Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 150 (2019) 687–694

Procedia Computer Science

www.elsevier.com/locate/procedia

13th International Symposium "Intelligent Systems" (INTELS'18)

Methods of a Heterogeneous Multi-agent Robotic System Group Control

O.V. Darintsev[a,b,]*, B.S. Yudintsev[b], A.Yu. Alekseev[b], D.R. Bogdanov[b], A.B. Migranov[b]

[a]*USATU, K. Marxa str., 12, Ufa 450008, Russia*
[b]*Mavlyutov Institute of Mechanics, prospect Oktyabrya, 71, Ufa 450054, Russia*

Abstract

The paper presents the implementation of the high-level decentralized control system for a group of mobile robots (MR), which provides a common information space in a group, and performs decomposition of the main task (task from operator) and generates subgroups on the basis of agents' functional purpose.

Keywords: robotics; robot group; distributed control system; neural network; cloud technology; common information space.

1. Introduction

Controlling distributed autonomous technical objects (agents) is one of the key areas of advanced robotics research, and in the field of control systems in general. To date, one of the most effective approaches to controlling such technical systems is the multi-agent control method [1,2]. This approach is based on the allocation of resources (tasks) between agents of the system based on a comprehensive assessment of the effectiveness of implementing the task by each agent. To evaluate the "effectiveness", different algorithms and criteria can be used depending on the type of distributed system. For homogeneous distributed systems (agents from a system with the same functional purpose), the tasks are distributed using an target function, the values of which are determined by the cumulative

* Corresponding author. Tel./fax: +7-347-235-5255.
E-mail address: ovd@uimech.org

1877-0509 © 2019 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2019.02.032

evaluation of the target functions of each agent. If the values of the objective function change within the specified limits, this ensures effective performance of the overall task (in accordance with certain criteria) [1,2].

To distribute tasks in heterogeneous systems, two main methods are used:

- The “plan improvement method” [3] distributes tasks based on an integrated assessment of the effectiveness of each agent's performance of the assigned task and an assessment of the “security” of the goal (i. e., the number of agents necessary to achieve the goal).
- The “auction” [4] distributes tasks according to the rules of the “market economy”, where each agent determines (based on the selected efficiency criteria) the “price” for accomplishing the task and, accordingly, agents with the “cheapest" offer form a subgroup for the task.

The analysis of articles on this subject indicates that to apply these approaches in a decentralized control system (CS) for a group of autonomous agents, it is necessary to solve several problems associated with the formation of the sub-tasks according to the agents’ functional purpose, and corresponding subgroups to implement them.

2. The problem of control groups of robots with different functional purposes

Suppose there are several autonomous mobile robots (agents) with different functional purposes (reconnaissance, combat, rescue, etc.). It is required to ensure the solution of the main task set by the operator, effectively using the functional capabilities of the group members. The effectiveness criterion is the MR’s “level of competence” – an integral assessment of the most significant properties of the agent, taking into account its current tasks and energy resources.

To solve this problem, it is necessary to fulfill the following conditions when synthesizing the control system of a robotic system consisting of robots with different functional purposes (heterogeneous multi-agent robotic system):

- Providing a common information space in the group, where the objects included in the group not only have the opportunity to exchange information with each other, but also “know” about the aims and tasks of the group and have the information about its members.
- Ensuring the effective access to the group’s common information space, i. e. the possibility of a simple interaction of an operator (for example, through a terminal or an application with a graphical interface) with a group: setting up a main task, obtaining the results of the task, and obtaining the information about the progress of the task in real time.
- The following elements of self-control at the level of interaction between robot agents: algorithms of group interaction in the process of executing the main task (i. e. the algorithms for synthesis of the solution of the problem), decomposition of the main task, distribution of the subtasks.

The fulfillment of these conditions ensures the possibility of effective application of the algorithms of group control of the robotic system.

