Autonomous Robotic Exploration and Navigation System using Tri-layered Mapping and Geometrical Path Planning Techniques

S Gowtham¹, R Praveen¹, M Parthiban¹, P Sai Charan¹, N Seenu² and RM Kuppan Chetty²

¹Mechatronics Engineering, School of Mechanical Sciences, Hindustan Institute of Technology and Science, Chennai – 603103.
²Centre for Automation and Robotics, School of Mechanical Sciences, Hindustan Institute of Technology and Science, Chennai – 603103.

Abstract. In the recent decade, robotic exploration is emerging as a vast research platform as it prolongs to be a challenging task, especially in unknown dynamic environments. Exploration plays a vital role in collecting data from the areas inaccessible to the human. The operation of manual exploration systems in complex environments is cumbersome due to ineffective noise filtering, poor visibility and shuffling of map characteristics. In order to overcome such drawbacks, our approach focuses on the development of autonomous robotic exploration by designing a robust SLAM-based system, incorporating tri-layered mapping technique for automatic map generation and a new geometrical boundary-based path-planning technique to improve the mobility of the robot during navigation. The proposed path-planning algorithm yields better trajectory planning and obstacle avoidance using the pose, odometry and orientation parameters of the robot. The performance of the developed robot is evaluated through real-time experimentation, and it is evident from the results that, the proposed approach is preferable for autonomous exploration applications.

Keywords. Mobile robot, autonomous robotic exploration, SLAM, tri-layered mapping, geometrical boundary-based path planning, autonomous navigation.

1. Introduction

Autonomous mobile robots have enormous applications. A few valuable mentions include space exploration, underwater robot exploration, military surveillance, AI security patrol services and material transportation. In large-scale environments, practical exploration requires complete area coverage within a surrounding to acquire useful intel. This notion necessitates the need for unmanned surveillance robots for broader exploration. The autonomous nature of such robots is essential to improve work efficiency by reducing user-reliance. Autonomous exploration robot is to be capable of planning its path in order to move in and around an area [1] automatically. A navigation system is
embedded in the robot so that it can identify its position. When an obstacle blocks the robot, the navigation algorithm employed in the robot allows it to reset and resume its course, once the stopping factors have been eliminated [2]. Map of the environment is a pre-requisite for path planning and navigation processes. However, an exploration environment does not comprise its map in prior. SLAM Simultaneous Localization and Mapping technology are used to construct the map for an unknown environment. Further, the map is used for navigation.

Various studies give an insight into the working principles and component selection for applications of SLAM-based robots [3]. For instance, mapping an area of sixteen square meters has been performed with the help of Microsoft Kinect [4] motion sensor using laser data. A four-wheeled mobile robot is made capable of self-navigation using SLAM algorithms. Research gives an insight into the advancements and specialization in the model used for localization and navigation for a restaurant service robot [5]. Depth cameras and low-cost lasers have been used instead of heavy and expensive laser sensors. Experiments that have been conducted in different environments indicate that this approach has good performance in complex indoor environments enabling the robot to be capable of restaurant services. It is seen to sense an incoming obstacle such as static objects and humans in motion. In such cases, the robot is seen to efficiently avoid the former factors by avoidance and positioning itself under a table [6]. Mapping and navigation simulation on mathematical modelling of indoor navigation is performed where navigation technology is implemented in a wheelchair for older adults [7]. Improved SLAM technology for mapping results in a better contrast of map-feature inference and navigation stack is eligible in enabling the robot to navigate at the shortest distance possible [8]. It is inferred that; constructing maps of indoor environments or relatively flat surfaces involve fewer complications than constructing the same on irregular terrains [9].

The challenges faced in existing conventional mapping techniques involve, acquisition of spatial models of an environment [10]. As robots are led into unknown areas, the varying factors in the surrounding poses difficulty in measurements as errors keep building up over time [11]. For instance, errors found in odometer data tend to cause further variations in detecting other motion parameters. Data association tends to become complicated when a robot consists of a wide array of sensors integrated within the hardware setup. Occupancy grid mapping renders in-effective in the presence of external disturbances such as noise [12]. In such cases, the mapping consistency is compromised. The existing mapping techniques employed in SLAM-based robots are only set to scan a single layer of an area leading to shuffling and low visibility of map characteristics such as obstacles and borders. The proposed methodology aims to overcome the drawbacks in existing map-construction techniques in a SLAM-based robot to improve localization and other characteristics.

