

Development of a Navigation System for an Educational Mobile Robot

Andrew A. Boshlyakov^{1*} and Alexander V. Maltsev²

¹Bauman Moscow State Technical University, 2nd Baumanskaya str., 5/1, 105005, Moscow, Russia

Abstract. The process of developing a navigation system for an educational mobile robot is considered. The analysis of the existing navigation systems of mobile robots is carried out, the hardware of the navigation systems is presented, their advantages and disadvantages are shown. A diagram of the interaction of on-board subsystems during the autonomous movement of a mobile robot is presented. The design of the basic chassis of the educational mobile robot and its technical characteristics are described. A block diagram of the interaction of ROS packages (robotic operating system) for creating a terrain map using a laser ranger is developed, as well as a block diagram of the interaction of ROS packages for the an educational mobile robot tracing. Considering the studies, conclusions are drawn on the applicability of the proposed implementation of a laser ranger based navigation system for an educational mobile robot.

1 Introduction

The successful development of robotics requires a solid and deep scientific base, which is created by combining efforts of scientists in the fields of mechanics, applied mathematics, control theory, computer science. Mobile robots are a very convenient object for defying, studying and finding solutions to modern problems of mechatronics. These robots have already left laboratory stands and are widely used in everyday life. Industrial mobile robots have long been a usual means of optimizing production cycles in the motor-car, aircraft, ship construction and instrument making enterprises. They deliver materials, technological or other equipment to the place of work. Robots are used by security officials for detecting and destroying explosive devices. Robotic systems are being intensively introduced into medical practice as equipment for delivering necessary things to patients. There is an active development and launching process of consumer robots for domestic use [1].

Mobile robots include a range of devices: sensors, actuators, computers or artificial intelligence elements. Their design and management require qualified specialists, the preparation of which calls for educational robots. Therefore, the involvement of students and graduates in the mobile robots research will allow to achieve a qualitative improvement in the training of highly qualified specialists [2].

* Corresponding author: boshlyakov@bmstu.ru

The autonomous control system for the mobile robot movement is a complex structure and requires the creation of a number of interconnected on-board systems, such as EMFS – environment model forming system; MPS - motion planning system; ES - executive system; NS - navigation system; SD - servo drives; TVS – technical vision system; CDB - cartographic database. The interaction of these systems in the aggregate is a diagram of an autonomous motion control system (AMCS) of a mobile robot (fig. 1). The first step in the implementation of the motion control system of an educational mobile robot is the development of a navigation system [3].

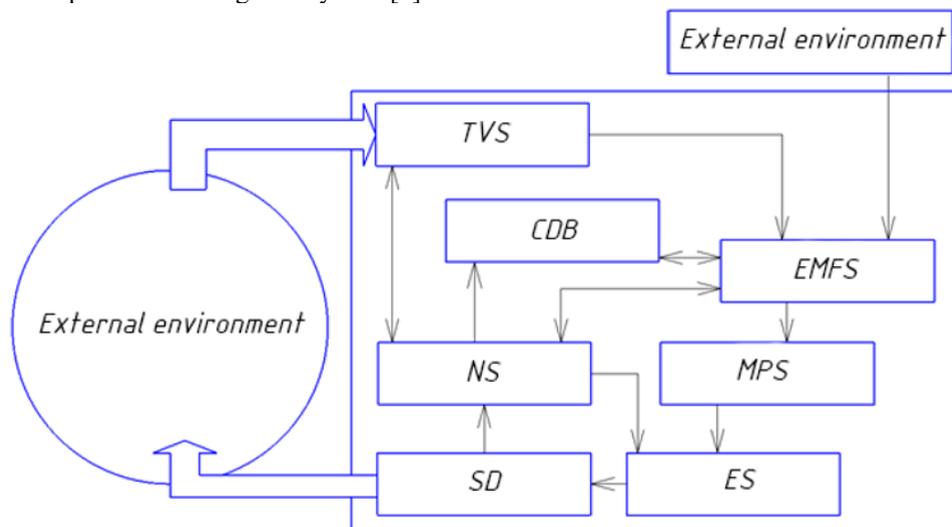


Fig. 1. The structural diagram of the AMCS.

