10th INTERNATIONAL SCIENTIFIC CONFERENCE
ON DEFENSIVE TECHNOLOGIES
OTEH 2022
Belgrade, Serbia, 13 – 14 October 2022

AUTONOMOUS ROBOT EXPLORATION IN GAZEBO SIMULATOR

NOVAK ZAGRADJANIN
University of Belgrade, School of Electrical Engineering, Belgrade,
Material Resources Sector, Ministry of Defence, Belgrade, zagradjaninnovak@gmail.com

NIKOLA KNEZEVIC
University of Belgrade, School of Electrical Engineering, Belgrade, knezevic@etf.rs

KOSTA JOVANOVIC
University of Belgrade, School of Electrical Engineering, Belgrade, kostaj@etf.rs

BOJAN PAVKOVIC
Military Technical Institute, Belgrade, bjnpav@gmail.com

Abstract: *An autonomous robot is often in a situation to perform tasks or missions in an initially unknown environment. A logical approach to doing this implies discovering the environment by the incremental principle defined by the applied exploration strategy. A large number of exploration strategies apply the technique of selecting the next robot position between candidate locations on the frontier between the unknown and the known parts of the environment using the function that combines different criteria. In this paper, an architecture is proposed that in Gazebo, using ROS and Matlab, can test different exploration strategies in order to build map of the environment. Besides, proposed architecture allow improving the exploration strategy's adaptability to different situations.*

Keywords: *robot path planning, frontier-based exploration of environment, Gazebo, ROS.*

1. INTRODUCTION

Autonomous exploration of the environment is an important area of study in robotics. In the literature, it is most often defined as a combination of mapping of the environment (in which the robot moves) and robot path planning, Fig. 1 [1].

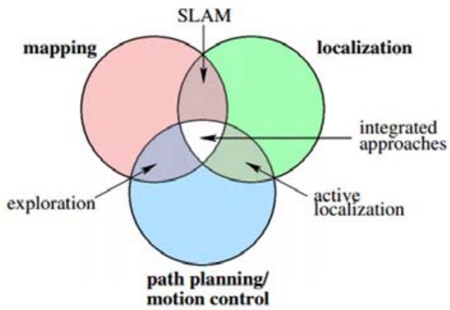


Figure 1: The basic tasks of the autonomous robot

Mapping is the problem of integrating the information gathered with the robot's sensors aiming to construct a map of an unknown environment. In order to form a map of the entire environment, it is necessary for the robot to go from one position to another (along a path defined by a planner) and from those positions to successively observe the environment, collect data and integrate them, until the task is completed. To perform this task reasonably, it is necessary to define an appropriate exploration strategy.

Generally, the environment exploration is not limited to the environment mapping, but is defined as a set of actions taken by an autonomous mobile robot in accordance with a previously defined strategy, with the aim to discover the characteristics of the environment that are of importance to the operation. Hence, exploration is the basis of numerous real-world applications of robotics, such as, in addition to the environment mapping [2], search and rescue operations [3], space missions, visual inspections, mining, robotic vacuum cleaners, etc. For example, in search and rescue operations, the goal is to locate injured persons or victims of natural disasters, fires or other accidents. There are frequent situations that in parallel or just before the basic operation, a mapping of the environment has to be performed, so that the operation can be accomplished. Similar scenarios of the exploration, combined with several types of tasks, can be seen in practice in other robotic applications.

Exploration can be broadly classified into two distinct approaches [4]. The first approach involves a prior knowledge of the environment, based on which off-line algorithms are used to define the exploration strategy. In this approach, the path of the robot is determined in advance (a predefined path). The second approach is applied when the environment is completely unknown or when there is not enough information for efficient application of off-line algorithms. In that case, the exploration is much more challenging and involves an online incremental exploration principle. If so, after collecting data from the current position, the robot selects the next position, moves to it, observes the environment from the new position and repeats this process until it completes the task or completes the operation [5]. The main problems, when using this exploration concept, is choosing the next position and planning the path. Most of the strategies for the exploration are based on choosing the next robot position out of the set of "candidate positions" that are on the frontiers between the explored free space and the unexplored parts of the environment, applying some criteria for their evaluation. These are so-called frontier-based strategies. The concept of frontier-based exploration of the environment was first officially presented in the *Yamauchi's* paper in 1997 [6].