3. Synthesis of the mobile robots with various functional purposes control system

3.1. Providing a common information space

One of the actual approaches for building such a robotic system is the implementation of the system within the framework of the "cloud" architecture [5]. This model of interaction assumes that the entire group of mobile robots represents a distributed relational database (DB), and each agent contains only a part of the information ((the database row) Fig. 1.) This row contains the information about the current state of an agent (current tasks, charge level, etc.), as well as the information about its functional purpose (using digital code of onboard equipment).

Also, each agent has its own model of behavior patterns, which contains a set of typical algorithms (“models of behavior”) that enable robots to perform tasks (or parts of a main task) within their functional purpose, for example:

- The behavior models database of an intelligence mobile robot contains a corresponding algorithm for planning trajectories for passing through a section of the terrain and effectively exploring the territory with sensors and conducting cartography.
- The behavior models database of a mobile communication robot contains an algorithm for locating a position on the ground to provide the maximum signal level, etc.

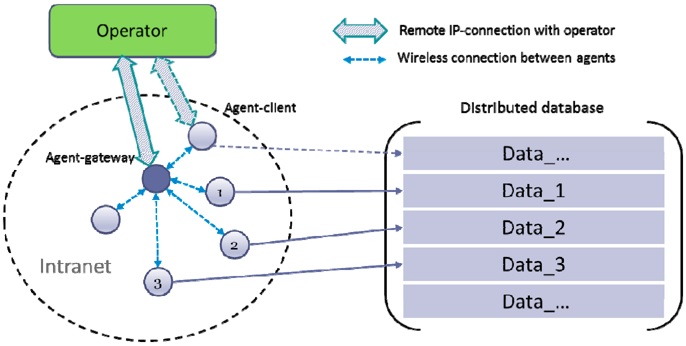

Fig. 1. Cloud model of interaction for a group of mobile robots.

At the same time, as mentioned above, each member of a group can obtain information about the functional purpose of another participant of this group from the distributed database, and thus, about its level of "competence" for solving a particular problem.

In the process of performing a collective task, group members actively exchange information about their status (for example, current coordinates, charge level, etc.) using IP sessions (IP SOCKET) via wireless communication channels. Each agent is capable of servicing 6 IP sessions simultaneously, using a SOCKET-server in multithreaded data processing mode. Thus, a group creates an internal information network (intranet), or, in other words, a common information space.

This organization of information exchange in a group ensures the fulfillment of the five basic conditions that are necessary for building an effective control system for a distributed robotic system [5]:

- the speed of connection initialization by agents, which is sufficient to obtain real-time data;
- security of data transmission;
- the integrity of data transmission;
- adaptability, i. e. the possibility of information exchange between agents in conditions when the group is distributed unevenly, and the form of distribution of agents in the workspace changes dynamically;
- energy efficiency of software and hardware implementation.

3.2. Providing access to the common information space

If a member of a group has the appropriate means of communication, the operator can connect to any such mobile robot, that is, to any available intranet site. Based on the fact that each agent is able to work with several IP sessions simultaneously, the mobile robot to which the operator has connected can receive information from other group members, and the information obtained can be structured (reduced to a relational form) and relayed to the operator.

Thus, each mobile robot in a group can act as a network gateway for accessing a distributed database (cloud connection point), providing structured data that can be easily interpreted in the operator's client application.

An IP connection can be initiated either from the operator's side to an agent (for the transmission of a command or a main task) or from an agent to the operator (to transmit the necessary information or request additional information or instructions).

The use of a decentralized control model implies that the operator's interaction with a group of mobile robots is carried out mainly by the high-level control (i. e. within the framework of setting a main task) and (if the solution cannot be developed within the group) at the strategic level. The group has to independently implement the direct decomposition of the main task, as well as the specific actions for the subtasks obtained at the tactical level, using the information database (DB) and behavior models database (BMD).