The proposed method employs a tri-layer mapping technique which builds a stack comprising three layers, inclusive of the live feed from obstacles in the course. The proposed algorithm automatically reconstructs the map by building consecutive layers and simultaneously discarding the previously built layers, such that three layers are stacked within the generated map at any given point of time. A better obstacle avoidance compared with conventional mapping techniques is inferred. This study presents a comparison between conventional mapping methods and the proposed tri-layer mapping method, demonstrating the effectiveness of the latter. Improvements have been made to the existing path planning technique to operate the robot such that, navigation can be accomplished with the shortest set of coordinates and angles. The proposed approach compiles a breakthrough in the implementation of tri-layered mapping and geometrical path planning techniques.

2. Tri-layered Mapping
The proposed method aims to improve conventional single-layered mapping techniques. The improvements incorporated in the algorithm is set to determine the displacements between the robot and the obstacles automatically, the shortest route obtained, increases the robot’s mobility and response in closely enclosed areas. Our approach is referred to as, tri-layered mapping technique. The concept of the tri-layered mapping is illustrated in Figure 1. The laser light transmitted from the rotating turret is used to acquire map data to be generated in three layers, which are then stacked over
one another in order to form a map of a surface. The next layer generated will automatically discard the first layer that was scanned in the previous scan cycle. The map generation process leaves three layers to be intact at any given instant of time. This technique is adopted to impart better stability to the map building process. As seen in the single-layered mapping, the dimensions are misaligned due to weak laser signal transmission. The proposed tri-layered mapping technique uses the application of robot odometry by fine-tuning the PID controller with the help of the RQT framework.

![Figure 1. Block diagram of proposed Tri-layered mapping](image)

2.1 Static and dynamic mapping
Mapping is differentiated into two types, namely: static and dynamic. Static mapping refers to the robot being stationary at one position, whereas the environment keeps changing. In dynamic mapping, both the robot as well as the environment, are factors of variation. In figure 1, the static and dynamic modes of the mapping are presented. In static mode, map generation in three layers refers to the insertion and deletion of consecutive layers simultaneously. For instance, the fourth layer added will discard the first layer generated in an area. In this way, a map is reconstructed in the static type. As the robot is kept static and surrounding conditions are varied, obstacles that block the path should be noted in the map for better navigation.

In the case of dynamic mapping, the robot’s motion is considered along with variations in environmental features. As the robot moves from one room to another the dynamic characteristics are taken into consideration, such that the first layer scanned, will be preserved for reference and the third layer of the current scan cycle will be discarded. This aspect poses a critical difference between static and dynamic techniques. As seen in the latter, the first two scanned layers will be used as a reference for map generation.
2.2 Comparison between the conventional single-layered mapping and tri-layered mapping

In a conventional single-layered mapping technique, the map reconstruction is difficult when it has been collapsed due to the failure or displacement of IMU sensors. Because there exists a single layer of map data [13]. Error in IMU or odometry can easily affect the conventional mapping [14]. Whereas, in tri-layered mapping, errors are sorted out with no loss in data, provided, the map is reconstructed using the layers scanned at the beginning. Conventional mapping does not have appreciable navigation response as in the dynamic mode of operation; the obstacle positions are continuously varying. The proposed mapping technique effectively determines the position and direction parameters of the obstacles while significantly reducing the effects of noise during the map building process. Table 1 consolidates the major comparison concerns between single and tri-layered mapping techniques.

Table 1: The differences inferred between the conventional mapping technique and the proposed tri-layered mapping technique.