Apart from frontier-based, other types of strategies are used to explore the environment. Some strategies, for example, involve human participation in the loop of choosing the next robot position, then applying a probability analysis of various benefits in the selection process between candidate positions or simply choosing the next position by random selection, etc. What is common to all classic environmental exploration strategies is that they aim to do the exploration in the shortest possible time or in the shortest possible total distance traveled by the robot.

Fig. 2. shows a Carnegie Mellon University study that explains that frontier-based strategies perform better than the other two analyzed types of environmental exploration strategies.

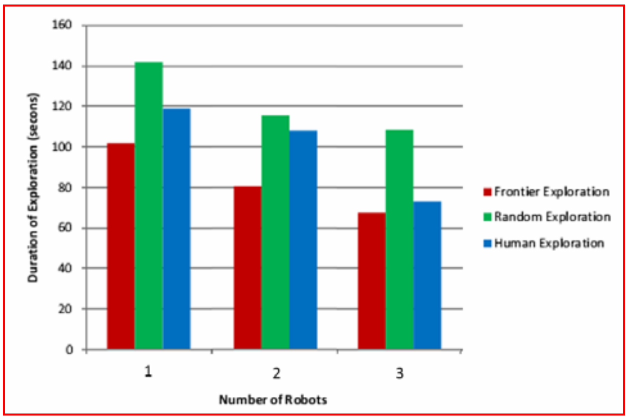


Figure 2: Different concepts of exploration

2. THE CONCEPT OF FRONTIER-BASED EXPLORATION

As already stated in Section 1, most exploration strategies belong to the group of frontier-based family. Even today it is one of the basic directions of research in the field of autonomous exploration of the environment.

The main steps of frontier-based exploration are presented in Fig. 3 and can be defined as follows [7]:

1. The selection of the next observation location according to an exploration strategy;
2. The reaching of the observation location selected in previous step. This step requires the planning and following a path, that goes from the robot's current position to the chosen location;
3. The acquisition of a partial map from the observation location, using data collected by the robot's sensors;
4. The integration of the partial map within the global map.

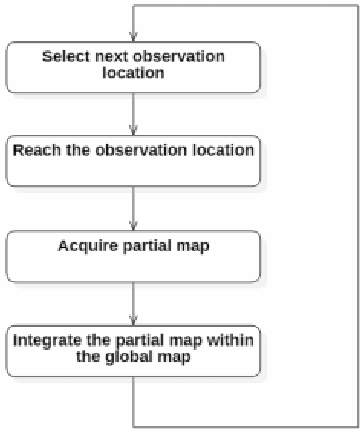


Figure 3: The main steps of the exploration process

The frontier determination techniques usually implies the environment representation in the form of an occupancy grid map. The boundary of the robot's field of view defined by the range of its sensors can be divided into free, obstacle and frontier arcs, Fig. 4. [8]. Free arcs are parts of the border to the explored part of the environment, whereas obstacle arcs are parts of the border to locally detected obstacles. Any arc that is neither obstacle nor free is a frontier arc and is, in fact, part of the border to the unexplored part of the environment. The fact that each frontier arc is actually one frontier can be adopted. Allow us also to take that the middle point of each frontier represents one candidate for the next robot position, which is a common approach in the papers [4,5,8] dealing with this topic.

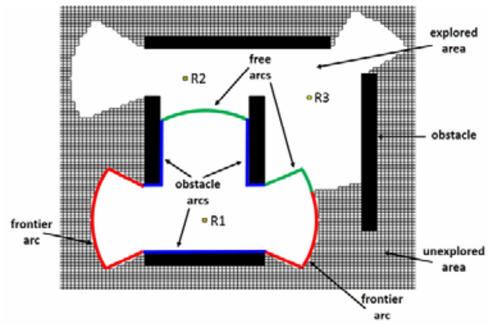


Figure 4: Free, frontier and obstacle arcs in exploration

When the robot reaches the selected location, it observes the surroundings, improves the knowledge of the environment and also updates the frontier list [8].