This paper proposes to use a radio-controlled machine as a base for the educational robot, the design of the electromechanical modules for controlling the movement of the camera and wheels of the educational mobile robot was developed, as well as a platform for placing an on-board computer, sensors and a technical vision system.

The purpose of this article is to develop a navigation system for an educational mobile robot.

2 Educational mobile robot design

Before developing a navigation system for an educational mobile robot, it is advisable to consider the composition and design of the basic chassis.

As a basic platform for an educational mobile robot, a commercially available semi-professional radio-controlled car model Kyosho EP Inferno VE Race Spec was chosen (fig. 2). The model has a symmetrical all-wheel drive and a completely independent suspension, its speed reaches 70 ... 80 km/h. The chassis is made of durable types of plastic, which provide withstanding collisions at fairly high speeds [4].

An educational mobile robot serves to develop the principles of designing and implementing the basic systems of mobile robots with a basic set of tasks. The user should be able to quickly repair or replace faulty elements, which means that he should easily access them.

Considering the design of the basic chassis, the use of sensors types that need to be mounted directly on the mechanical part or inside the structure will be inappropriate.

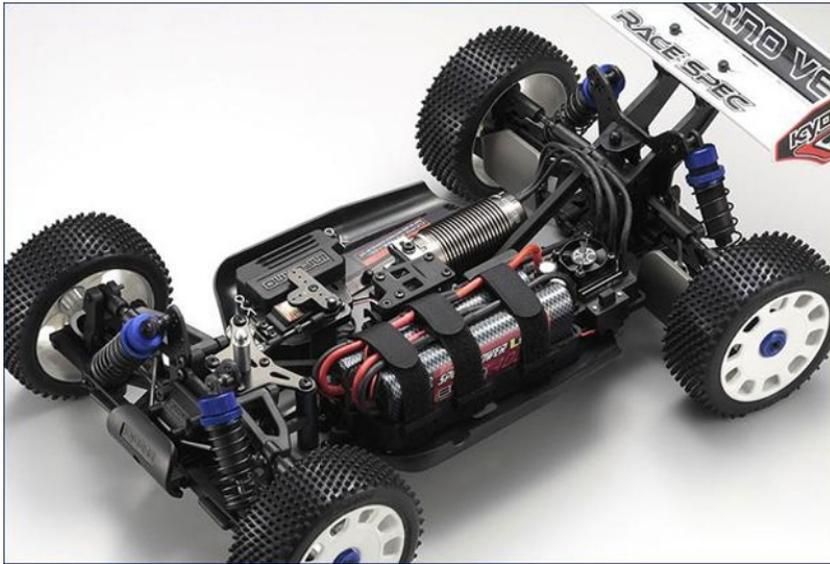


Fig. 2. Basic radio-controlled car.

3 An analysis of options for building a navigation system for an educational mobile robot

A literature review was conducted to select a navigation system for an educational mobile robot [5–7]. The process of literary sources analysis made it clear that a navigation system construction is significantly difficult. Firstly, it is necessary to determine the minimum hardware that is used in most existing mobile robots. This is done so that the educational robot includes the most affordable, but necessary set of elements. Secondly, students can disable the main components of the navigation system while working with an educational mobile robot, so it is unacceptable to use hardware that requires intervention in the robot design, because in this case the replacement of faulty elements will be a very time-consuming process. Thirdly, after selecting the hardware part, it is necessary to pre-select the type of navigation system and the algorithm suitable for its implementation. Fourth, it is necessary to program the required algorithm in order for the navigation system to perform it. To do this, it is advisable to choose a software or development environment that allows you to work with selected hardware elements.

For successful navigation in space, the on-board robot system must be able to build a route, control motion parameters, for example, the rotation angle of the wheels and their rotation speed. It is necessary to correctly interpret the information about the world received from the sensors, and constantly monitor own coordinates. The principle of navigation system operation also depends on the type that the developer chooses, and the type of navigation system is determined by the hardware. Consider the main types of navigation systems for mobile robots [8]:

1. passive navigation system - information about its own coordinates and other characteristics of its movement is received from external sources, for example: GPS or radio beacons;

active navigation system - information about its own coordinates and other characteristics of its movement is received from sources located on the body or in the construction of a mobile robot, for example: encoders, accelerometers, laser rangefinders, etc.

