ROS-Based Global Path Planning for Autonomous Ground Robot Using the Pre-Built Map of the Environment

Hamza Mukhtar
Alkhawarizmi Institue of Computer Science
University of Engineering and Technology
Lahore, Pakistan
hamza.mukhtar@kics.edu.pk

Muhammad Ahmad Hasan
Alkhawarizmi Institue of Computer Science
University of Engineering and Technology
Lahore, Pakistan
ahmed.hasan@kics.edu.pk

Muhammad Usman Ghani Khan
Alkhawarizmi Institue of Computer Science
University of Engineering and Technology
Lahore, Pakistan
usman.ghani@kics.edu.pk

2021 International Conference on Robotics and Automation in Industry (ICRAI) | 978-1-6654-2343-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRAI54018.2021.9651374

***Abstract*—Optimal global pathway planning has been a great area of research for many years and continues to develop new approaches. Without a suitable global path, a robot can get stuck in local minima. For path planning, a map is required which has been developed implying different sensors by moving the robot manually in an environment that is troublesome, costly, time taken to process, and lacks accuracy. This paper suggests a way to utilize pre-built maps of the environment. All the environments are built according to the structural map, we aim to use the pre-built map for path planning which is faraway accurate and detailed even to the extent of millimeters. We are introducing an approach that converts the pre-built map into a robot understandable format through image processing. The superiority and effectiveness of our approach have been verified using Robot Operating System (ROS) packages. ROS-supported path planning algorithms are applied on pre-built maps and results are compared with the result of the same approaches on sensor maps which produce better results in terms of pathway length and time taken.**

***Keywords*—robot navigation, turtlebot, mobile robot, SLAM, robot operating system, map building**

I. Introduction

Mobile robotics is the most fascinating phenomenon that has been emerging in the area of artificial intelligence. With the 4th industrial wave, a rapid increase of mobile robots is disrupting the dynamics of the industrial sector. There are numerous scenarios where tasks can become arduous for humans and also face efficiency issues. Mobile robots are hyper-intelligent autonomous machines capable of cognition of environment, gather and process the sensors data, localize itself, and devise a path to move on [15]. The immense employment of intelligent Mobile robots has disrupted the manufacturing industry, agriculture, military, and rescue in natural calamities due to their potential to work in hazardous environments and to work round the clock. In recent years, robots are being used to deliver food in restaurants as well as home delivery and searching, excavations, and disposal of mines, bombs and to transfer goods in warehouses and factories. The ability of mobile robots to navigate autonomously plays an anchoring role in autonomous cars, logistical and service robots, robots for personal assistance, and as a workforce in the industry. Due to the security situation, these days robots are also becoming a vital instrument for surveillance [13]. According to the World Economic Forum (WEF), the mobile robotics market was worth USD 135 billion in 2019 and showing growth at the rate of 17% a year which will be worth USD 180 by 2021 [6]. For the job market, automation will create 58 million jobs in USA in next 5 year as predicted by WEF. [7]. Fast and ongoing disruption to job markets due the Fourth Industrial Revolution has been further accelerated because of Corona pandemic-related recession of 2020.

For mobile robots, autonomous navigation is considered one of the key requirements. Navigation is a combination of procedures able to move the robot from the initial location to the destination point in any of the 3 dimensions. For the robot to move in a 2-dimensional (2D) environment, a map of the environment is needed along with a planned path that should be an optimal path in terms of length to reach a destination efficiently. Path planning for robot navigation is very crucial to move robot collision-free from obstacles which is an intrinsic part of autonomous mobile-based robots. Map for robot navigation, a prerequisite for path planning, contains critical information about the environment in which the robot needs to move on. Trajectory planning can be described as providing a map of its environment, dynamically produce a feasible pathway while handling obstacles that connect a starting point and destination through the shortest path by considering optimal parameters like minimum distance, off from obstacles, minimum turnaround time, and minimum power consumption.

Formation of maps for unfamiliar environments with localization of robots is the first major step for mobile robot navigation. The Representation of the environment along with the position of the robot is a problem on center stage for mobile robots. Mapping of an unknown environment is usually done through the common method SLAM (Simultaneously Localization and Mapping). Extraction of information from the environment regarding mapping process accompanying localization of robot's position through SLAM based approach involves sensors like depth camera, Kinect, ultrasonic and laser scan [14]. For path planning, the presence of an environment map is mandatory. One approach to obtain a map of the environment is through SLAM-based techniques as described in Robot Operating System (ROS) [14]. Through SLAM, a map is built by moving the robot manually in the environment. Sensors are mounted on the robot for the extraction of environmental information. Eventually, these extracted parameters are processed through different algorithms to form a map. Observing this approach, the whole process cycle for the creation of a map is needed to perform for every new environment which requires time, human effort, and energy.

Path planner module can be categorized into global and local planning [22], where the former device ends to end the shortest trajectory while managing static obstacles and the latter component dynamically recalculates the pathway in dynamic conditions to avoid obstacles locally. Reference trajectory is the ultimate output of the Global planner module.

This global reference is devised before the beginning of navigation by assuming a static environment. The local path is rooted in global reference and the occupancy grid is built dynamically by utilizing environment data gathered through the robot's sensors. Global reference is devised once as the shortest path before the start of navigation, later on, it is revised concerning the variations occurring in the environment to guarantee a swift and collision-free navigation while escaping static and dynamic obstacles. Navigation of a mobile robot based on global reference is a key ingredient in this regard as conjunction towards the destination in varying environments is tough to guarantee. In absence of global reference, local planning of freeways may be stuck in local minima where it can be troublesome to come out of it. The presence of accurate numeric information regarding the environment is not necessary for humans to form decisions because they can do adaptive control to a larger extent. The prodigious ability of humans to perform navigational and cognitive decisions without explicit parameters of the environment is remarkable but for robots, these sorts of tasks are challenging and have to consider parameters of a workable environment.