3. FRONTIER-BASED EXLORATION STRATEGIES

To evaluate the candidate p in order to select the next robot position in frontier-based exploration strategies, different criteria are proposed in the literature. The following criteria are most commonly used [9]:

- $L(p)$, the minimum length of a collision free path or the minimum path cost from the current robot position to r usually calculated by using a path planner,

- $A(p)$, the expected information gain obtained by simulating the robot's perception from the location p; it is calculated or estimated based on the size of the unexplored area that would be explored from that location, given the current status of the occupancy grid map and the robot's sensing range,
- $P(p)$, the probability that the robot, if it arrives at the location p, will be able to communicate with the base station and send the collected data; this probability generally directly depends on the distance of p from the base station.

- Classic exploration strategies

There are various approaches using one of the above-mentioned criteria or a larger number of them in the form of the function that uniquely describes each candidate p. The classic exploration strategies are are summarized in [9].

The first strategy is trivial and it chooses for the next robot position the one up to which the path planner, in relation to the current robot position, calculated the minimum length or cost of the path, counting all candidate positions. It is usually denoted by Dist_Min.
In the next strategy, $L(p)$ is combined with $A(p)$ in the form of the linear function (1):

$$u(p) = A(p) - L(p) \tag{1}$$

The parameter β regulates the influence of the criterion $L(p)$ versus $A(p)$.

Another approach describes the candidate p as the following exponential function (2):

$$u(p) = A(p) \cdot \exp(-\lambda \cdot L(p)) \tag{2}$$

The parameter λ is greater than zero and weights both included criteria. This strategy is named after the authors as the GBL strategy. The GBL strategy is most often proposed in the literature as a good enough choice. It can be considered as a classic strategy and has been used as a reference strategy in a significant part of the relevant literature that studies the subject area [8,9].

These exploration approaches are mainly used for the map building process. In this process and especially in the case of search and rescue missions, however, it is usually important to introduce a criterion related to the probability of establishing communication between the robot from the location p and the base station ($P(p)$), so that the information can be forwarded as soon as possible to be further used. The introduction of this criterion was proposed in the form of the following function (3) [9]:

$$u(p) = (A(p) \cdot P(p)) / (L(p)) \tag{3}$$

On the other hand, different additional criteria for the selection of the next robot position are proposed in the literature. For example, the overlap ($O(p)$) of the current environment map and the part of the environment visible from p can be proposed as an additional criterion [8,9]. In addition to the path cost, the criteria such as the recognition of the uncertainty of a landmark, the number of the features visible from the location, the length of the visible free edges, and the number of the rotations and stops required from the robot to reach a location can be considered, also [8,9]. Criteria selection depends on the mission specifics and the exploration goals. In this context, the introduction of the criteria that (if possible) will take into account the types of facilities for each candidate location is proposed in the paper [5] (which considers exploration in search and rescue missions) in order to initially direct a mission to the residential area, where the largest number of the victims of a disaster are logically expected.

- MCDM-based exploration strategies

Taking into account the above-mentioned, a more recent direction of research in this area includes the implementation of different MCDM methods in MCDM-based exploration strategies. MCDM provides a broad and flexible approach to the selection of the utility function that can be used to evaluate candidates for the next observation location. In [4,9], for example, the Choquet fuzzy integral is proposed. This approach enables a researcher to take into account the relative relationship between criteria, such as redundancy and synergy, which is its main characteristic. The experimental results in those papers show that respectable results are obtained by using MCDM-based exploration strategies compared to the classic exploration strategies. In [5], the proposed approach to the selection of the next location from a set of candidates within the exploration strategy uses a standard MCDM method—PROMETHEE II. Here, an attempt is made to take advantage of the characteristic of this method referring to the fact that, in addition to weights, preference functions are used as additional information for each criterion. Autonomous robot navigation strategy based on MCDM Additive Ratio ASsessment (ARAS) method is proposed in [10]. The greedy area exploration approach is suggested, while the criteria list consists of: battery consumption rate, probability to collide with other objects, probability to yaw from course, probability to drive through doors, probability to gain new information, length of the minimum collision-free path. In [11], the implementation of the on-line MCDM-based exploration strategy that exploits the inaccurate knowledge of the environment (information obtainable from a floor plan) is proposed. The results of the experiment show that the proposed approach has a better performance in different types of environments with respect to the strategy without prior information. Although the use of more accurate prior information leads to a significant improvement in performance, the use of inaccurate prior information could lead to certain advantages also, managing to reach the high percentages of the explored area travelling a shorter distance, with respect to the strategy not using any prior information.

