UDC 004.4, 004.932.7, 004.021

Implementation of the Pathfinding System for Autonomous Navigation of Mobile Ground Robot

Andrey V. Bokovoy[*†], Maxim B. Fomin[*], Konstantin S. Yakovlev[*‡]

[*] *Department of Information Technologies, Peoples' Friendship University of Russia (RUDN University), Miklukho-Maklaya str. 6, Moscow, 117198, Russia*

[†] *Institute for Problems of Artificial Intelligence, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Vavilova str. 44/2, Moscow, 119333, Russia*

[‡] *Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia*

Email: 1042160097@rudn.university, fomin_mb@rudn.university, yakovlev@isa.ru

The paper considers the problem of autonomous navigation of unmanned ground vehicle and the way to solve it by using simultaneous localization and mapping methods based on the data, provided by the laser rangefinder, and path planning algorithms. We propose the control system's architecture (including low-level communication protocols and high-level planning and mapping mechanisms) and it's implementation based on Robot Operating System (ROS) framework. The system is planned to use as a toolbox for pathplanning algorithms evaluation on real robotic system in real environment. We also provide the visualization of current state of system. The evaluation is carried out on a Nexus wheeled robot, which specification is also given. Future work includes multi-robot modification of developed system (shared map over all users of the system), exploration algorithms implementation (including multi-agent exploration), multi-agent pathplanning algorithms embedding and moving the whole system's operation on board of mobile ground robotic system.

Key words and phrases: ground unmanned vehicle, mobile robot, localization and mapping, laser rangefinder, path planning.

In: K. E. Samouylov, L. A. Sevastianov, D. S. Kulyabov (eds.): Selected Papers of the 1st Workshop (Summer Session) in the framework of the Conference "Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems", Tampere, Finland, 20–23 August, 2018, published at http://ceur-ws.org

1. Introduction

Nowadays unmanned ground vehicles (mobile robots) are widely used for academic and practical purposes. The application area of such robots varies from being evaluation testbeds for methods and algorithms developed in laboratory environment to real world applications like search-and-rescue, monitoring, guarding etc. Increasing the autonomy is one of the core tasks in modern robotics [1, 2] as, obviously, autonomous mobile robots have more capacities to solve the tasks than remotely-operated robots, especially when the large groups and coalitions of robotic systems are involved [3–5]. The ability for self-navigation (e.g. without external control by a human) in known or unknown environment is the basic block needed to achieve high level of robot's autonomy [6].

Different approaches to solve the navigation problem exist. Applicability of different approaches depends on the properties of the robot's environment (known or unknown environment, availability of global position systems and etc.) and on-board sensors. Modern navigation methods can be split to **reactive** and **deliberative**. **Reactive** methods operate by handling "at-the-moment" sensors' information and perform movement based on the current state of the system and the surrounding environment. This approach is mainly used for navigation in dynamically changing environment, e.g. for obstacle-avoiding tasks, path following etc. This approach is also useful when the time horizon is short and decisions should be made very quickly. **Deliberative** methods assume that the robot posses some information about the environment (e.g. it has the map) and performs the navigation taking into account this information. This class is represented by path planning algorithms, simultaneous localization and mapping (SLAM) algorithms etc. We follow the deliberative approach in this work.

The main tasks we consider are mapping of unknown environment, localization in resultant (or known apriori) map and path planning. We follow the typical approach when localization and mapping are combined into the coherent SLAM, e.g. simultaneous localization and mapping, framework [7–10]. A decision on how to solve a SLAM problem, e.g. which method to use, depends vastly on the type of on-board sensors mobile robot carries. In our case we rely on the scanning laser rangefinder (lidar) and inertial measurement unit, and utilize the SLAM method, that integrates the information from these sensors. The output of the SLAM algorithm is the 2-D map (occupancy grid) of the environment with the blocked and free areas pointed out. Having such a map well-established heuristic search algorithms like A* and others [11, 12] can be utilized to solve path planning queries.

The main goal of this work is to investigate the ways of creating a coherent, modularized software control system used for autonomous navigation of a wheeled robot having SLAM and pathfinding as the main components that can be plugged in and out for evaluating different approaches and methods.