| Conventional mapping                                                                 | Tri-layered mapping                                                                 |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 1. The conventional mapping method consists of a single layer scan [13]. It lacks mapping accuracy in uneven ground surfaces due to shuffling of map characteristics during a few intervals of time. | 1. Tri-layered mapping generates a map of three layers stacked upon one another to prevent shuffling of map characteristics at any instant of time. |
| 2. It is comparatively ineffective to the proposed methodology based on navigation. The map cannot be reconstructed [14] once the sample points are shuffled and scattered. | 2. Due to better viewing angle and usage of PID controller for odometry, it has excellent efficiency on navigation. If the map starts to scatter, it can be easily reconstructed. |
| 3. Obstacle representation and various factors like offset in wheel alignments or undesired scattering of the map due to acquisition of inappropriate IMU values. | 3. The sample points generated in this map is not scattered due to the layer-stacking countermeasure incorporated into it. |
| 4. Pre-existing navigation algorithms will instead deem ineffective, due to its lack of accuracy and reliability of the generated map. | 4. Modified navigation is implemented for this tri-layered mapping |
The proposed tri-layered mapping algorithm is implemented, and an environment for the robot has been fed to provide navigation. In order to move the robot autonomously, a navigation stack is proposed in this paper. The schematic working of the proposed path planning approach is described in the next section.

3. Geometrical Path Planning and Navigation Technique

The proposed method for navigation is set to improve the existing navigation, which is widely being used. Conventional path planning uses the Cartesian coordinate system, whereas, the Geometrical path planning technique is set to calculate minimal angular dimensions and its corresponding coordinates simultaneously with respect to successive obstacles to reach the destination.

![Figure 2](image_url)  
**Figure 2** Concept of the virtual circle based path planning

It uses geometrical virtual circle in order to shrink the considerable area for path planning. The angular dimension which provides the shortest distance between the robot and the target positions is estimated from the map LiDAR data. The concept of the proposed path planning is illustrated in Figure 2. The virtual circle is generated with the straight line distance between the robot and the target positions. The LiDAR sample pulses are shown in Figure 3, based on the acquired sample pulses the occurrence of obstacle(s) is/are identified, and then the path is planned in such a way that the path has more space than the robot’s footprint dimension.

![Figure 3](image_url)  
**Figure 3**: Methodology used in geometrical path planning and navigation technique
At the same time, the shortest distance position yielding LIDAR pulse angle is determined by ensuring a 10° angular distance away from the boundary of the obstacle. Another example of proposed path planning is illustrated in Figure 3. Figure 3 (a, b & c) demonstrates the step-by-step estimation of angular dimensions by using the proposed geometrical path panning concept. Figure 2(a) represents the initial start position and the destination position of the robot in an uneven box-shaped environment. The circle represents the virtual circle plotted with the radius of the straight line distance between the robot and the destination point. The blue line represents the calculated angle for reaching the destination along with the shortest distance. The angular dimension is estimated by accumulating 10° with the corner point angle of the obstacle. In the case of figure 3a, the estimated angular dimension is 66.5° in such a scenario; the robot can correct its odometry to accurately trace the path between the robot and the corner of the obstacle. Figure 3(b) represents the next identified best angular dimension to reach the target in the shortest distance. If the robot is oriented straight with respect to the target, the calculated angular dimension is 0°. Figure (c) represents the next shortest distance angle (0°) to reach the target position.

The navigation stack is solely developed with the help of ROS packages. If the robot has the generated map of the environment, the SLAM-based wheeled mobile robot can navigate to its destination with ease using the calculated parameters. The advantages of Geometrical path planning over conventional path planning are listed below in tabular format.

### Table 2: Difference between conventional and geometrical path planning and navigation technique

| Conventional path planning | Geometrical path planning |
|---------------------------|---------------------------|
| Conventional path planning uses coordinates to move the base to the destination; this approach has less accuracy compared to the geometrical path planning technique [15]. | Geometrical path planning uses angular dimensions and coordinates for navigation which uses geometry position, which means odometry, base footprint, IMU of the robot and shortest angle deviation required to reach the destination. |
| Due to the usage of a coordinate system without angle measurement in 2D-mapping, this approach will lead to wobbling in motion during path tracing. Navigation requires higher set precisions, about 0.001 for PID values [15]. | Virtual circle (Figure 2) creates a radius from the centre point of the robot to the destination point given in the 2D map. Adopting the virtual circle technique causes PID to consume significantly less amount of values with an accuracy of 0.01. |
| Path-planning of the robot will be varied according to the coordinates generated continuously taking into account of dynamic variables. Thus, the time consumption for navigating from the start point to the destination is high [16]. | To sum up, the adopted technique calculates the shortest angle and the nearest distance to the destination in a comparatively lesser time. |

The robot uses a set of encoder motors to get coordinate values and LIDAR data to know the range of destination as well as to localize itself in an unknown environment. There are some aspects when it comes to navigation trajectory, robot footprint, goal tolerance, obstacles and optimization, et cetera. Geometrical path planning uses acceleration and velocity values to coordinate itself and for avoiding obstacle minimum obstacle avoidance is set in the param file. Before inculcating navigation, PID configuration, odometry calibration, and the EKF (Extended Kalman Filter) have been applied.