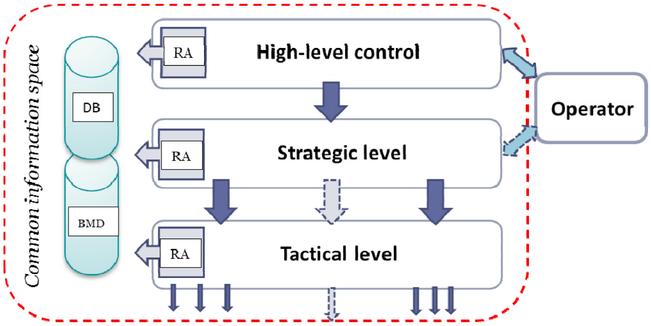

Fig. 2. Interaction between the operator and a group of mobile robots.

3.3. Providing self-control

The provision of self-control is largely determined by the set of hardware tools for implementing the hardware and software platform of a group member (agent). That is, the agent's hardware capabilities must be sufficient for autonomous sub-tasking within the main task assigned to the entire group. At the moment, Raspberry Pi 3 platform is used as a means of implementing hardware support of an agent at the strategic level. The platform stm32f407 is used for hardware support of an agent at the tactical (executive) level. The interaction between the high and low levels of control is performed by the interrupt system and the UART port. To implement a parallel communication port between the tactical and strategic levels, a 16-pin data bus and 3 additional control signals for interrupting and flagging are reserved.

The Raspberry Pi 3 platform has sufficient performance and the necessary number of external pins to connect various functional hardware modules (video camera, GLONASS / GPS, etc.) to develop algorithms and techniques for group (collective) control. Based on the capabilities of the BCM43438 radio module, used in the Raspberry Pi 3 platform, each agent can be both a Wi-Fi client and an access point at the same time. Using the capabilities of the Wi-Fi module, the connection between the agents is carried out by authorizing (as a client device) one (or several) agent in the Wi-Fi network of another agent, which in turn can also initiate a connection to another similar network or to operator.

Each agent has its own unique visible SSID of the RaspN type (where N is a unique ID), so when scanning wireless networks, each agent “sees” its closest environment. The name of the SSID of each agent can change, since the code containing the broadcast command (RaspN_broadcastCode) can also be added to the agent identifier. Therefore, an agent performs a direct connection to the specific wireless network by BSID, i. e. by MAC address. Each agent “knows” the BSID values of all the other agents in a group (and their current status). These values are contained in a separate table in the SQLite database, over which an agent automatically creates network routing rules.

The application layer for supporting an agent's network interaction with other agents and the operator is implemented using specialized software developed in Python3 using IP-SOCKET (SOCKET-server). The SOCKET server has a multithreaded architecture and, thus, allows up to 6 SOCKET connections to be serviced in a simultaneous mode: 1 for incoming agent session, 4 for incoming connections from other agents, and 1 for outgoing connection to the operator or another agent. The incoming connection of the operator is serviced on a separate (control) port.

The SOCKET server of each agent, in addition to transferring data between agents and the operator, can execute simple commands (for example, providing data from the internal database or system directives for executive devices) and commands that run (scenario) group interaction algorithms. All commands that are proper for execution are received through the “taskdesk” (program buffer) and sorted by priority (if assigned). The result of executing commands is written to the data buffer and then passed to the destination agent or operator (Fig. 3).

