International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2479-2482

Offline Drone Control in Simulator

A.V.Koluzov[1], I.S.Shustov[1], R.N.Sadekov[1], R.R.Bikmaev[1] and M.Sh.Gareev[1*]

[1] *Federal State Autonomous Institution "Military Innovation Technopolis "ERA", Anapa, 353456, Russia.*

[2] *NUST MISIS, Moscow, 119049, Russia.*

Abstract

The results of the development of an autonomous control system for a drone moving indoors are presented. Aircraft navigation is based on data from a laser lidar and an inertial module. The drone is capable to bend around obstacles and find a way out of the premises. The results are demonstrated on a virtual training ground.

Keyword: UAV, Drone, Navigation, Autonomous Control Systems, SLAM.

1. INTRODUCTION

Currently, the use of unmanned aerial vehicles (UAVs) in the military is becoming increasingly important. The recent attack by a small number of UAVs on oil-producing complexes in Saudi Arabia has proved the economic efficiency of using such weapons. Relatively small investments in the acquisition of UAVs and the training of operators, if used properly, can lead to multibillion-dollar losses on the party that became the target of the attack.

The use of UAVs is often carried out in conditions of communication interference with the operator, the signal loss of the satellite navigation system (SNA) due to landscape conditions or the impact of electronic warfare. In this regard, it is extremely important to develop a system for autonomous navigation and UAV control based on information from onboard sensors.

Currently, existing solutions in the field of UAV automated control systems use SNA information and are capable of only providing a short-term flight if it disappears. Inertial navigation systems do not provide high positioning accuracy for a long time.

When UAVs move in conditions of dense urban environment, sensors of technical vision systems (cameras and lidars) acquire a special role. For orientation and navigation of UAVs in well-explored rooms, it is advisable to use wall markers recognized by the technical vision system. Such technology can be implemented on the basis of the Aruco library [1]. The disadvantages of using this approach are infrastructure dependency.

This drawback can be avoided by using simultaneous localization and mapping (SLAM) approaches. This technology allows to build a map of the surrounding space and navigate in it in previously unknown places. SLAM algorithms can be built on the basis of both optical and laser sensors.

Using cameras, for example using direct sparse odometry (DSO) packages, allows to get a space map and navigate in unknown rooms, however, due to the great computational complexity, lower accuracy and light dependence, using this approach is time-consuming [2, 3]. A simpler, more reliable and more accurate approach is the use of 2D laser lidars.

In a virtual environment, there is a drone model consisting of a 2D lidar, an altimeter and inertial sensor modules. The drone must fly in an unknown space - a room containing obstacles and fly out of it when the target object, the doorway, is recognized. It is required to develop an autonomous control system (ACS) that ensures that the drone performs the task with this equipment and flight conditions.

2. COMPLEX SOLUTION

2.1 Overview

Autonomous control system of drone must solve some complex subtasks like that:

- navigation in a non-deterministic environment;
- recognition of obstacles;
- route planning;
- drone control.

The structure of the ACS is shown in Figure 1. The data sources are on-board sensors: lidar, altimeter, module of inertial sensors. The system operates in a robot operating system (ROS) environment (kinetic version) and Ubuntu 16.4.

