Research Complex for Unmanned Autonomous Vehicles Control Systems

Konstantin Yu. Kotov
Fuzzy Technologies Laboratory
Institute of Automation and Electrometry of the Siberian Branch of the Russian Academy of Sciences (IA&E SB RAS)
Novosibirsk, Russia
kotov@idisys.iae.nsk.su

Alexandr S. Maltsev
Fuzzy Technologies Laboratory
Institute of Automation and Electrometry of the Siberian Branch of the Russian Academy of Sciences (IA&E SB RAS)
Novosibirsk, Russia
alexandr@idisys.iae.nsk

Elizavita E. Prishlyak
Fuzzy Technologies Laboratory
Institute of Automation and Electrometry of the Siberian Branch of the Russian Academy of Sciences (IA&E SB RAS)
Novosibirsk, Russia
aliza-lis01@mail.ru

Maksim A. Sobolev
Fuzzy Technologies Laboratory
Institute of Automation and Electrometry of the Siberian Branch of the Russian Academy of Sciences (IA&E SB RAS)
Novosibirsk, Russia
smaxandr@gmail.com

***Abstract*—The software architecture of the testbench complex for unmanned vehicles is presented. The complex is intended for development, modeling and experimental research of control systems of wheeled, tracked and multi-rotor vehicles. This article presents the prototype of wheeled unmanned vehicle, created for training and bench tests in field of collaborative robotics control. Such test bench should solve the problems of researching the control algorithms of individual robots and their groups which ensure the formation of a group of robots with a given orientation in space and automatically control their joint movement along the desired trajectory with preserving the spatial configuration. The solution is achieved using the proposed software and hardware architecture, which is a set of independent programs that exchange signals through the Robot Operating System. The result is the control system for group of mobile robots that includes the motion surface, the technical vision system, mathematical methods of motion control and specialized software. This test bench can reduce the costs associated with the development, implementation, and testing of unmanned vehicle control systems.**

Keywords—mobile robots, prototype, motion control, simulation, technical vision

I. Introduction

Recently, in the tasks of controlling mobile robotic devices, in the first plan is the development of algorithms for controlling joint movement of a given formation [1, 2] or joint achievement of a set goal, such as moving a large load [3] or tracking an object [4]. For the possibility of research the characteristics and features of these algorithms, numerical modeling and experiments with real models of robots are carried out at specially created testing ground and stands [5-7]. The key tasks in the creation of such systems are navigation tasks, data exchange between individual robots in a group, algorithms for joint movement, avoiding obstacles and other robots.

At the same time, the algorithms for controlling the group and the robots in it can be ranked according to the degree of awareness of individual group members about the state of the entire system. For systems in which there is a connection between individual units, it becomes possible to distribute goals, assessing the capabilities of robots and the state of the system. This type of problem is reduced to finding the optimal distribution of roles, which are solved by linear programming methods [8], approaches to the construction of multi-agent systems [9], as well as by the methods of modern automatic control theory [10]

This paper presents a complex designed for the development and debugging of control systems for unmanned vehicles of wheeled, tracked and multi-rotor types in the tasks of trajectory motion control, group decentralized control, positioning in space and avoiding obstacles in conditions of uncertainty. The complex includes a simulation stand designed for testing and debugging prototypes of control systems, a set of special configurable software that implements original methods of navigation and motion control of unmanned autonomous vehicles and prototypes of robots with special expansion boards that provide communication and control functions Fig. 1-2.

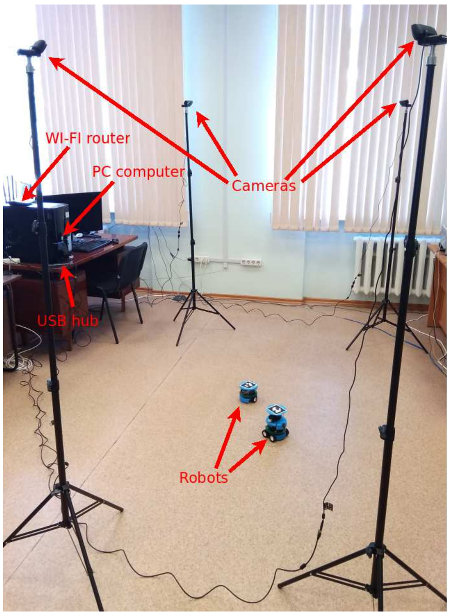


Fig. 1. External view of complex.