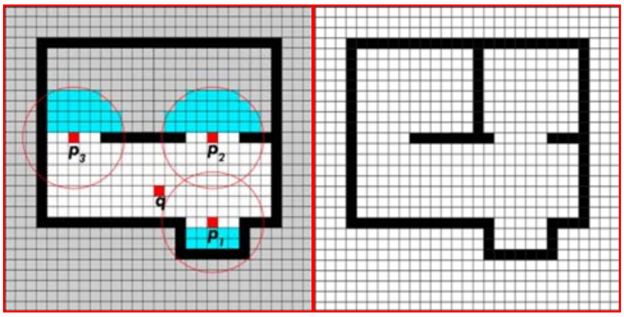
This paper generally belongs to the above-mentioned direction of autonomous exploration research, bearing in mind the fact that this issue is not sufficiently considered in the existing literature.

The exploration strategies based on MCDM in this paper are using the standard SAW, COPRAS and TOPSIS methods proposed in [9].

- Determination of criteria values

The criteria $L(p)$, $A(p)$ and $P(p)$ are used for candidate selection in this paper. To determine the value of criterion $L(p)$, the powerful D* Lite algorithm is used [8,12]. The criterion value $P(p)$ illustrates the Euclidean distance p from the base station [8,9].

The value of the criterion $A(p)$, which actually illustrates the pre-estimated gain of data about the environment that the robot can "see" with its sensors if it reaches the position p, is calculated in the manner described below [11]. How candidate positions are defined in the frontier-based environment exploration is described in Section 2. In order to evaluate the visibility of the point s with the robot sensor of the range r from the point p (in the case of positioning the robot at the point p), it is necessary to check whether the point s belongs to the corresponding line of sight. Fig. 5. illustrates the assessment of information potential for three different candidate positions.



a) b)

Figure 5: Evaluation of the information potential value (cells marked in blue) of candidate positions $p1$, $p2$ and $p3$. The current position of the robot is q (*a)* map of the explored part of the environment, *b)* real map of the environment).

In Fig. 5. it can be seen that there was an error in the assessment of the informative potential for the candidate position $p2$, because in the unexplored part of the environment there is another partition wall which, if the robot chooses that position, will reduce the actual perception by the sensor compared to the assessment performed based on its maximum ability.

4. MAP BUILDING EXPLORATION IN GAZEBO 3D SIMULATOR

Gazebo is a specialized environment for 3D robot simulation. It enables credible testing of robots and their activities in 3D scenario simulations an can be used as a plausible substitute for testing on real robots.

In the present case, the virtual model of *Turtlebot 3* robot was used. The specified robot model for mapping environment as a sensor used *LIDAR LDS-01*, range 3.5m and angle resolution 1°. Packages operating under Robot Operating System (ROS) were used to control the robot, odometry and SLAM function, i.e. to integrate data collected with a robot sensor, forming a map of the environment and simultaneously determining the position of the robot in relation to that map. ROS provides a collection of tools and libraries for the development and optimization of robotic systems, which is especially suitable in combination with other specialized software packages intended for robotics such as *Gazebo*, but also in combination with *Matlab*, etc.

Notably, the following packages were used in this approach: *turtlebot3_gazebo* and *turtlebot3_gmapping*. these packages use the following types of messages: */odom* (for information on the robot's position obtained by odometry), */scan* (for information collected from *lidar* sensor), */map* (data to map the environment), */cmd_vel* (linear and rotational speed of the robot), */tf* (describes the relationship between the coordinate system of maps and coordinate systems of the robot), */joint_states* (information on robotic actuators), */hack_scan* (modified / */scan* data). The ROS architecture used with the nodes displayed, the package names and the data flows is shown in Fig. 6.