2. Specification of navigation system

2.1. Hardware and software organization of ground robot

As the experimental platform for developing the navigation system, we use ground unmanned vehicle, that is based on Nexus platform as shown in Fig. 1.

The hardware specification of robotic platform:

- Chassis with kinematic differential schema;
- Driving wheel drive with reducer and encoder;
- Chassis controller;
- Battery;
- On-board embedded controller ODROID-C2 [13]:
    - ARM architecture;
    - 4 cores;
    - 2 GHz;
    - 2 GB RAM;

Figure 1. Ground research platform

  - 8 GB flash memory.
- Web-camera;
- 6-DOF manipulator;
- Scanning lidar;
- Programmable servo controller;
- Wireless router.

The on-board computer is powerful enough to run Linux-based operating system and is capable of processing complex algorithm in real-time. As the main framework for interfacing individual components (sensors, controllers and etc.) and autonomous control we chose Robot Operating System (ROS) [14].

Using ROS for system's organization makes low-level robot control (servomotors signal control, sensor access etc.) more abstract for end-user through standard high-level protocol called topics and services. Also, ROS makes applications, built for this framework, modular by executing different parts independently in nodes.

The communication with ground platform is done using Wi-Fi, making possible to control robot remotely in autonomous, semi-autonomous and manual modes. In case of fully autonomous control, all the algorithms may be ported directly to ground robot without additional rework.

The on-board computer grants standard interface to the following control mechanisms and sensors' data:

- Laser rangefinder (lidar) data in LaserScan format
- Environment map in OccupancyGrid format
- Odometry data in Odom format
- Robot's position in resultant map in geometry_msgs/PoseStamped format
- Robot movements using control vector with geometry_msgs/Twist commands
- Position following (geometry_msgs/Pose)

The architecture of hardware' communication is shown in Fig. 2.

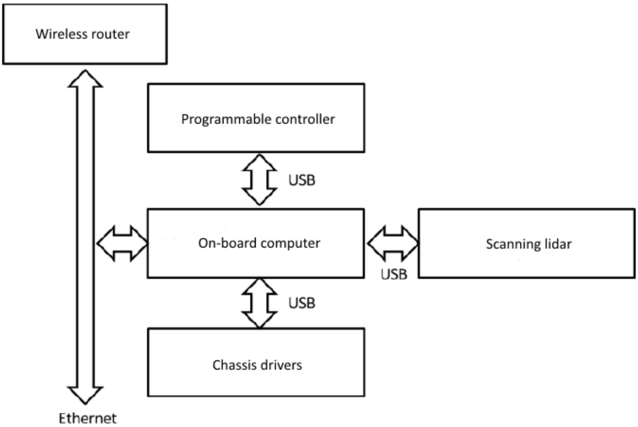

Figure 2. Hardware communication architecture

Localization and mapping is done using on-board computer with data from lidar sensor and odometry. For simultaneous localization and mapping problem we use known algorithm "gmapping" [15], that uses an array of distances from robot to objects in the environment and the data from chassis' sensors to build new map or update an excising map of unknown environment and localize the robot in this map. Also, the algorithm makes possible to localize robot in known map.

The on-board computer also implements the action_server and action_client mechanism for known-goal-following tasks. In case of path following algorithm, mobile platform "Nexus" implements the Pure Pursuit [16, 17] following algorithm.

2.2. Software architecture of autonomous control system

We use C++11 for autonomous control system implementation. Application was released as a ROS framework node, that uses the standard access interface for system's components for normal processing, visualization with RViz [18] package and debugging (rosbag).

The system consists of 2 main parts:

1. Pathplanner
2. action_client for pathfollowing and building the trajectory

The software architecture of control system is shown in Fig.3. For demonstration purposes of pathplanning algorithm we use Theta* [19], but the system provides the mechanism to replace the pathplanning algorithm with other one. For future work, we plan to experimentally evaluate more complex pathplanning algorithms, including angle-constrained methods LIAN [20].