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2479-2482

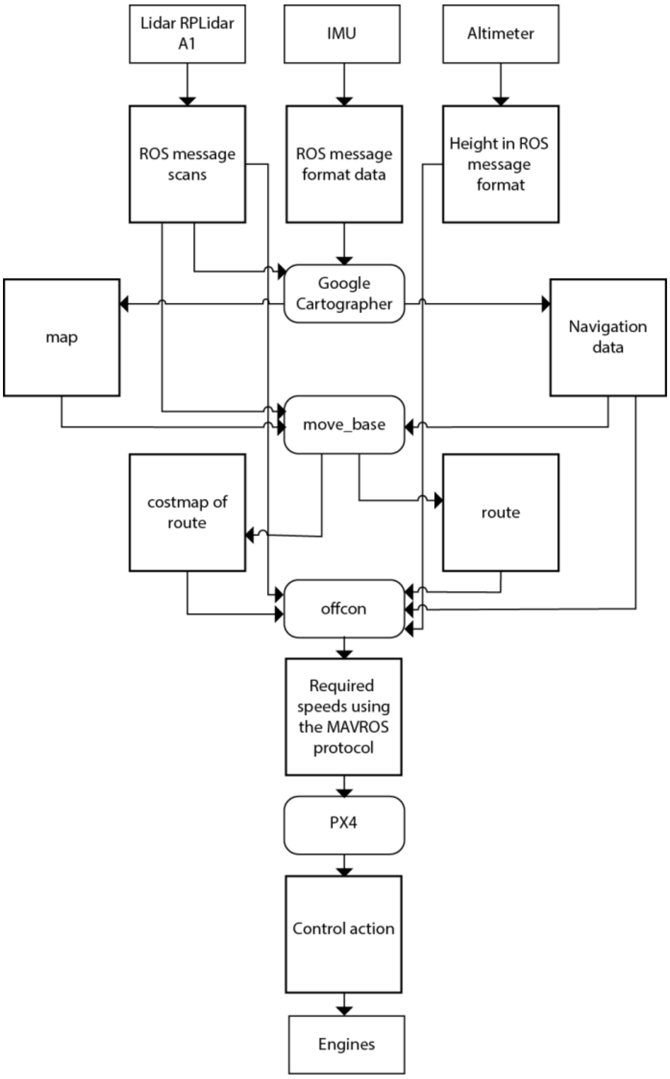

Fig 1. The main elements and data flows of the control system

2.2 Navigation Subsystem

The navigation module is implemented in the form of a Google Cartographer package - special software from Google. Flexible package configuration allows the use of a wide range of sensors. The parameters presented in table 1 are responsible for the configuration of the input data set.

The map-building algorithm is represented by a two-layer architecture: the local layer (Local SLAM) and the global layer (Global SLAM).

After pre-processing, the data from the on-board sensors enter the local layer, which is responsible for compiling a local map attached to the drone. The lidar data is correlated with each other, merged into a single scan, and a submap is compiled that has the minimum possible distortion. The submap passes through the motion filter with the parameters motion_filter (max_time_seconds, max_distance_meters, max_angle_radians) (see table 1) to cut off redundant maps. Submaps are connected to each other based on a search for common contours of the selected obstacles into a single global map of the entire known space in the global mapping layer. The position of the drone is also determined in this layer [4].

Table 1. Google Cartographer Settings

Parameter Name	Value	Description
num_accumulated_range_data	1	Data from the lidar is not accumulated for insertion into a submap, but is used immediately
use_imu_data	True	Inertial sensors are used
motion_filter.max_time_seconds	0,5	Motion filter, maximum relevance time of the last submap in seconds
motion_filter.max_distance_meters	0,01	Motion filter, maximum displacement value for discarding a submap in meters
motion_filter.max_angle_radians	0,01	Motion filter, maximum value of the rotation angle for discarding the submap in rad

2.3 Recognition of Obstacles and Route Planning Subsystem

The move_base node are used to build a movement route. It converts the cartographer's map into a cost map, with the help of which the route is calculated based on the algorithms "A *" [5], "D *" [6], Dijkstra's algorithm [7]. The cost map is shown in Figure 2.

There is the two-dimensional map in the problem to be solved. Each pixel of the map has a value from 0 to 252. This value represents the cost of passing the route through the pixel. The move_base node calculates the route to the target using the Dijkstra algorithm, based on the cost map, minimizing the total cost of the path. In this case, close obstacles obtained using the lidar are combined into one obstacle based on the parameters of the corresponding cluster_* parameters. The total cost consists of the length of the route and the costs of each pixel along which this route passes.