Gazebo communicates as a simulator with ROS, and ROS sends data about the map to Matlab where strategies are executed for the selection of the following exploration position and planning robots to that position. After that, the algorithm was performed in Matlab, which regulates the control of the robot movement to monitor the calculated path. Specific challenges represented the following:

- Adjusting the map view from ROS in Matlab in order to comply;

- Customize the map from Matlab to select the next robot position and planning robot paths. It was necessary to form a buffer zone (several rows of artificially blocked fields) around obstacles to prevent physical contact of the robot with the same;

- Adaptation of LIDAR data so that the algorithm can do the mapping of open space according to line-of-sight. The physical sensor is designed in such a way, when an obstacle is much further than its range, the feedback signal is not registered. For such situations, it is defined that there are no obstacles in the range of sensors.

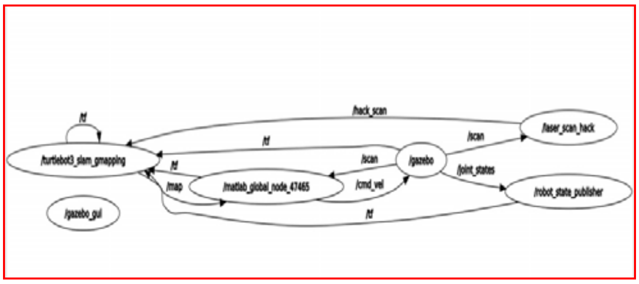


Figure 6: The ROS architecture using nodes, package names and data streams

For testing, two test environments were created in Gazebo. The first environment (environment A, Fig. 7.) had dimensions 10mx10m, while the second environment (environment B, Fig. 8.) is more complex and had dimensions 30mx30m. The both environments were 2D discretized into 0.2mx0.2m cells.

All the MCDM methods were tested with two combinations of criteria weights: 0.7, 0.2, 0.1 (Strategy 1) and 0.5, 0.4, 0.1 (Strategy 2), for the criteria $L(p)$, $A(p)$ and $P(p)$, respectively. The Strategy 2 can be characterized as more aggressive, because it gives more importance to the criterion of the expected information gain, forcing the robot to take more risks and to travel greater distances in order to explore more space, regardless of the complexity of the environment [8]. By changing the weights during exploration, we can switch between different behaviors, varying the criteria's importance that drive robot decisions. This is a significant advantage of MCDM-based exploration strategies over other approaches. In order to test different approaches, GBL strategy was also tested (expression (2), where λ = 0.2, the same value reported in the papers [8,9]).

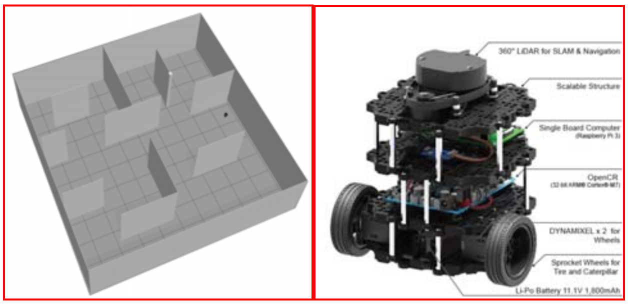


Figure 7: Map A created in Gazebo simulator for the purposes of testing exploration strategies, the black dot illustrates the robot (left side). Turtlebot 3 robot and its basic components (right side).

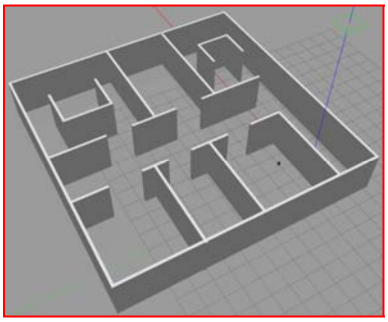


Figure 8: Test environment B created in Gazebo simulator for the purposes of testing exploration strategies.

The exploration results using the TOPSIS 1, COPRAS 2 and GBL strategies, the corresponding robot paths generated by the D* Lite algorithm in the test environment A for the starting location x=13 and y=7, are shown in Fig. 9, 10 and 11, respectively.