978-1-6654-3408-9/21/$31.00 ©2021 IEEE

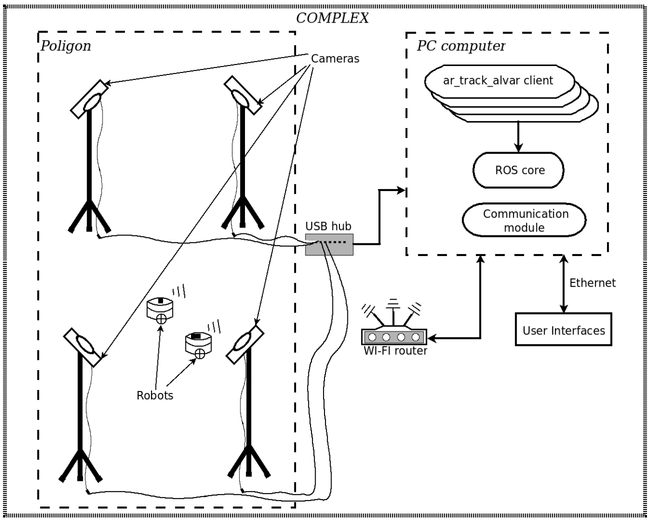


Fig. 2. Component connection framework of complex.

The purpose of this robot is to develop a control complex for a group of unmanned vehicles to solve the problem of joint movement in a given formation along the required route. The architecture of the software, the data exchange mechanism and control algorithms provide the ability to add a new robot to the group, the ability to form both an independent task for each vehicle, and a common task of movement in the required formation for all robots in the group. The complex provides operation in the simulation mode for studying the properties of the laws of motion control and development of system components, and in the control mode of real wheeled, tracked and multirotor robots.

II. Software Architecture

The software for the testbench complex is written in Python and is a set of independent processes (modules). The exchange of data between modules is carried out through a common data server using the Robot Operating System (ROS). The complex includes several modules [11]:

A. *Single Robot Control Module*

Generates control signals for linear and angular velocities based on information about the current and desired position of the robot at each moment of time. To calculate the current position, a navigation system based on the Kalman filter is used, in which the motion model data is combined with coordinates data from an external vision system. The mathematical algorithm for generating motion control commands is described in [11]. The transmission of commands of linear and angular velocities to the robot's motors occurs through the driver of the on-board system based on ROS [12].

B. *Single Robot Simulation Module*

It includes a mathematical model of the movement of the unmanned vehicle, receives signals from the motion control unit of a separate robot as input, calculates and publishes the current coordinates and orientation angles of the vehicle. The module is designed to use the complex in the simulation mode for software development and study the properties of the control laws of an individual robot and a group of robots as a whole.

C. *Coordinate Determination Module*

It is represented by two processes, the first of which is a tcp client that collects data from the Swtrack vision system [13], and the second process performs processing of the received data and writes information about the current position of each robot and its heading angle to a common data server. The computer vision system receives data from webcams mounted on top, jointly calibrated in a single coordinate system tied to the plane of movement of the robots. In the development of this complex, the Swtrack-based vision system was replaced by the *ar_track_alvar* [14] module from the Robot Operating System. In both cases, special markers are installed on the robots. Unlike swtrack, the system based on AR markers allows determining not only two coordinates and a heading angle using a marker, but a complete set of three coordinates and three orientation angles of the robot Fig. 2.

D. *Visualization Module*

Graphic program for indicating the position and orientation of each robot and load, implemented on the basis of the Gtk and Cairo libraries. The program also provides input of commands from the operator, which allows setting the formation of robots and the route of movement on the map of the area Fig. 3.

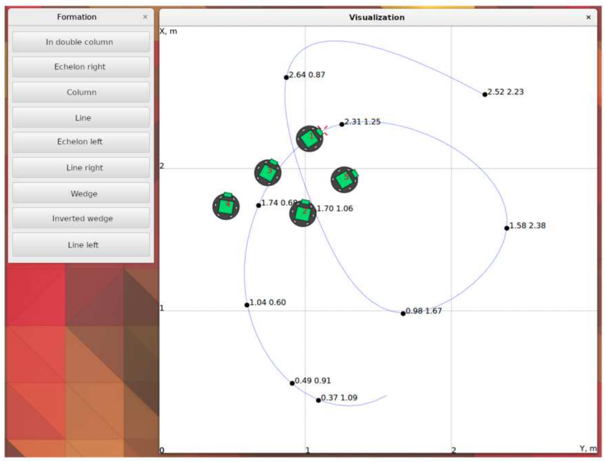


Fig. 3. Indication of robot position and route planning.