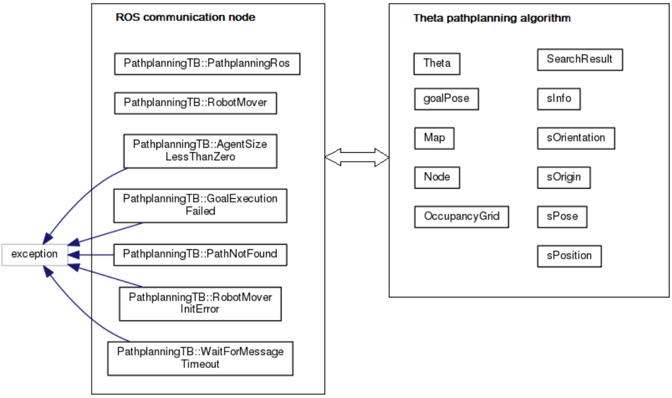

Figure 3. Software architecture for autonomous control system

The system can operate in 2 modes: pathplanning in known and unknown environments. For known environments, unknown parts of map considered as obstacles and the trajectory planner assumes this cells occupied. For unknown environment, unknown parts of the map considered as the new type of cells. As the algorithm proceeds, the system plans the trajectory as if this cells were free. When the system reaches such a cell, system replans the trajectory using new observations and map.

3. System's demonstration

Fig. 5 shows the visualization of our control system's execution using RViz ROS package. The figure demonstrates the mapping, localization in built map and the trajectory from robotic system to goal point planned with Theta* algorithm.

As an experemental environment, we used the 6x3.5m room. The map was built using lidar and odometry data. The size of a single cell is 0.05m (the size is related to lidar's error). The trajectory is planned without concidering the robot's size. The robot's position and orientation are corrected during movements using inertial measurement unit data. If the robot goes far from planned trajectory, then the robot's position may be corrected manually.

The pathplannig output then goes to pathfollowing algorithm as an array of midpoints and the robot proceed towards this points one-by-one. In case of significant deviation from planned trajectory, the robot attempts to return to the nearest position of built trajectory. The robot moves with 0.2 m/s speed.

This demonstes the working capacity of our system in real environment using PathPlanning and SLAM algorithms with "Nexus" robotic platform.

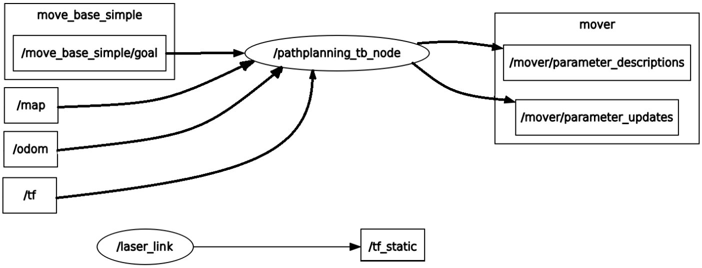

Figure 4. Software architecture for autonomous control system

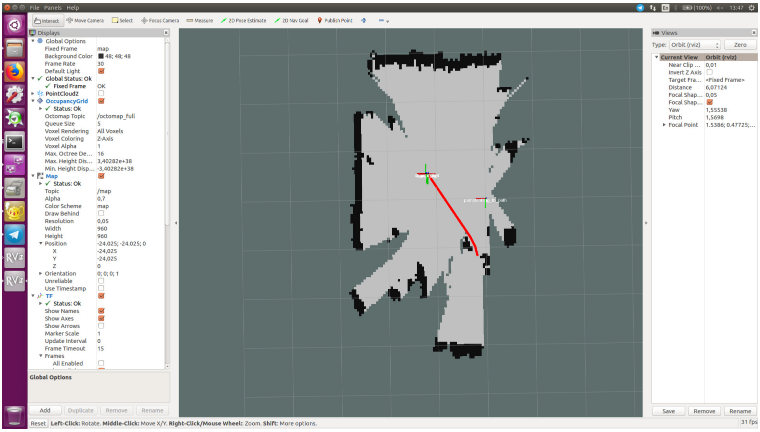

Figure 5. Software architecture for autonomous control system

4. Conclusions

This work presents the specification for unmanned ground robot autonomous control system based on "Nexus" platform. We described the software architecture and the mechanism of communication with robotic system. The final system may be executed on remote control system or on-board of ground robot. Also, the system may be adapted to any occupancy grid-based pathplanning algorithm.