3.1 Choosing a hardware of a navigation system for an educational mobile robot

Considering the fact that the educational mobile robot is being developed based on the KYOSHO Inferno radio-controlled model for working in a structured environment, some limitations are imposed on its navigation system hardware. Since the dimensions of the educational mobile robot are less than one meter, and the GPS has measurement accuracy within a few meters, the use of this system can lead to inaccurate positioning and the robot will follow an incorrect path. Beacons are located at fixed points of a route, and the educational mobile robot loses the ability to choose an alternative path and recognize the position of local obstacles. Thus, the use of a passive navigation system is impractical.

The hardware of the active navigation system includes mechanical and optical gyroscopes and accelerometers. Such sensors are used to determine speed, rotation angle, acceleration, but it is impossible to determine the presence of obstacles on the route using them (similar to odometers, with which it is also impossible to determine the obstacle presence). But, in addition to this, such sensors must be installed inside the educational robot, therefore, the replacement of faulty parts will be much more difficult. Also, stereoscopic systems, requiring good lighting, are referred to the active navigation system, they are difficult to implement, and their accuracy depends on the distance to the object observed.

Analyzing the operation of some navigation systems and the parameters of various sensors, it can be concluded that for an educational mobile robot, it is most efficient and profitable to use the Lidar neato xv-11 laser ranger. This sensor is used in robotic vacuum cleaners; it has a small cost relative to other sensors of this type. The accuracy of laser ranger is about ten millimeters, and the viewing angle is 360 degrees, unlike other sensors. Using a lidar, an educational mobile robot will be able to determine its position and the presence of local obstacles at the same time, without using other hardware elements, which will allow it to correctly build a route depending on the presence of a clear path and with the required accuracy.

Thus, some elements were selected considering the frequency of their use in modern robotics, so that students work with relevant and modern devices. However, elements such as sensors are very expensive and it will be extremely difficult to replace them, so the choice of lidar was based only on the price criterion and not on its technical parameters.

The power source for the entire system is a 14.4V Li-Po 2S zippy battery. The supply voltage from the battery goes to the system elements via picoPSU - 160 - XT DC / DC voltage converters from 19V to 3.3, 5 and 12V. The converters were chosen so that they have a built in overcurrent protection. The voltage to the travel controller is supplied directly from the battery. The regulator is a controller that generates the required voltage and supplies power to the servo drive, which is responsible for the movement speed and direction of the educational mobile robot.

From the 3.3V voltage converter, power is supplied to the lidar dc motor, the control of this motor is already organized by the manufacturer, so a separate input for the control signal is not provided. The rotation speed of the lidar is from 3 to 5 hertz, this is not enough to solve serious problems in real conditions, but it is enough for students to understand the principle and be able to see the results of the operation of this type sensors, and the cost of the sensors differs by two orders.

From the 5V voltage converter, power is supplied to the STM32F407DISCOVERY microcontroller, camera rotation servos and wheel rotation servo. The servo drives of the camera are responsible for rotation along the vertical axis.

From the 12V voltage converter, power is supplied to the intel NUC on-board computer. It is possible to power the Intel NUC directly, bypassing the converter, however, at the time

of connection, the current can increase sharply and the on-board computer may be damaged.

The control signal is sent from the operator's remote computer to the on-board computer, after which it is processed and sent to the microcontroller via COM-port. Next, the microcontroller generates control signals in the form of pulse-width modulated signals and transmits them to the servos.

Also, on-board computer receives information from the lidar via the USB-UART interface.



Fig. 3. Educational mobile robot prototype.

3.2 Development of ROS based software of a navigation system for an educational mobile robot

The hardware of the navigation system does not differ from the hardware of the technical vision system, and the technical vision system software is the first step in building a navigation system for an educational mobile robot.

The following tasks and limitations were set for the navigation system:

- global map creation for a room during the first run of the system over an unknown terrain in manual control mode;
- displaying the location of the educational mobile robot on a global map with an accuracy of 0.5 meters (considering the dimensions of the robot);
- the use of devices in the navigation system that involve a change in design is prohibited;
- displaying the obstacles on the local map that are not presented on the global map.