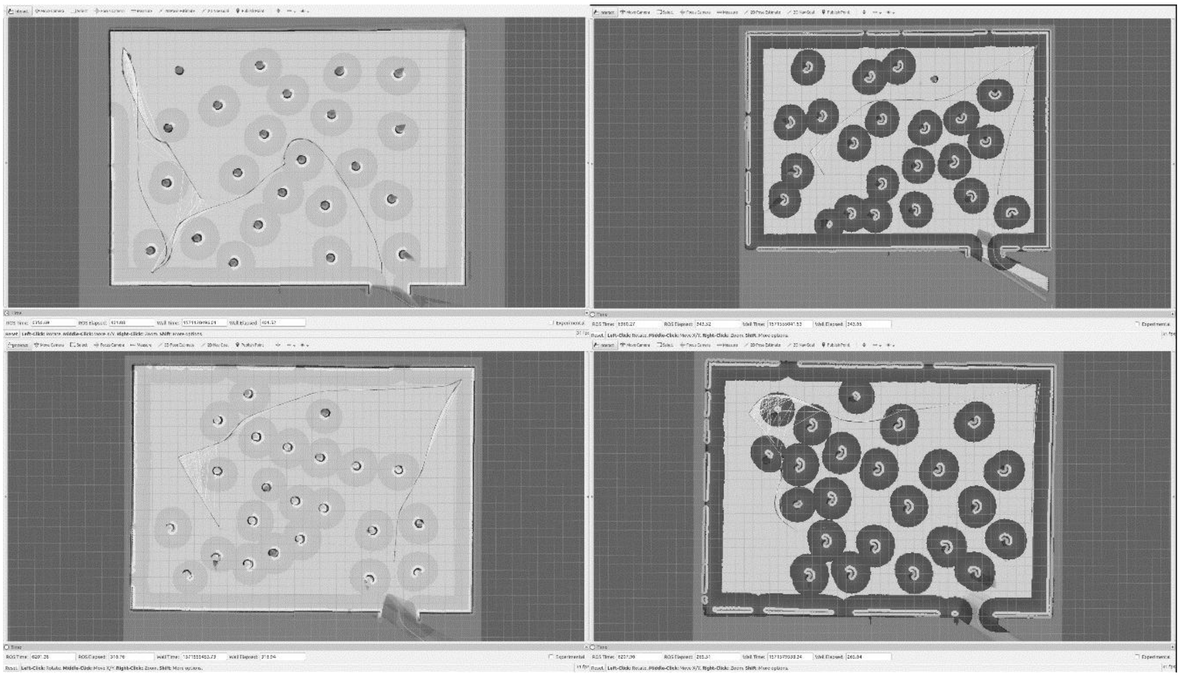

Fig 2. Successful passage of a virtual UAV training ground

In Figure 2, some areas are highlighted in bright color. Here the cost reaches maximum values, as this areas are in close proximity to the obstacle. This area is called lethal because the drone center will collide with it when approaching. Around the lethal area are visible areas with color gradients. The settings for the distribution of cost are regulated by the parameters given in table 2 [8].

Table 2. Some set values of move_base

Parameter Name	Value	Description
cluster_max_distance	1	The maximum distance between two occupied pixels in order to combine them into one obstacle, meters
cluster_min_pts	2	The minimum number of points required to create a cluster, the number
inflation_radius	2	The distance on the cost map in meters from the obstacle to the beginning of the zone of zero cells of the map, meters
cost_scaling_factor	5	The coefficient responsible for the rate of decrease in the value of the card cell to zero
neutral_cost	66	Value for neutral cells
cost_factor	0,7	Parameter affecting the smoothness of obstacle avoidance
lethal_cost	250	Lethal cell value

2.4 Drone Control Subsystem

The Offboard Controller library provides a high-level program abstraction, allowing the developer to operate the drone with high-level commands: follow to a specific point, hold a point or height, keep orientation to a specific point, and the like. Based on the received commands and data from the position calculator, the controller generates control actions on the autopilot. In addition, the library has the ability to dynamically connect the so-called activities to the controller: stand-alone program modules that allow you to make changes to the basic behavior of the controller. Therefore, for example, the library provides collision avoidance activity, which, when connected and correctly configured, allows changing control actions in such a way that the drone begins to avoid approaching obstacles and flies off to a safe distance from dynamic obstacles.