E. *Group Control Module*

The process of forming high-level commands for control systems of an individual robot and the trajectory of the group and the distribution of robots relative to the group leader. The group control system receives a command from the operator and displays information about the order and position of each robot relative to the group leader. Each robot acts independently of the others in accordance with the command for the entire group, data on the position of the group leader, the required location relative to him and data on the position of other robots in order to avoid collisions. The robots, independently of each other, receiving the command and the rest of the data, begin to move, maintaining a given position relative to the group leader and each other in accordance with the mathematical control law. Each robot sets its current status in the network storage according to which the grouping control module sets the status of the entire group for indication to the operator. After the completion of the action, each robot sets a flag on the completion of the current command, and when the flags from all robots are set, the group control generates a signal about the execution of the current command by the whole group.

The control of the unmanned vehicle as part of the group is carried out using the software module for bringing the object to the target position, the module for determining the identifier and coordinates of the object and the client module to the on-board control system. The exchange of data between program modules is also carried out through the organization of a shared memory resource of the Redis data server. The information about the coordinates of the object and the data received from the onboard sensors, necessary for controlling the movement of the object, is sent to the data server. All software modules function as independent processes in asynchronous mode.

In the process of determining the identifier and coordinates of the object, the image from the video cameras of the system is processed for determining the coordinates of objects. The tasks performed by the software module for determining the identifier and coordinates of an object include color transformations of a video stream, thresholding of video frames, search and recognition of markers, and calculation of absolute coordinates.

After defining and identifying the objects, the command from the group management system is read by the data server. In the subroutines for bringing the robots to the target position, the received command is compared with the known ones and the corresponding program module is launched. The block diagram of the software is shown in Fig. 4.

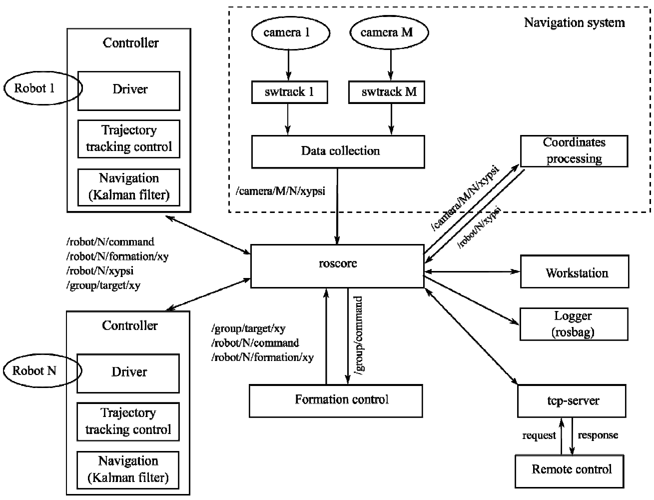


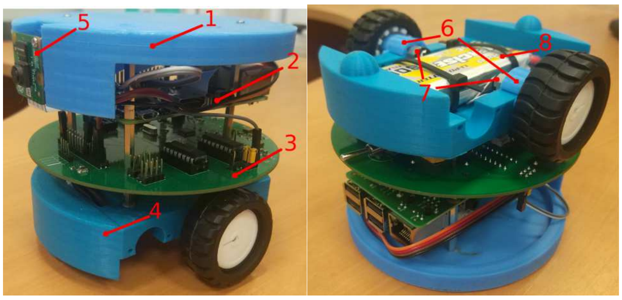
Fig. 4. The software block diagram.

III. A Prototype of a Wheeled Unmanned Vehicle

In 2020, the first prototype of a mobile wheeled robot of its own design was developed in the laboratory of fuzzy technologies of the IA&E SB RAS. The main defining requirements were:

- modularity of design, providing the ability to connect additional equipment and batteries;
- compact size, allowing both bench and floor experiments;
- the possibility of replacing stepper motors with DC motors with increased shaft rotation frequency to ensure sufficient movement speed;
- the ability to control the discharge of batteries and controlled shutdown of DC-DC conversions by a microcontroller signal - parking function at the charging station;
- ensuring the operation of stepper motors in half/full step mode.