The navigation system in ROS comes down to the need of a data source that receives information about the external environment and converts it such a way to receive the coordinates of the base_link frame relative to the map frame. The general relationship of

frames has already been described for mobile platforms. In this case, the `base_footprint` frame does not provide information about the height of the robot relative to the map or odom frame, the `base_link` frame is rigidly fixed to the robot body and includes changes in roll and pitch angles compared to the `base_stabilized` frame. The odom frame is calculated on the basis of the odometry source, it is convenient for local maps because it is indiscrete, but it cannot be used for a long time, because the odometry accumulates an error and there is a high probability of getting the mobile robot in the wrong position. This structure is tree-like and is presented in fig. 4.

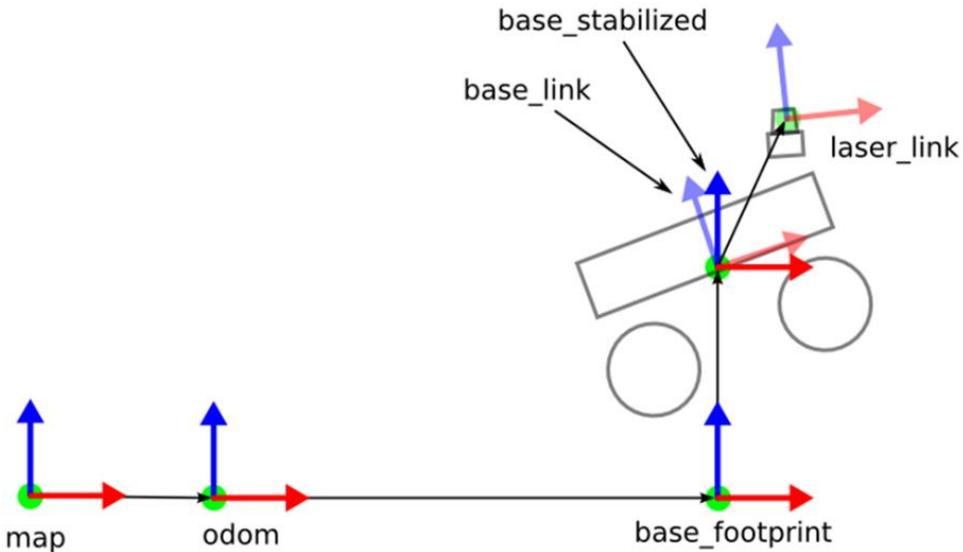


Fig. 4. Frame interaction for mobile platforms.

The GMapping package solves the SLAM problem based on the data received from the laser ranger, however, to solve the localization problem, it is necessary to connect the `amcl` package in parallel. But even in this case normal operation is not guaranteed, since GMapping requires an odometry source. This problem was solved by the functionality of the laser scan matcher package, and Map Server was used for communication between `amcl` and GMapping. Thus, four packages will be considered [5]:

- GMapping;
- AMCL;
- Laser scan matcher;
- Map Server.

Using the `slam_gmapping` node from the GMapping package, a 2D room map can be created. Node receives messages from the `tf` topic in the form of transformations necessary for the connection of laser, base and odometry frames. And it publishes the messages in topics:

- `map_metadata` - receiving data about a map, which is updated with a given periodicity;
- `map` - representation of map data in a form suitable for display in `rviz`.

There are also a number of parameters that can be configured, but in this case they are all set by default.

At the second stage, when the room map was received, it is necessary to initiate the localization of the mobile robot. This can be done using the AMCL package.

AMCL is a probabilistic localization system for a robot moving on a plane surface. Such a system implements the adaptive Monte Carlo approach, which uses a particle filter

to track the position of the robot on a known map. This approach only works with maps obtained by laser rangefinders.

Fig. 5 shows how the adaptive approach works. At the initial moment of time, the robot is at some indefinite spot and depending on the objects surrounding it, the robot determines its possible position, and the most probable of all robot positions is displayed.

The amcl node subscribes to the following topics:

- tf (tf/tfMessage) – receives the necessary frames;
- scan (sensor_msgs/LaserScan) - laser ranger data;
- initialpose (geometry_msgs/PoseWithCovarianceStamped) - this topic is necessary to initialize the particle filter;
- map (nav_msgs/OccupancyGrid) - a map that was built in advance.