### 3.1 RQT framework

RQT is a software framework pre-built in ROS. It is a software segment incorporated in this method to enable GUI (Graphical User Interface) tool development. This process is considered essential so that the graphics processing segment within the workstation can let the user visualize finer details of point cloud data perceived from the surroundings during the map generation process. The RQT plug-in used in the ROS environment to facilitate graphics processing as the Raspberry Pi3 cannot solely process
the entire workload, which is considered to be the acquisition unit to collect the scanned data samples from the RPLIDAR in binary format. Before launching the framework, it is essential to create perspective files for PID. The data obtained by driving the robot using teleoperation twist keyboard will be given in RQT, which is correspondingly reflected in the library file of the robot. Furthermore, the PID has been tuned until the errors have been eliminated.

3.2 Odometry of the robot
The odometry aspect reveals the rate of change in position and angular deviation of a robot over a given time. Implementing this feature provides the user with accurate values over short distances. Several processes can influence odometry; it involves encoder value, accelerometer value and EKF. Odometry errors tend to accumulate over time due to slippage in wheels and numerical integration error. Errors can either be found in circuitry or within the hardware components of the robot. The wheel displacements are first measured with the help of encoders in-built in the motor. The wheel velocity is assumed to be constant since the last encoder readings are taken at the time, \( \Delta T = 0 \). The chassis planar twist and calibration factor ‘c’ is then determined. It is essential to calibrate the components and troubleshoot the errors before getting started with experimentation. Odometry will be shown as trajectory curves (Figure 7) at which side the robot is moving; in this manner, the error will be sorted out in this segment.

3.3 EKF (Extended Kalman Filter)
The nature of the system influences the selection of the appropriate filter. If the system uses linearity principles based on time, the output expected from the system is not much of a complication as the user is aware that unknown factors are not present to influence such systems [17]. One such example of a filter is a regular Kalman filter [18]. This filter holds well with linear-systems that does not deal with much complication. In this paper, the extended version of the EKF (Extended Kalman filter) is used.

![Figure 3: Flow chart of EKF (Extended Kalman Filter)](image)

This concept is an extension of the existing Kalman filter to non-linear systems. Since the SLAM-concept which includes autonomous path planning, localization and navigation involves various unknown variables that can change constantly with time, it becomes a system that should be able to
cope up with such uncertainties. For such non-linear systems, EKF is used. Figure 3 describes the operation flow of the Extended Kalman Filter utilized for localization. One of the main advantages of the geometrical path planning technique is that the robot will start moving before map acquisition as the destination point is specified before the collection of data samples. The virtual circle plotted by the algorithm using a set of particle filters will ensure that the shortest coordinates and angular dimensions among consecutive obstacles are determined. Visualization in Rviz platform infers that the reference vectors adapt its course according to the motion of the robot whose time intervals are almost identical to the existing approach. Furthermore, the angular dimension is an additional variable, determined simultaneously to provide quicker responses to local localization to yield better navigation compared to conventional path planning that holds its current state until its position and odometry is being published. Errors that occur during global localization might prevent the robot not to reach its destination in conventional path planning. In geometrical path planning, if map knowledge is provided to the robot, it can reach its destination using the shortest path compared to conventional path planning technique.

3.4 Robot navigation in a known environment

The map generated before exploration of known environments is stored in the robot’s memory unit. In this case, navigation is made quicker with improved efficiency as the static features such as walls, boundaries, corners, and heavy objects remain stationary. The robot is made to adjust its course automatically to change in position of obstacles. Path tracing in a known environment is predetermined due to the map’s availability; the process is made quicker due to the stability of the navigation stack. The velocity and acceleration values are obtained from the accelerometer and the encoders in-built in the robot. By corrections in the PID controller as well as using IMU and pose-twist results from the robot using EKF filter, the robot is made to navigate in a known environment without wobbling or getting off course. The endpoints of obstacles are scanned using the LIDAR whose external protrusions if any are calculated relative with the robot’s motion. A virtual circle is bound upon every obstacle to find the shortest route by determining the shortest angular dimensions, minimal angle of over 10 degrees is set, and taking into clearance consideration as the robot must not collide with the obstacle. In this way, navigation time consumption is minimized with improved tracing efficiency.