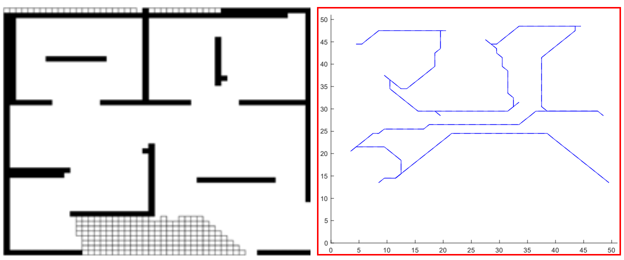


Figure 9: Map A built after exploration and the robot path generated by the D* Lite algorithm in the Gazebo with the implemented COPRAS 2 exploration strategy.

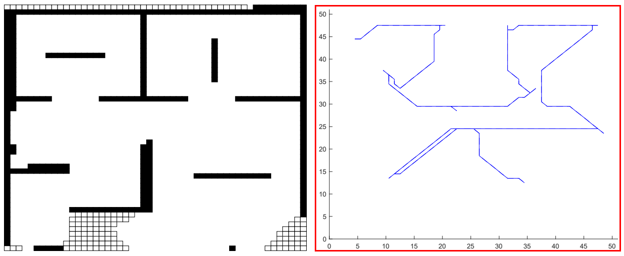


Figure 10: Map A built after exploration and the robot path generated by the D* Lite algorithm in the Gazebo with the implemented TOPSIS 1 exploration strategy.

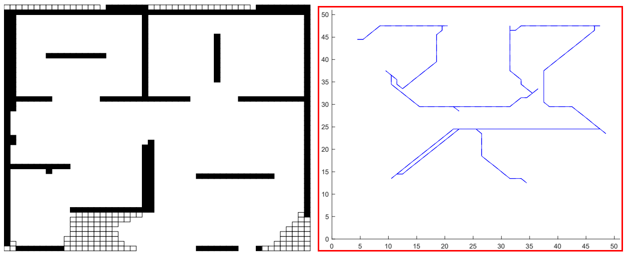


Figure 11: Map A built after exploration and the robot path generated by the D* Lite algorithm in the Gazebo with the implemented GBL exploration strategy.

The exploration results using the TOPSIS 1, COPRAS 2 and GBL strategies, the corresponding robot paths generated by the D* Lite algorithm in the test environment B for the starting location x=7 and y=7, are shown in Fig. 12, 13 and 14, respectively.

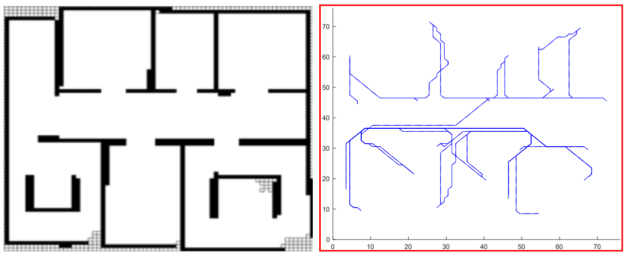


Figure 12: Map B built after exploration and the robot path generated by the D* Lite algorithm in the Gazebo with the implemented TOPSIS 1 exploration strategy.

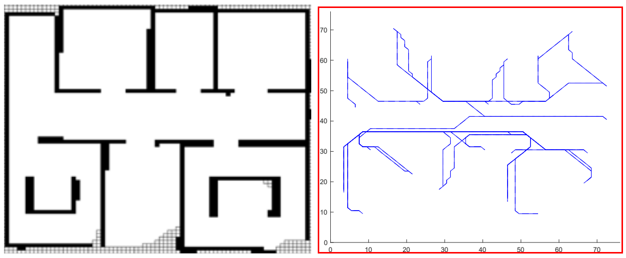


Figure 13: Map B built after exploration and the robot path generated by the D* Lite algorithm in the Gazebo with the implemented COPRAS 2 exploration strategy.