The amcl node publishes the following topics:

- amcl_pose (geometry_msgs/PoseWithCovarianceStamped) - estimated location of mobile robot system (MRS);
- particlecloud (geometry_msgs/PoseArray) - a cloud of particle points.

Amcl and gmapping interact through Map Server package. GMapping publishes a map in the map topic, and amcl subscribed to this topic reads this map. Also, it is possible to save the room map and use it if necessary by means of the map_saver node of the map server package.

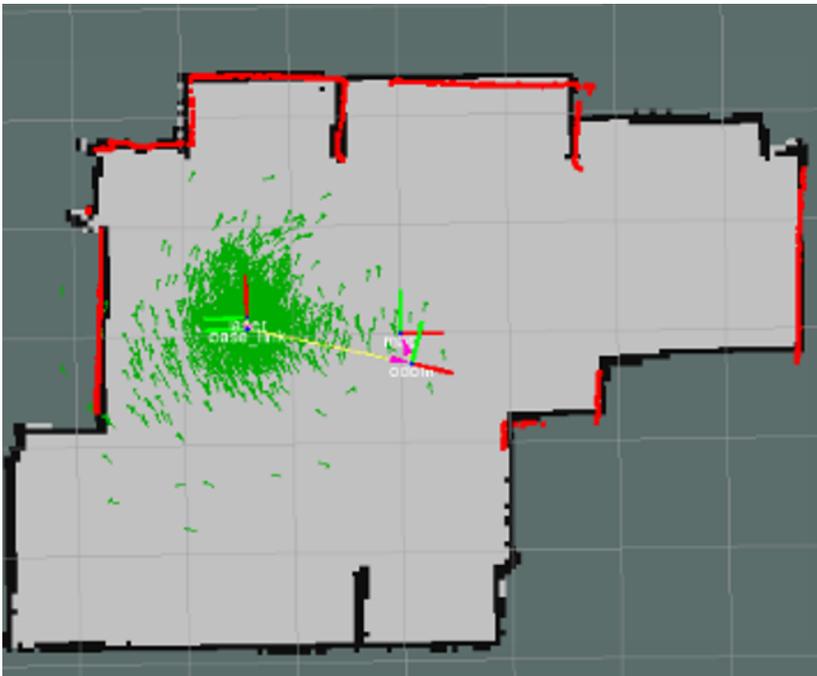


Fig. 5. amcl package operation.

Next, consider the laser scan matcher package. At this stage, a room map, the robot localization in the form of a laser frame, i.e. the position of the lidar relative to the map frame, i.e. coordinate systems of the map were obtained. But this is not enough for the correct operation, because without the tree structure that the ROS developers made, this approach will not work, so the transition between map and laser frames through the odom frame should be implemented.

This problem was solved by using the laser scan matcher package. It is a tool that allows you to compare matches between consecutive sensor messages and publish the

estimated position of the odometry source relative to the map frame. Laser_scan_matcher node is subscribed to topics:

- scan - data received from a laser ranger;
- cloud - data representation in the point cloud form;
- odom - odometer reports. It is used if set to true.

The laser_scan_matcher node publishes a topic:

- pose2D - position of the robot in the selected frame.

The node also has many parameters, which in this case are selected experimentally.

For this package to work correctly, base_link - laser frames have to be already implemented, this task is performed through the tf topic and map - base_link. The amcl package solves this task.

Thus, with the correct interaction and configuration of all four packages, the problem of simultaneous localization and mapping is solved.

The results were much better when using the GMapping package in conjunction with the amcl, laser scan matcher and map server packages, than with the Hector Slam package. Despite Hector Slam, GMapping does not impose restrictions on the rotation speed of the lidar.

A room map was constructed (fig. 6) and used in further work as the basis for the navigation system [6].