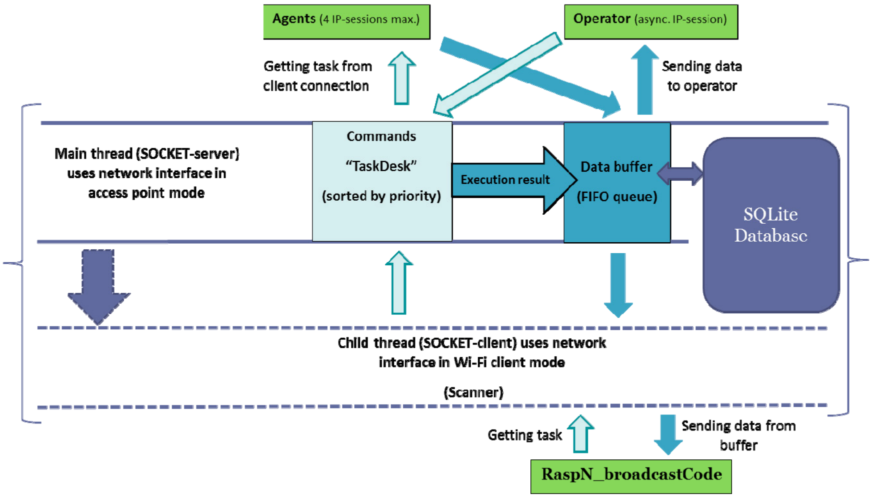

Fig. 3. Functional diagram of the managing software (SOCKET-server) of the agent.

Messages (both commands and data) are transmitted in blocks in json format:

{"Id": integer, "creator_MAC": string, "sender_MAC": string, "type": string, "data": string, "status": integer},

where:

- id – identifier (name) of an agent in the network;
- creator_MAC – MAC address of the device to which this message was created;
- sender_MAC – MAC address of the device to which this message was sent;
- type – RESPONSE / REQUEST / CMD (whose data field contains the command);
- data – data (information or command);
- status – message status code (0 if there are no errors, <0 if there is an error).

If type = CMD, and the value of the status field is greater than zero, then the status field indicates the priority of the command, i. e. the higher the value, the higher this command appears when it hits the "task pool".

4. Methods (scenarios) of group control

As already mentioned above, the cloud interaction model assumes the possibility of a simplified interaction of the operator with a group of mobile robots, that is, the possibility of setting up a main task in a simple (syntactically) form, for example: "explore the territory {coordinates of the region}", "attack the target {coordinates of the target(s)}" etc. It should also be taken into account that the directive from the operator can be sent to any agent of the group (cloud connection point). If the agent's functional capabilities or energy resources is insufficient to accomplish the task, in this case, a scenario is required to decompose the main task and distribute the sub-tasks in the group in accordance with the types of onboard functional purpose of certain mobile robots.

An algorithm for the distribution of tasks in a group of robots with different functional purpose was proposed within the framework of the research:

- Obtaining a main task (in a simplified syntactic form) from the operator by one of the agents (agent-coordinator).
- Determination of the task's semantic load by the coordinating agent (i. e. the type of task: combat, reconnaissance, rescue, etc.).
- Determination of the general types of onboard equipment necessary to accomplish the task (for example: armament, cross-country chassis, technical vision system, etc.).
- Transfer the digital code of the onboard equipment necessary to perform the task to all mobile robots in the group through a broadcast signal.
- Each agent determines the level of his "competence" (numerically) to solve this task based on the data received, taking into account its energy resources and current tasks.

- Agents transfer the received values to the coordinating agent, which selects the mobile robot with the highest value and passes on its functions to further coordinate the task (appoints a new coordinating agent).
- Since the new coordinating agent has the highest level of “competence” in solving a certain type of tasks, accordingly, it also has built-in algorithms for solving this problem, for example: algorithms for finding the optimal trajectory of movement, algorithms for cartography, combat operations, etc. In this case, this mobile robot determines the number of necessary agents (“competent”) for the task, distributes specific targets for each agent.

To determine the type of the main task and the corresponding onboard equipment, an intelligent method of parsing the text entered by the operator is used. This method uses the technique of image recognition using convolutional networks.

The task coming from the operator in text form is divided into words or phrases that are encoded into a 36-digit array containing indexes of corresponding letters in cells, for example, the word “intelligence” has the following form:

[[2, 8], 0, 4, 0, 6, 5, 0, 0, 3, 0, 0, 7, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, 0, 0, 0, 0, 0]