The results provide us an opportunity to evaluate modern pathplanning algorithms on real robots.

Acknowledgments

The publication has been prepared with the support of the "RUDN University Program 5-100" and partially supported by RFBR grant No 17-07-00281.

References

1. T. Li, X. Chang, Z. Wu, J. Li, G. Shao, X. Deng, J. Qiu, B. Guo, G. Zhang, Q. He, et al., Autonomous collision-free navigation of microvehicles in complex and dynamically changing environments, ACS nano 11 (9) (2017) 9268–9275.
2. S. Emel'yanov, D. Makarov, A. I. Panov, K. Yakovlev, Multilayer cognitive architecture for uav control, Cognitive Systems Research 39 (2016) 58–72.
3. M. Sokolov, R. Lavrenov, A. Gabdullin, I. Afanasyev, E. Magid, 3d modelling and simulation of a crawler robot in ros/gazebo, in: Proceedings of the 4th International Conference on Control, Mechatronics and Automation, ACM, 2016, pp. 61–65.
4. L. Vig, J. A. Adams, Multi-robot coalition formation, IEEE transactions on robotics 22 (4) (2006) 637–649.
5. J. Guerrero, G. Oliver, Multi-robot coalition formation in real-time scenarios, Robotics and Autonomous Systems 60 (10) (2012) 1295–1307.
6. E. Magid, D. Keren, E. Rivlin, I. Yavneh, Spline-based robot navigation, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, 2006, pp. 2296–2301.
7. S. Thrun, Simultaneous localization and mapping, in: Robotics and cognitive approaches to spatial mapping, Springer, 2007, pp. 13–41.
8. H. Choset, K. Nagatani, Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization, IEEE Transactions on robotics and automation 17 (2) (2001) 125–137.
9. C. Stachniss, J. J. Leonard, S. Thrun, Simultaneous localization and mapping, in: Springer Handbook of Robotics, Springer, 2016, pp. 1153–1176.
10. J. J. Leonard, H. F. Durrant-Whyte, Simultaneous map building and localization for an autonomous mobile robot, in: Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on, Ieee, 1991, pp. 1442–1447.
11. A. Andreychuk, A. Bokovoy, K. Yakovlev, An empirical evaluation of grid-based path planning algorithms on widely used in robotics raspberry pi platform, in: Proceedings of The 2018 International Conference on Artificial Life and Robotics (ICAROB2018).
12. K. Yakovlev, Hga*, an efficient algorithm for path planning in a plane, Scientific and Technical Information Processing 37 (6) (2010) 438–447.
13. M. Hähnel, H. Härtig, Heterogeneity by the numbers: a study of the ODROID XU+ E big. LITTLE platform, in: Proceedings of the 6th USENIX conference on Power-Aware Computing and Systems, USENIX Association, 2014, pp. 3–3.
14. M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, Ros: an open-source Robot Operating System, in: ICRA workshop on open source software, Vol. 3, Kobe, Japan, 2009, p. 5.
15. J. M. Santos, D. Portugal, R. P. Rocha, An evaluation of 2d slam techniques available in robot operating system, in: Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, IEEE, 2013, pp. 1–6.
16. R. C. Coulter, Implementation of the pure pursuit path tracking algorithm, Tech. rep., Carnegie-Mellon UNIV Pittsburgh PA Robotics INST (1992).
17. T. Hellstrom, O. Ringdahl, Follow the past: a path-tracking algorithm for autonomous vehicles, International journal of vehicle autonomous systems 4 (2-4) (2006) 216–224.
18. H. R. Kam, S.-H. Lee, T. Park, C.-H. Kim, Rviz: a toolkit for real domain data visualization, Telecommunication Systems 60 (2) (2015) 337–345.
19. A. Nash, K. Daniel, S. Koenig, A. Felner, Theta^*: any-angle path planning on grids, in: AAAI, Vol. 7, 2007, pp. 1177–1183.
20. K. Yakovlev, D. Makarov, E. Baskin, Automatic path planning for an unmanned drone with constrained flight dynamics, Scientific and Technical Information Processing 42 (5) (2015) 347–358.