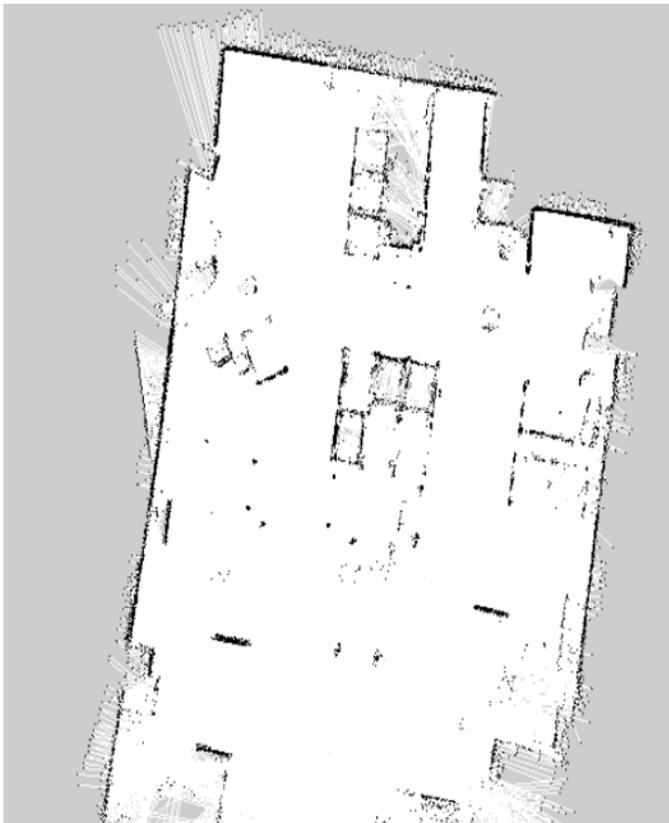


Fig. 6. Example of GMapping package operation.

3.3 Development of ROS based software of a navigation system for an educational mobile robot in autonomous and non-autonomous modes

The control system hardware includes the on-board computer, the operator's computer, the joystick, and the stm32f4discovery microcontroller. ROS, Clion, and PyCharm software were installed on the intel NUC and on the operator's computer.

In manual control mode, data from the joystick is sent to the operator's computer, where it is received by the joy node and sent as a float type message to the joy topic. Because there is the common ROS master on the operator's computer and on-board computer, then data from topics is available on both devices [7].

After the data has arrived in the topic on the on-board computer, it must be redirected to the microcontroller. To solve this problem there are libraries `ros.py`, HAL and server libraries in CLion.

The capabilities of the `ros.py` library allow to create nodes in ROS, receive messages from topics into familiar variables and functions, as in the usual object-oriented approach, and therefore messages that ROS uses can be transmitted as variable values.

The capabilities of the server library allow to receive and transmit data via the COM-port. It is necessary that messages received from the joy topic are transmitted to the microcontroller.

The HAL libraries and the STM32CUBEmx program were integrated into CLion for programming stm32f4discovery. The STM32CUBEmx program sets the timing and initial settings of the necessary modules, in this case USB and timers. The USB module is configured as COM-port, since the data transfer speed on the serial port is sufficient for the control signals to arrive in series and processed sequentially, the serial port is also programmed faster, unlike the HID device.

Thus, the signal is sent from the source of the control signal - the joystick to the joy node, then it is read using the program code and sent via the COM-port to the microcontroller. At the microcontroller, the signal is received into the buffer, from the buffer it passes through the processing function, and it is redirected to timers to generate the servo protocol and to send the signal to the servo drives.

The difference between autonomous mode and non-autonomous mode is that during autonomous movement, the control signal is generated by the `teb_local_planner` path planner in ROS, and not by the joystick. The operator sets the target on the map, the planner generates a trajectory and a message with two parameters is sent to the topic `cmd_vel`, one of which is proportional to the speed of movement, and the second - to the wheel rotation angle [9].

For the planner to operate, it is necessary that all previous stages of development work, namely the technical vision system and navigation system, as opposed to movement in manual control mode, where the navigation system is more of an auxiliary character to display the position.

The planner generates a local obstacle map. If some objects are not displayed on the global map, they will be considered when planning a route to a given target. The route is formed according to the a-star or Dijkstra's algorithm, depending on the choice of operator.

The `teb_local_planner` package forms the trajectory in two stages, the first stage is the formation of the trajectory on the global map and it is optimized during the run time, minimizing the trajectory's execution time, considering the defined distance to obstacles and the dimensions of the robot.