The program thus contains a “dictionary” with the codes of words corresponding to various types of tasks - a set of features or, by analogy with convolutional networks, the “convolution kernel”. The “convolution” operation is similarly applied to the word codes in the task text, i. e. elementwise multiplication of the word code (or expression) with the word code from the characteristics dictionary, then the definition of the arithmetic mean and application of the activation function to the obtained value. After the convolution, at the output we get the vector of values [0 ... 1], the number of such values corresponds to the types of tasks defined in the dictionary. This vector is transferred to the input of a classical fully connected artificial neural network, each output of which corresponds to a certain type of necessary onboard equipment (vision system, armament, manipulator, etc.). Thus, at the output we get a vector of values (code) that determines the type of necessary technical equipment for the task. This code is sent to all agents via the broadcast channel, and each agent submits it to the input of its own artificial neural network (or fuzzy controller), with the addition of two more values to the input vector:

- [0 ... 1] – the level of the current battery charge;
- [0 or 1] – the number of tasks in the data buffer (whether the TaskDesk is full or not).

This artificial neural network or fuzzy controller has a single output, the value of which varies within [0 ... 1]. This value determines the “competence level” of each agent.

The functional diagram of the method concerned is shown in Fig. 4.

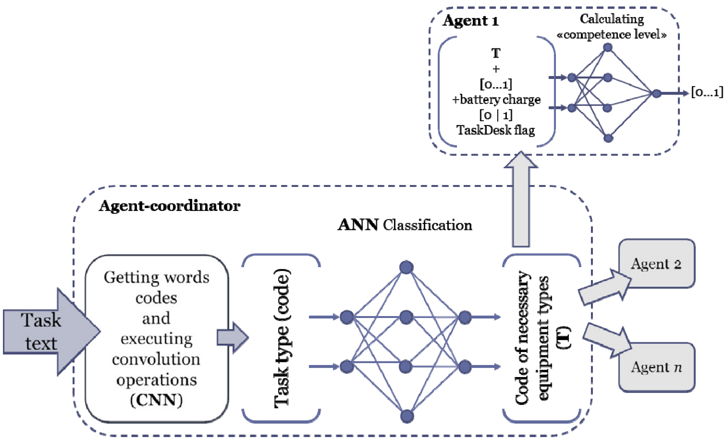

Fig. 4. Functional diagram of the task distribution method.

After assigning the tasks and defining the group members for their execution, each mobile robot performs its sub-task in an autonomous mode, using the corresponding built-in algorithms. For example, to solve the problem of simultaneous localization and mapping (SLAM), “competent” agents have algorithms for cartography based on ultrasonic or laser range finders, as well as algorithms for planning conflict-free trajectories for mobile robots in the common working space [6,7]. If additional decomposition of the subtask or an estimate of the effectiveness of its implementation is required within the formed (homogeneous) subgroup, it is possible to apply additional multi-agent control algorithms, for example, algorithms using the target function [1] or the “plan improvement” algorithm [3].

5. Simulation results

To carry out software simulation of the algorithm for the main task decomposition and the formation of the subgroups of agents based on their types of onboard equipment, a software module was developed in the programming language Python 3.5.6.

The text analyzer uses 5 test dictionaries of keywords and expressions in json format (the analysis for each of the dictionaries is performed in a separate thread):

- {«military» : [«attack», «destruction», «assault»]}
- {«exploring» : [«survey», «explore», «investigate»]}
- {«transport» : [«transport», «transfer», «deliver», «carry»]}
- {«rescue» : [«chemical», «radiation», «extreme temperature», «aggressive environment»]}
- {«communication» : [«communication channel», «communicate with operator», «communication support»]}

The artificial neural network has 5 neurons in the input layer, 50 neurons in the intermediate layer, and 7 in the output layer.

The output of the program is shown in Fig. 5:

a
```
Input cmd:
Investigate territory ()

Text analyzer (Text CNN output):
0.000000 military
1.002713 exploring
0.000000 transport
0.000000 rescue
0.000000 communication

Functional types of agents:
0.000000% military
100.000000% exploring
0.000000% transport
0.000000% rescue
0.000000% communication

Onboard systems classification (ANN output):
0.998183 Obstacles detection and ranging
0.990840 Global positioning system (GPS/GLONASS)
0.983967 Camera (Objects identification system)
0.013143 Longrange connection support with operator
0.013623 Weapon
0.008156 Sensors for environmental monitoring
0.998082 Off-Road Chassis
```

