure 4. The structure of the used database.

2.2 Launching cyber-physical objects control processes

2.2.1. Starting a leader control process. There are two leaders in this simulation: L0 and L1. When running the leader.py program in the terminal, the attributes are used as main parameters: unique identifier, route number, address and port of the server socket. As additional parameters, you can specify a database other than the default one at startup.

The result of launching the *leader.py* program with its parameters in the terminal is shown in figure 5.

```
Project — Python leader.py -id=L0 -route=0 -ip=127.0.0.1 -port=13254 — 85×10
AlexBook:Project alexei$ python3 leader.py -id=L0 -route=0 -ip=127.0.0.1 -port=13254

05-06-2018 00:03:06,786209 - drone - INFO - Current position: x -1.0, y -1.0, z 0.0.
05-06-2018 00:03:06,786728 - move - DEBUG - Motors starting...
05-06-2018 00:03:09,791225 - move - DEBUG - Taking off...
05-06-2018 00:03:11,793266 - move - DEBUG - On air!
05-06-2018 00:03:11,807013 - server - DEBUG - IP address and port were sent to DB.
05-06-2018 00:03:11,819900 - server - DEBUG - Listening on port 13254 for clients..
05-06-2018 00:03:14,812282 - control - DEBUG - Waiting to start command...
```

Figure 5. Leader-object control program running.

2.2.2. Starting a subordinate control process. In this simulation, there are eight subordinates d0.7. When launched in the terminal employee.py program, the following main parameters are set: unique identifier, range of the communication module, individual movement speed.

The result of the *employee.py* execution for the subordinate with the identifier d0 in the command line is shown in figure 6.

```
Project — Python employee.py -id=d0 — 89×15
AlexBook:Project alexei$ python3 employee.py -id=d0

05-06-2018 00:48:37,380430 - device - DEBUG - Current position: x -0.96, y -0.96, z 0.0
05-06-2018 00:48:37,381135 - move - DEBUG - Motors starting...
05-06-2018 00:48:40,381907 - move - DEBUG - Taking off...
05-06-2018 00:48:42,396958 - move - DEBUG - On air!
05-06-2018 00:48:42,477124 - leader_conn - INFO - Selected as the leader L0.
05-06-2018 00:48:42,507551 - leader_conn - INFO - Explored leader IP 127.0.0.1:13254
05-06-2018 00:48:42,518607 - leader_conn - DEBUG - Trying to connect to the leader L0...
05-06-2018 00:48:43,169377 - leader_conn - DEBUG - INVITE was sent to L0
05-06-2018 00:48:43,203531 - leader_conn - DEBUG - ASKSTATUS was received from leader L0.
05-06-2018 00:48:43,238509 - leader_conn - DEBUG - STATUS was sent to leader L0.
05-06-2018 00:48:43,256019 - leader_conn - DEBUG - OK was received from L0
05-06-2018 00:48:43,267575 - leader_conn - DEBUG - Connected to the leader L0.
```

Figure 6. Subordinate-object control program running.

2.3 Running a simulation of group movement in 3D space

The last step of simulation preparations is sending the *start* command to the leaders, which they are expecting before start moving along the routes given to them. Accordingly, with the *pause* command, the movement of the group would be suspended. Sending commands to leaders is carried out by the program *control.py*. When starting, be sure to specify the control command and the unique identifier of the leader to whom it is sent.

Using *control.py* to send commands to the leaders for their movement start, we will start the simulation of the group interaction algorithm. An example of sending a *start* command to the L0 leader is shown in figure 7.

```
Project — -bash — 87×6
AlexBook:Project alexei$ python3 control.py -id=L0 -command=start

05-06-2018 01:27:29,893087 - console - INFO - Explored L0 leader IP 127.0.0.1:13254.
05-06-2018 01:27:29,893178 - console - DEBUG - Connecting to L0 leader 127.0.0.1:13254.
05-06-2018 01:27:29,936900 - console - DEBUG - Connected to L0 leader 127.0.0.1:13254.
05-06-2018 01:27:29,937236 - console - DEBUG - Sent to L0 leader 'start' command.
```

Figure 7. Sending a moving *start* command to the leader-object.

The program *db_monitoring.py* is used for graphic monitoring of the database in real time. It displays a three-dimensional interactive map with routes and the current location of leaders and subordinates. The interface of this program is shown in figure 8.

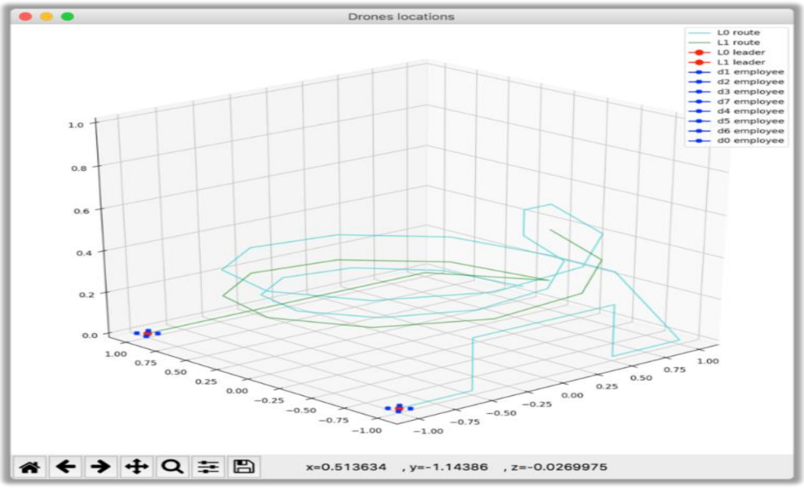

Figure 8. Screenshot of the spatial Database monitoring program interface.

The result of the simulation can be observed in the program window (see figure 8), where real-time changing of the locations of cyber-physical objects groups along the specified routes is shown.

3. Conclusion

Thus, the possible types of group management strategy are presented. The substantiation of the choice of a specific strategy of group management is given. The article also presents an algorithm of group motion. The stages and results of its modeling are also described. The importance of a specific task of modeling goals is also dictated by the fact that all subsequent stages of modeling are carried out with a focus on a specific goal of the study.

References

[1] Vasilyev I A, Polovko S A and Smirnova E U 2013 *Organization of group control of mobile robots for tasks of special robotics. Sci. and techn. lists of SPbGPU. Informatics. Telecomm. Management* **1(164)**

[2] Borovich V S, Gutsul V I and Klestov S A 2018 *Collectives of intelligent robots. Areas of application/under ed. V.I. Syryamkina*

[3] Torgashov L A, Gutsul V I and Romanenko S V 2013 *Development and creation of robotics platform of high cross-country capacity, as an element of emergency response//Journal of Science of Siberia* **9**

[4] Dmitriev S P and oth. 2004 *Information Reliability, Control and Diagnostics of Navigation Systems* (St. Petersburg: Publishing House of the GNC of the RF CRII "Electric Appliance")

[5] Kalyaev I A, Gaiduk A R and Kapustyan S G 2009 *Models and algorithms of collective management in groups of robots* (Moscow: FIZMATLIT)