Topics published by `teb_local_planner`:

- `global_plan` - global trajectory that the robot is trying to work out;
- `Local_plan` - a local trajectory that is optimized by the planner and followed by the robot.

As well as `teb_poses`, `teb_markers`, `teb_feedback`.

teb_local_planner is subscribed to topics:

- Obstacles - local obstacles;
- Via_points – target setting.

Before using the planner, you must also configure the move_base package, which is responsible for calculating speeds and rotation angles. To use move_base, configuration files must be defined that specify all the physical parameters of the robot, the size of the local map, maximum rotation angles, maximum speed, etc.

All parameters in each case are determined by a unique experimental method [10].

Fig. 7 shows a fragment of the system in autonomous mode. The possible positions of the mobile robot are indicated in red, the coordinate system is the actual position, the white square with black dots represent the local map and local obstacles. Fig. 8 shows an example of setting a target; the path is shown in green.

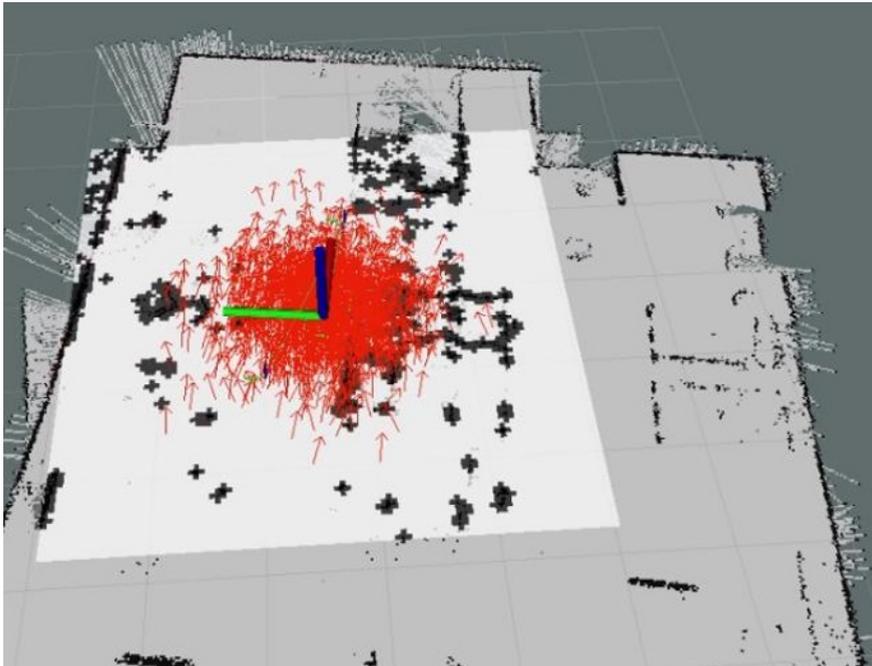


Fig. 7. A fragment of the system in autonomous mode.

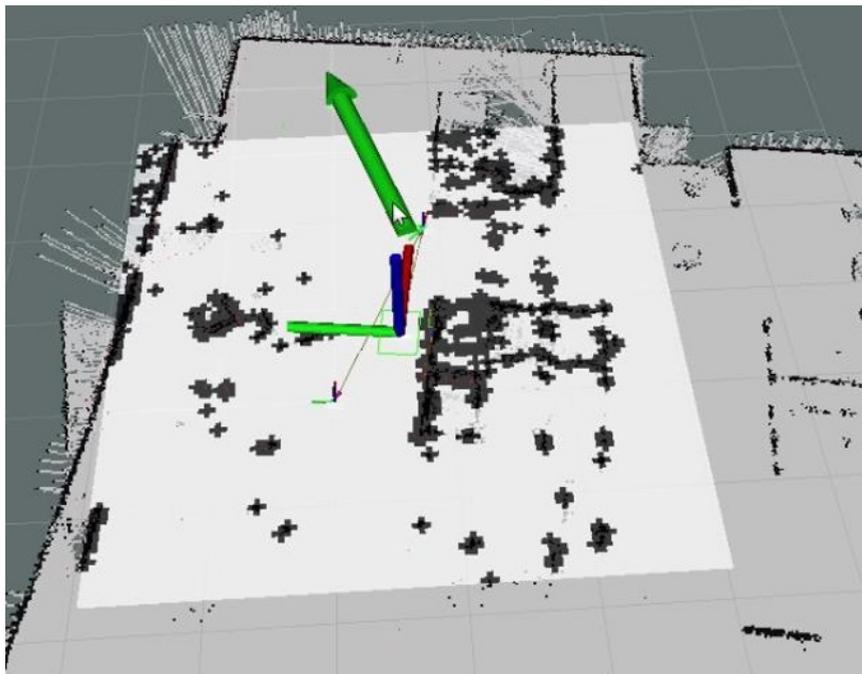


Fig. 8. Target setting.