2.5 Flight Mission Subsystem

The flight mission is set as follows. A special node tracks changes in unexplored space. This node sets a new flight target for the route-planning algorithm whenever the drone flies to a previously set target at a certain distance, or if the drone hangs in place for a while, i.e. the correct path to the goal was not built. The location of the new target is chosen in such a way that it be outside the border of the rectangle, inside which the entire investigated area would fit, closer to one of the corners. The angle is selected clockwise or counterclockwise. The target window or door is recognized as the free space in and behind the line of obstacles, significantly expanding the investigated area.

4. RESULT

The drone used in this work is equipped with an RPLidar A1 all-round laser rangefinder with a frequency of 5 Hz and 360 measurements per revolution. The radius of measurements is 12 meters. The maximum range of a laser altimeter is 10 meters. Pixhawk with PX4 firmware is installed as an autopilot. NVidia Jetson is installed as an on-board computer with Ubuntu 16.4 and ROS installed. The inertial sensor module is integrated in Pixhawk.

As a result of the tests, the operability of the developed system was proved. According to the test results, in ninety-six percent of launches, the drone successfully found a way out of the room, flying around obstacles, as shown in Figure 2. The only reason for the failure of some test flights was the malfunction of inertial sensors.

Table 3. Virtual Tests Results

№	The essence of verification	Successfully	Unsuccessfully	Causes of Failures
1	We checked the stability of holding the orientation of the drone by accidentally moving around the polygon	2	3	Failure of inertial sensors during a sharp change of direction
2	The stability of the task of flying out of the room (four different configurations of obstacles inside) was tested, with 25 tests each	96	4	Google Cartographer crashes when the drone changing direction

4. CONCLUSION

A drone control system based on using technical vision methods for positioning has been developed. It allows to complete the autonomous fly task. The system uses data from on-board sensors and google cartographer and the move_base package as parts of itself. The effectiveness of the system is confirmed by tests on various types of virtual landfills. Thus, the developed system allows for the autonomous execution of a flight mission for a drone in conditions of loss of communication with the operator and GPS.

Acknowledgement

This work was supported by a grant from the President of Russia (Grant Number MD-2102.2019.9).

REFERENCE

[1] Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, Rafael Medina-Carnicer. "Speeded Up Detection of Squared Fiducial Markers" *Image and Vision Computing 76* – June 2018.

[2] J. Engel, V. Koltun, D. Cremers. "Direct Sparse Odometry." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* P. 17, 2018.

[3] R. Wang, M. Schwörer, D. Cremers. "Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras." *International Conference on Computer Vision (ICCV),* P. 9, 2017.

[4] W. Hess, D. Kohler, H. Rapp, D. Andor. "Real-Time Loop Closure in 2D LIDAR SLAM." *Robotics and Automation (ICRA), IEEE International Conference on. IEEE,* pp. 1271-1278, 2016.

[5] P. E. Hart, N. J. Nilsson, B. A. Raphael. "Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics SSC4.* - vol. 2, pp. 100-107, 1968.

[6] Stentz, Anthony. "Optimal and Efficient Path Planning for Partially-Known Environments." *Proceedings of the International Conference on Robotics and Automation,* pp. 3310-3317, 1994.

[7] Dijkstra E.W. "A Note on Two Problems in Connexion with Graphs" *Numer. Math – Springer Science + Bussiness Media*, vol. 1, Iss. 1, pp 269-271, 1959.

[8] Kaiyu Zheng. "ROS Navigation Tuning Guide." P 23, 2016.