The appearance of the created prototype is shown in Fig. 5.



a b

Fig. 5. Prototype unmanned vehicle of wheeled type. (a - front view: 1-cap, 2-microcomputer, 3-control board, 4-chassis, 5-camera; b - underside view: 6-mufts, 7-stepper of DC Motors, 8- Li-Po battery).

The robot consists of a main chassis, DC motor bushings, control board, microcomputer, camera cover and other equipment. The main battery is a Li-Po battery located at the bottom of the chassis base.

Fig. 6 shows the appearance of a prototype with a four-wheel chassis layout and an installed Lidar (circular view lidar).

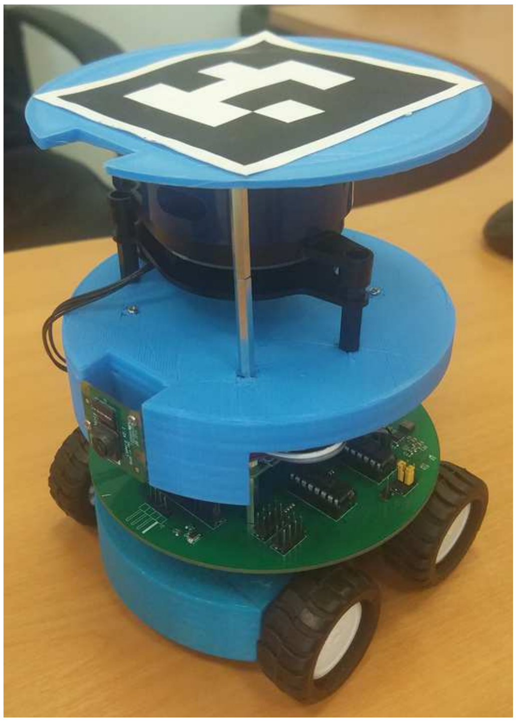


Fig. 6. The appearance of a prototype with a four-wheel chassis layout and an installed Lidar.

The robot control system has a two-level structure, Fig. 7.

Data exchange between controllers of the upper and lower levels is realized via the serial interface UART or SPI, which has a large bandwidth. A 32-bit STM32F103C8T6 controller with a clock frequency of 72 MHz is used as a low-level control controller. The functions of the lower-level controller are to receive control commands to measure the speed of rotation of the motors, obtain information about the battery charge level, receive data from the encoders of the motors, generate a PWM control signal to the motor windings according to a given algorithm.

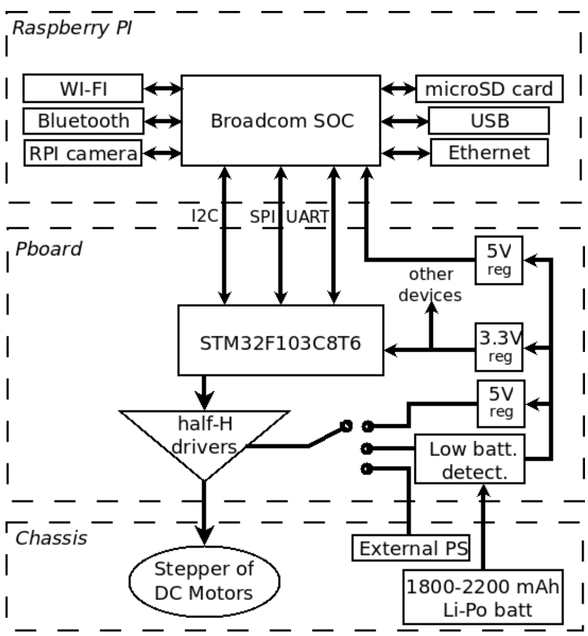


Fig. 7. The control system block diagram.

Calculation of control actions taking into account a given route of movement and current coordinates of the device, integration and filtering of sensor data are performed at the upper level - a single-board microcomputer Raspberry Pi 3B with a 4-core Cortex-A53 (ARM v8) 1.2 GHz processor and 1 GB RAM. The operating system is Ubuntu 20.04 LTS.

The software of the created control system has a modular architecture and is built on the mechanism of interprocess communication Robot Operation System. The system includes software modules that perform the following functions:

- communication between controllers of the upper and lower levels (rosserial protocol);
- receiving images from the front camera;
- receiving data from the lidar of a circular view;
- trajectory motion control and state vector estimation in the extended Kalman filter;
- data logging.