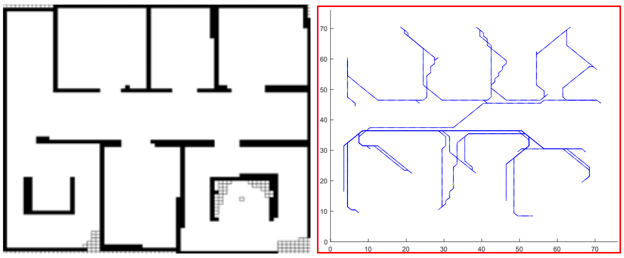


Figure 14: Map B built after exploration and the robot path generated by the D* Lite algorithm in the Gazebo with the implemented GBL exploration strategy.

Table 1. The average exploration results in terms of the travelled distances for the environment

Exploration strategy		Travelled distance	
		Environm. A	Environm. B
MCDM	SAW 1	251.45	1.097,2
	SAW 2	258.36	1.102,1
	COPRAS 1	240,29	1.050,8
	COPRAS 2	247,52	1.078,9
	TOPSIS 1	216,52	1.024,3
	TOPSIS 2	218,93	1.036,1
GBL		216,52	1.032,1

Analyzing Table 1, the TOPSIS method had the best results, but additional research is needed, bearing in mind that the results also depend on the starting positions [8].

5. CONCLUSION

In paper, an architecture is proposed that in Gazebo, using ROS and Matlab, can test different strategies for exploration in order to build map of the environment. It can also be used to determine the optimal mission plan to improve the exploration strategy's adaptability to different situations (limited exploration time, the need to change the exploration direction in different parts of environment during the mission, etc.).

References

[1] C. Stachniss, "Robotic mapping and exploration," Springer, 2009.

[2] C. Gomez, A. C. Hernandez and R. Barber, "Topological frontier-based exploration and map-building using semantic information," Sensors, vol. 19, no. 20, pp. 4595, 2019.

[3] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, O. von Stryk et al., "Hector open source modules for autonomous mapping and navigation with rescue robots," in RoboCup: Robot World Cup XVII, ser. Lecture Notes in Artificial Intelligence (LNAI). Berlin: Springer, pp. 624–631, 2013.

[4] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for an autonomous mobile robot," in Proceedings of the 4th European Conf. on Mobile Robots, Mlini/Dubrovnik, Croatia, pp. 259–264, 2009.

[5] P. Taillandier and S. Stinckwich, "Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots," in Proceedings of the IEEE Int. Sym. on Safety, Security, and Rescue Robotics, Kyoto, Japan, pp. 321–326, 2011.

[6] [6] B. Yamauchi, "A frontier-based approach for autonomous exploration," in Proceedings of the IEEE Int. Sym. On Computational Intelligence in Robotics and Automation, Monterey, California, USA, pp. 146–151, 1997.

[7] M. Kulich, T. Juchelka and L. Preucil, "Comparison of exploration strategies for multi-robot search," Acta Polytechnica, vol. 55, no. 3, pp. 162–168, 2015.

[8] N. Zagradjanin, D. Pamucar, K. Jovanovic, N. Knezevic and B. Pavkovic, "Autonomous exploration based on multi-criteria decision-making and using D* Lite algorithm," Intelligent Automation & Soft Computing, Special Issue: Soft Computing Methods for Intelligent Automation Systems, pp. 1369-1386, 2021.

[9] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for search and rescue autonomous robots," in Proceedings of the 10th Int. Conf. on Autonomous Agents and Multiagent Systems, Taipei, Taiwan, pp. 99–106, 2011.

[10] R. Semenas and R. Bausys, "Autonomous navigation in the robots' local space by multi criteria decision making," Open Conf. of Electrical, Electronic and Information Sciences, Vilnius, Lithuania, pp. 1–6, 2018.

[11] M. Luperto, D. Fusi, N. Alberto Borghese and F. Amigoni, "Exploiting in accurate a priori knowledge in robot exploration," in Proceedings of the 18th Int. Conf. on Autonomous Agents and MultiAgent Systems, Montreal, Quebec, Canada, pp. 2102–2104, 2019.

[12] S. Koenig and M. Likhachev, "D* Lite," in Proceedings of the Eighteenth National Conf. on

[13] Artificial Intelligence, Edmonton, Canada, pp. 476–483, 2002.