3.5 Robot navigation in an unknown environment

Our proposed methodology aims to improve navigation in unknown environments in order to overcome the drawbacks seen in map shuffling and data association in existing techniques. Geometrical path planning embedded into the ROS environment functions uses libraries added in separate segments as PID and odometry values. The code present in each segment intermittently generates a set of parameters as the robot progresses. A reference point is first set to initialize the exploration operation. A sample area of a current position of the robot is fed as a map. Due to motion in an unpredictable environment, obstacle avoidance is incorporated into the robot, which is set to adapt along with the changing position of obstacles. The proposed autonomous mapping and navigation methodologies are evaluated in real-time experimentation. The experimental setup and the results are discussed in the next section.

4. Experimental Results

An In-house robotic platform with a LIDAR assembly has been developed for experimentation, as shown in Figure. 4. It consists of an RPLIDAR- A1M8, for collecting the data samples from the surroundings to perform mapping and exploration. Teensy is a microcontroller board which uses the Arduino firmware for acquiring IMU and raw encoder data. The acquired data is then transmitted to the RaspberryPi3 for filtering. The compatibility of Teensy is better suited for our experimentation due to better computation speed and connectivity compared to Arduino boards.
The robot is driven using the skid steer model using two encoder motors. An 11.1V, 10000mAh, 25C Li-Po battery is used as the power source for the robot. The Raspberry Pi is not capable to solely acquire IMU, and encoder data as tri-layered mapping and navigation stack processes are running in the foreground. In order to minimize RAM consumption and computational load on the microprocessor, a Teensy microcontroller board is implemented to send raw data to the Raspberry Pi. Thus, the Teensy controller board is interfaced with the Raspberry Pi3 using the Arduino-Raspberry bridge through the ros.h library.

![Final outlook of the exploration robot](image)

**Figure 4.** Final outlook of the exploration robot

The RQT framework is used along with other software frameworks to configure the PID controller for the robot. A precise set of tuned PID values are acquired by reducing the error margin close to 0. The acquired values frequently change in real-time as the direction of the wheel turn influences the feedback signal from a pair of rotary encoders mounted on the rear wheels, as shown in Figure 4. EKF is implemented to fuse the calculated odometry data with the IMU data to reveal the position and orientation of the robot.

The pose and twist results (Figure 5) are useful for estimating the odometry of the robot over time using covariance, linear and angular acceleration values. Such that is used for working on the navigation stack. These values are acquired from the accelerometer and encoders by applying the EKF filter.

![Result of Pose and Twist of the robot](image)

**Figure 5.** Result of Pose and Twist of the robot
The odometry data predicts the change in position of the robot over time. The motion of the robot is the trajectory of the robot as projected (Figure 7). It is built using EKF and odometry values from the sample values collected by the Raspberry Pi. From this application, advanced motion principles are developed into the robot. It imparts error detection by comparing real-time values with the desired values. The four significant aspects of being considered while acquiring the trajectory output are– base footprint, base link, imu link, odom value. By checking the odometry values, one can calculate the misdirection of the robot which can then be corrected with the application of a PID controller. In Figure 7, the mobility of the robot represented by the trajectory between the starting and ending positions is shown as red colored arrow marks. In the Figure, a yellow colored reference trajectory line shows the deviation in the real trajectory with respect to the reference trajectory. The occurred deviation is rectified with the help of PID and acquired values of accelerometer and encoder signals.

The RPLIDAR is initialized to generate 8000 laser pulses for each rotation. The laser pulses have been collected and oriented using IMU. The output of the LIDAR pulses from the Raspberry Pi is sent to the workstation for visualizing the generated sample points (Figure 7). Rviz is a visualization tool in ROS that displays the pulses generated real-time with respect to the distance from corresponding obstacles. The point cloud data obtained from this data is crucial to undertake navigation within the environment.