IV. CONCLUSION

To control a group of mobile autonomous objects, a software and hardware complex that includes mathematical laws for the formation of control commands that ensure movement along specified trajectories, maintaining the required location of objects in the group and rebuilding the group, methods for avoiding obstacles, evaluating and compensating for the error of location determination algorithms in autonomous movement in a previously unknown environment has been developed.

Universal control methods, tools for modeling and testing, which are present in the complex, allow reducing the development time and costs of creating control systems for real objects. The objects can be both ground robots and unmanned autonomous vehicles, the use of which can significantly save human resources and increase efficiency in ensuring life safety. The use of the complex makes it possible to reduce the costs associated with the development, implementation and testing of control systems for both single and groups of unmanned autonomous vehicles.

The presented complex was used to create a demonstrator for group control of wheeled robots and to develop the laws of flight control along a given trajectory of quad-rotor unmanned vehicles.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (state registration №. 121042900050-6)

REFERENCES

[1] P. Ogren, E. Fiorelli, and L.N. Ehrich, “Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment,” IEEE Transactions on Automatic Control. 2004; 49(8):1292–1302.

[2] V. Jose, A. Lounis, and M. Youcef, “Adaptive leader – follower formation in cluttered environment using dynamic target reconfiguration,” Distributed Autonomous Robotic Systems. Part of the Springer Tracts in Advanced Robotics book series. 2016; (112):237–254.

[3] J.S. Ortiz, J.V. Aldas, and V.H. Andaluz, “Mobile manipulators for cooperative transportation of objects in common,” Towards Autonomous Robotic Systems: 18th Annual Conference (TAROS, Guildford, UK, July 19–21, 2017). Part of the Lecture Notes in Computer Science. 2017; (10454):651–660.

[4] K. Hausman, J. Muller, A. Hariharan, N. Ayanian, G. Sukhatme, “Cooperative multi-robot control for target tracking with onboard sensing,” The International Journal of Robotics Research. 2015; (34):1660–1677.

[5] M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokic, E. Wilhelm, R. Bouffanais, K. Dick, and P. Yue, “Swarm-enabling technology for multi-robot systems,” Frontiers in Robotics and AI. 2017; (4):11–22.

[6] W. Zheng, F. Zhou, and Z. Wang, “External vision based robust pose estimation system for a quadrotor in outdo or environments,” Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM-2014). Setubal: Science and Technology Publications; 2014: 718–723.

[7] A.G. Millard, J.A. Hilder, J. Timmis, and A. F. T. Winfield, “A low-cost real-time tracking infrastructure for ground-based rob ot swarms,” Proceedings of the 9th International Conference (ANTS 2014). Gewerbestr: Springer International Publishing; 2014: 172–184.

[8] F. Borrelli, D. Subramanian, A.U. Raghunathan, and L. T. Biegler, “MILP and NLP techniques for centralized tra jectory planning of multiple unmanned air vehicles,” Proceedings of the 2006 American Control Conference. Minneap olis: IEEE; 2006: 217–223.

[9] M. Figat, and C. Zielinski, “Methodology of designing multi-agent robot control systems utilising hierarchical Petri nets,” IEEE International Conference on Robotics and Automation (ICRA-2019). Montreal: IEEE; 2019: 44–50.

[10] S.A. Belokon’, Yu.N. Zolotukhin, K.Yu. Kotov, A.S. Mal’tsev, A.A. Nesterov, M.A. Sobolev, M.N. Filippov and A.P. Yan, “Control of the mobile robots in a leader – follower formation [Upravleniye mobil'nymi robotami v sostave gruppy lider—vedomyye]”, *(in Russian)*, Mechatronics, Automation, Control [Mekhatronika, Avtomatizatsiya, Upravlenie], vol. 17, no. 3, pp. 166—172.

[11] K. Yu. Kotov, A.S. Maltsev, E.E. Prishlyak and M.A. Sobolev, “Group control complex for unmanned vehicles [Kompleks gruppovogo upravleniya bespilotnymi apparatami]”, *(in Russian)*, Computational Technologies [Vychislitel'nyye tekhnologii], 2021; 26(1):99–111.

[12] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, “Ros: An open-source robot operating system,” Proceedings of ICRA Workshop on Open-Source Software. IEEE; 2009; (3):142–146.

[13] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, “Swistrack — A flexible Open-Source tracking software for multi-agent systems,” Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2008: 4004–4010.

[14] ROS AR TRACK ALVAR Package Summary. http://wiki.ros.org http://wiki.ros.org/ar_track_alvar (accesed Aug. 1, 2021)