b
```
Input cmd:
investigate territory with extreme environmental conditions

Text analyzer (Text CNN output):
0.000000 military
1.002713 exploring
0.000000 transport
0.655465 rescue
0.000000 communication

Functional types of agents:
0.000000% military
60.470782% exploring
0.000000% transport
39.529218% rescue
0.000000% communication

Onboard systems classification (ANN output):
0.999977 Obstacles detection and ranging
0.999961 Global positioning system (GPS/GLONASS)
0.999945 Camera (Objects identification system)
0.938154 Longrange connection support with operator (LTE/sat
0.015289 Weapon
0.854241 Sensors for environmental monitoring
0.995789 Off-Road Chassis
```

Fig. 5. (a) – the result of the simple command analysis; (b) – the result of the command with an additional condition analysis.

As can be seen from the figure, the program first analyzes the text command for the keywords from the given dictionaries and, based on the “semantic load” of the text, determines the composition of the subgroup and the necessary onboard equipment of the agents.

6. Conclusion

At the moment, the methods of intellectual distribution of tasks and their practical application are being tested on the basis of hardware platforms Raspberry Pi 3 (at the strategic level) and stm32f407 (executive level), for solving the problems of simultaneous localization and mapping (SLAM) using the planning algorithms presented earlier [6].

Acknowledgements

This research was supported by Russian Foundation of Basic Research, projects no. 16-29-04165-ofi_m and Program No.29 “Advanced Topics of Robotic Systems” of the Presidium of the Russian Academy of Sciences.

References

[1] Diveev AI, Shmalko EYu, Zakharov DN. Acceleration of the multilayer network operator method using MPI for mobile robot team control synthesis. *XII International Symposium «Intelligent Systems», INTELS'16*. 5-7 October 2016, Moscow, Russia; p. 88-93.
[2] Skobelev PO, Simonovac EV, Zhilyaev AA, Travin VS. Application of multi-agent technology in the scheduling system of swarm of Earth remote sensing satellites. *XII International Symposium «Intelligent Systems», INTELS'16*. 5-7 October 2016, Moscow, Russia; p. 396-402.
[3] Mikhailov BB, Nazarova AV, Yushchenko AS. Autonomous mobile robots – navigation and control. *SFedU News. Technical science* 2016;2(175). URL: https://cyberleninka.ru/article/n/avtonomnye-mobilnye-roboty-navigatsiya-i-upravlenie (application date: May 28, 2013).
[4] Beloglazov DA, Soloviev VV, Titov AE, Finaev VI, Shapovalov IO. The method of distributing the goals in groups of intelligent mobile robots. *TSU News. Technical science* 2016:11-3. URL: https://cyberleninka.ru/article/n/metod-raspredeleniya-tseley-v-gruppah-intellektualnyh-mobilnyh-robotov (application date: May 28, 2013).
[5] Darintsev OV, Yudintsev BS, Alekseev AYu. The architecture of the control system of the robot team on the basis of cloud technologies. *The Tenth All-Russian Multiconference on Control Problems (MMP-2017), Proceedings of the 10th All-Russian Multiconference in 3 vol.* Managing Editor: IA. Kaliayev. Vol. 3. Robotics and Mechatronics (R&M-2017) 2017; p. 169-171.
[6] Yudintsev BS, Darintsev OV. Experimental studies of the efficiency of an artificial neural network trajectory planning system for a group of mini (micro) robots. *Proceedings of the 2nd International Conference «Information Technologies for Intelligent Decision Making Support» and the Intended International Workshop «Robots and Robotic Systems”*. Vol. 3. May 18-21, Ufa, Russia, 2014; p. 274-278.
[7] Yudintsev BS, Darintsev OV. Intellectual system for planning the trajectories of mobile robots, built on the Hopfield network. *Modern problems of science and education.* 2014;4; URL: https://elibrary.ru/item.asp?id=22285526 (reference date: November 14, 2017).