![Figure 6. Results for odometry of the robot](image6)

![Figure 7. The generated RPLIDAR laser signals](image7)
It requires EKF laser data, PID as well as the odometry of the robot to control the dual motors with ease. The figures 8 & 9 represent the final output of the robot, which includes the tri-layered map as well as its navigation stack, respectively. Navigation and mapping stacks are merged together to find shortest distance by avoiding the obstacle. The map has been pre-saved using geometrical path planning technique. The formulated path is represented in the Figure using a black line.

![Figure 8. Result of Tri-layered mapping](image)

![Figure 9. Navigation using geometrical path planning](image)

The following graph (figure 10) represents the odometry of the robot during navigation. Odometry represents the change in position over time; the trajectory of the robot is inferred to be similar to the curve plotted in the graph.

![Figure 10. Odometry graph](image)

5. Discussion
In this section, the results obtained from experimentation are discussed. It can be inferred that; the mapping efficiency is improved by using the proposed algorithm. Efficient exploration is carried out using the compilation of a tri-layered map of an area along with geometrical path planning technique. The software frameworks added into the virtual environment provides better visualization of a generated map in real-time by improving the GPU performance. The workstation is not necessarily bound to wired connection; such a user-friendly setup allows users to collect data samples from the LIDAR into their computers wirelessly via localhost.
It should be noted that the probability of map shuffling is well minimized such that the robot can perform precisely even in the dynamic mode of operation. Our proposed tri-layered mapping technique enables the LIDAR to scan three layers of an area and to stack them together. This feature yields robustness to the generated map by improving better obstacle dimension-recognition and better noise filtering.

The geometrical path planning technique comprises of PID and odometry values as well as simultaneous feedback from the IMU to provide ease of mobility and quicker path tracing. The algorithm makes use of a virtual circle plotted for each obstacle to determine minimal angular dimensions and its respective coordinates, including the robot’s clearance such that it can perform the shortest possible turns without collision.

6. Conclusion and Future work
In this paper, an autonomous SLAM-based system has been proposed. The proposed tri-layered mapping method is employed in SLAM technology and fused with a hardware setup equipped with high rpm encoder motors. The presented results show that the robot is capable of performing a tri-layered map generation. The comparison table in Section 4.2 demonstrates the improvement in mapping efficiency compared to the conventional techniques. The geometrical path planning technique provides better navigation and mobility to the robot. The proposed methodology is distinguished with the existing conventional methods to depict clarity of the generated map with the improvements seen in the navigation stack. The system additionally comprises a PID controller to impart accurate odometry to the robot. It is essential to note that, the robot can infer most environmental aspects such as obstacles, boundaries, elevation and descent with the help of a cost-effective RPLIDAR. Further interfaced wirelessly from a computer, we can visualize the map getting generated in real-time as the mobile robot successfully evades the obstacles in an environment.

Our future work falls into the following perspectives:
1. Machine learning will be adopted in the proposed approach using TensorFlow software library to impart human detection to the robot, whose input will be acquired using a camera module.
2. The current two sets of wheels will be replaced with all-terrain wheels similar to an aluminium tank track chassis, for improved mobility and to compensate rough and wet terrains. The robot will be made lighter by minimizing the payload on the motors to reduce power consumption.
3. The scope of operation will be further extended using dual motor-operated wheels will be transitioned to four-wheeled encoder motor-operated robot to improve mobility and path tracing.
4. Upon broader applications, the adopted technology will be experimented in autonomous self-driving vehicles once the expected accuracy and robustness is achieved.

References:
[1] Ortiz, S., Yu, W. and Li, X. (2019) ‘Autonomous navigation in unknown environments using robust SLAM’, IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 1, pp. 5590–5595. doi: 10.1109/iecon.2019.8926620.
[2] Yan, Y. P. and Wong, S. F. (2019) ‘A navigation algorithm of the mobile robot in the indoor and dynamic environment based on the PF-SLAM algorithm’, Cluster Computing. Springer US, 22, pp. 14207–14218. doi: 10.1007/s10586-018-2271-3.
[3] Ma, L, Liu Y, Gao Y and Zhu C. (2018) ‘PlantBot: A new ROS-based robot platform for fast building and developing’, Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018, pp. 930–935. doi: 10.1109/ICMA.2018.8484404.
[4] Hampton, B, Al-Hourani A, Ristic B and Moran B. (2017) ‘RFS-SLAM robot: An experimental platform for RFS based occupancy-grid SLAM’, 20th International Conference on Information Fusion, Fusion 2017 - Proceedings. doi: 10.23919/ICIF.2017.8009744.
[5] Shen, D, Huang Y, Wang Y and Zhao C. (2018) ‘Research and Implementation of SLAM Based on LIDAR for Four-Wheeled Mobile Robot’, 2018 IEEE International Conference of Intelligent Robotic and Control Engineering, IRCE 2018. IEEE, pp. 38–41. doi: 10.1109/IRCE.2018.8492968.