4 Conclusion

Analysis of the basic chassis design of the educational mobile robot and the structure of the mobile robot control system allowed to select elements of the navigation system and form its structure. Carefully designed hardware does not require significant changes of the robot design and allows to easily rebuild the robot control system depending on the required configuration.

The developed software for processing data obtained during the operation of the laser ranger, showed the possibility of locating the educational robot in the area. The used composition of the ROS packages, namely: `amcl`, `map_server` and `gmapping`, allows to determine the position of the educational mobile robot in rooms and training laboratories based on a probabilistic localization system using a laser ranger.

Courses: Programming robotic control systems, designing technical vision and navigation systems of robots, microprocessor technology.

This training robot is an open system that allows to build and adapt it to the needs of specific learning areas.

In the future, it is planned to finalize the software, which will be able to determine the spatial position of the robot with an accuracy of a centimeter.

References

1. B.B. Mikhaylov, A.V. Nazarov, A.S. Yushchenko, *Autonomous mobile robots- navigation and control (in Russ.)*, Izvestiya YuFU. Tekhnicheskie nauki, no. 2, pp. 48–67, (2016).
2. V.S. Lapshov, V.P. Noskov, I.V. Rubtsov, *Experience of development special-purpose autonomous mobile robotic complexes (in Russ.)*, Vestn. Mosk. Gos. Tekh. Univ. im.

- N.E. Baumana, Priborostr. [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], spec. iss. «Special robotics and mechatronics», pp. 7–24. (2011).
3. V.P. Noskov, V.I. Rubtsov, I.V. Rubtsov, *Matematicheskie modeli dvizheniya i sistemy tekhnicheskogo zreniya mobil'nykh robototekhnicheskikh kompleksov [Mathematical motion models and technical vision systems of mobile robotic complexes] (in Russ.)*, Moscow, Bauman MSTU Publ., (2015).
 4. Radiupravlyaemaya mashina s elektrodvigatelem KYOSHO 1:8 EP 4WD Inferno VE RTR [Radio-guided machine with KYOSHO 1:8 EP 4WD Inferno VE RTR electro engine]. hobbycenter.ru: website. URL: https://hobbycenter.ru/kyosho/1_8_ep_4wd_inferno_ve_rtr.html (accessed: 15.08.2019). (in Russ.).
 5. A.V. Tkachev, S.V. Shanygin, *Overview of the mobile robots that use the on-board navigation system for offline path planning to a specified target (in Russ.)*, Molodoy uchenyy [Young Scientist], no. 19, pp. 215–219 (2019).
 6. A.A. Minin, *Parameter estimation of mobile robot self-motion equipped with laser range sensor. Ekstremal'naya robototekhnika (in Russ.)*, Tr. Vseros. nauch.-tekh. konf. [Extremal robotics. Proc. Russ. Sci.-Tech. Conf.], Sankt-Petersburg, pp. 205–213 (2007).
 7. V.N. Gerasimov, *Sistema navigatsii servisnogo robota v srede s dinamicheskimi prepyatstviyami (in Russ.)*, Avtoref. diss. kand. tekhn. nauk [Navigation system of service robot in environment with dynamic obstacles], Moscow, Bauman MSTU publ. (2015).
 8. G.P. Kalach, G.G. Kalach, S.A. Travnikov, *Perspectives the use of robotic systems in the interests of the military security of the state*, Nauka. Obschestvo. Oborona, no. 1(6) (2016). DOI: 10.24411/2311-1763-2016-00005 URL: <https://www.noo-journal.ru/nauka-obshestvo-oborona/2016-1-6/article-0060/> (in Russ.).