[6] Harkishan G, Neha J, Aaron M, Chinnay S and Kiran G. (2018) ‘PCL-Based Autonomous Wheelchair Navigation in Unmapped Indoor Environments’, 2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018. IEEE, pp. 291–296. doi: 10.1109/UEMCON.2018.8796660.

[7] Fuchun, L., Yifeng, C. and Yunze, L. (2017) ‘Research on indoor robot SLAM of RBPF improved with geometrical characteristic localization’, Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017, pp. 3325–3330. doi: 10.1109/CCDC.2017.7979080.

[8] Xi, W, Ou Y, Feng W and Yu G. (2018) ‘An improved serial method for mobile robot SLAM’, 2017 IEEE International Conference on Robotics and Biomimetics, ROBIO 2017, 2018-January, pp. 1–6. doi: 10.1109/ROBIO.2017.8324466.

[9] Fan, X, Xin Z, Li J and Li Y. (2019) ‘A fire protection robot system based on SLAM localization and fire source identification’, ICEIEC 2019 - Proceedings of 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication, pp. 555–560. doi: 10.1109/ICEIEC.2019.8784563.

[10] Tuna, G, Gulez K, Gungor C and Mumcu V. (2012) ‘Evaluations of different Simultaneous Localization and Mapping (SLAM) algorithms’, IECON Proceedings (Industrial Electronics Conference), pp. 2693–2698. doi: 10.1109/IECON.2012.6389151.

[11] Ilbragimov, I. Z. and Afanasyev, I. M. (2018) ‘Comparison of ROS-based visual SLAM methods in homogeneous indoor environment’, 2017 14th Workshop on Positioning, Navigation and Communications, WPNC 2017, 2018-January, pp. 1–6. doi: 10.1109/WPNC.2017.8250081.

[12] Sundram, J, Nguyen V, Soh G, Bouffanais R and Wood K. et al. (2019) ‘Development of a Miniature Robot for Multi-robot Occupancy Grid Mapping’, ICARM 2018 - 2018 3rd International Conference on Advanced Robotics and Mechatronics. IEEE, pp. 414–419. doi: 10.1109/ICARM.2018.8610745.

[13] Abdelrasoul, Y., Saman, A. B. S. H. and Sebastian, P. (2017) ‘A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM’, 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation, ROMA 2016. doi: 10.1109/ROMA.2016.7847825.

[14] Shen, D., Xu, Y. and Huang, Y. (2019) ‘Research on 2D-SLAM of indoor mobile robot based on laser radar’, ACM International Conference Proceeding Series. Doi: 10.1145/3351917.3351966.

[15] Shen, D, Huang Y, Wang Y and Zhao C. (2018) ‘Research and Implementation of SLAM Based on LIDAR for Four-Wheeled Mobile Robot’, 2018 IEEE International Conference of Intelligent Robotic and Control Engineering, IRCE 2018. IEEE, pp. 38–41. doi: 10.1109/IRCE.2018.8492968.

[16] Papachristos, C, Kamel M, Popovi´ M, Khattak S, Bircher A, Oleynikova H, Dang T, Mascarich F , Alexis K and Siegwart R. (2019) ‘Autonomous Exploration and Inspection Path Planning for Aerial Robots Using the Robot Operating System’, Studies in Computational Intelligence. Doi: 10.1007/978-3-319-91590-6_3.

[17] Wei W and Shirinzadeh B. (2019) ‘Orientation Correction for Hector SLAM at Starting Stage’, 2019 7th International Conference on Robot Intelligence Technology and Applications (RITA) November 1 - 3, 2019, Daejeon, Korea.

[18] Xiao, Y., Ou, Y., & Feng, W. (2017). ‘Localization of indoor robot based on particle filter with EKF proposal distribution’. 2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM). doi:10.1109/iccis.2017.8274839.