In our approach, a pre-built map of the environment is in focus. Most environments and structures are built upon maps so it would be a handy approach if mobile robots can utilize pre-built maps of the environment which will save time, human effort, and cost. These pre-built maps, developed in different tools, are modified in such a form on which robots can perform path planning for navigation. It will be a generic mechanism for robots to start navigation on pre-built maps instead of constituting maps by scanning the environment. This pre-built map is used for planning to devise a global shortest freeway without losing the accuracy of the path.

The main contributions of our work are mentioned below.

- By utilizing the pre-built maps of the environment, a methodology is proposed for global path planning which ends the need for the cumbersome process of SLAM for map formulation.
- The proposed methodology can minimize the chances of local minima traps during global planning.
- The effectiveness of the work is verified by comparing the results of ROS-based planning algorithms on SLAM-based maps and pre-built maps. The superiority of our work for global path planning in consideration of the shortest path and minimum planning time is witnessed.

Conventional approaches [4, 3] do planning on maps formulated through different sensors which is a cumbersome process. The proposed approach uses the pre-built map of the environment and uses ROS packages to implement planning. Route planning on pre-built has higher accuracy as compared to the sensor-made map. The remaining paper is arranged as follows. In part. 2, we have described prior proposed work in literature. Part. 3, explains the multiple steps of the proposed methodology. The demonstration of the proposed work on the simulator is described in part 4. Part 5 of this article compares the performance of our methodology with prior approaches. Part 6 summarizes the whole discussion along with future work.

II. Literature review

A great amount of research has been done for mobile robots to formulate an optimal route to perform collision-free navigation between two points with minimum time and energy [1]. Different conventional techniques for freeway planning such as the roadmap technique of visibility graph [29] face trouble due to higher computational cost in formulating the global reference in bit complex environments. Structured technique, Voronoi network, acquires unsmooth reference network through incoming data from sensors [30]. Through this, the Simplified environment model is made as complex obstacles are not needed to decompose. Path planning complexity is reduced to a greater extent due to fewer nodes of free space of environment with network composition. Below we have categories the different approaches into three classes.

A. Iterative Search

Pathway planning using a smooth approach had been proposed to build an optimal trajectory made up of the Bezier curve with the constraint of curvature [11]. Different flavors of stochastic optimization algorithms have been proposed to devise an optimal global reference for mobile robots. These intelligent methodologies are nature-inspired and witnessed the capability of parallel computing and robustness. One of the stochastic search-rooted algorithms is GA [17]. GA is a somewhat representative approach to devise an optimal freeway whose inspiration comes from natural evolution. There is another widely used algorithm to tackle global planning called the Ant colony algorithm. It undergoes the functioning of ants that fix pheromones during the search for follower ants to observe a feasible path. The shortest path is chosen by ants based on the number of pheromones among different paths [18]. Besides GA and Ant colony, there are other algorithms inspired by a natural process, one such type is the artificial bee colony but it lacks to produce an accurate optimal path and also not smooth enough due to the composite space road map approach [18]. Different research approaches suggest that post-processing can produce a smoothed map if not optimal before post-processing but it will not be an optimal pathway always [22].

B. Evolutionary Strategy

Path planning can be believed to be an optimization problem. The research community has devised numerous heuristic approaches to tackle this phenomenon [25], and undeniably, the particle swarm optimization (PSO)approach stands out among its rivals and has been extensively utilized in trajectory planning for navigation of mobile robots [8]. With slight variation, PSO using accelerated particles has formulated where best particle PSO is employed for global planning [9]. An amalgam of parallel metaheuristic PSO-based approaches has been evolved for global pathway planning where three parallel PSO-based algorithms have integrated along with communication engine for the production of an optimal path which then made smooth with a smoother called B-spline but only shortness of path tacked in this approach. With swift convergence and non-complexity, PSO algorithms are popular to resolve path planning issues [9].

Based on downscaled path magnitude, PSO optimizes the length of the path [25]. For smooth global planning, a variance of PSO has been developed with modified PSO along with $\eta3$-splines hinged kinematic constraints [24]. However, there is a higher possibility of local optima trap because PSO algorithms seem incapable to maintain equilibrium in exploration and exploitation well enough [25]. Quantum-behaved PSO (QPSO)approach is a quantum-based particle swarm optimization algorithm. In quantum space, δ-potential is a nice shape of the particle swarm in the context of quantum motion; particle nature in an accumulation state is much different. It can be utilized to devise an optimal solution; there, the search performance of QPSO in comparison with standard PSO seems much better.

A lot of work has been done to enhance the performance of QPSO. For accuracy and better global convergence, different approaches are being proposed to improve QPSO. Gaussian potential in GQPSO introduces a mechanism for searching the particle position [20]. The weighted version of QPSO includes weights for the calculation of the mean optimal position in QPSO to reflect the significance of particles while evolving. According to fitness values, weighted QPSO sorts the particle in downward order, then the weight coefficient is assigned to every particle which decreases linearly with the rank of particles [5]. In conventional QPSO, a parallel operation approach is introduced which unites the particles information possessing quantum nature and information of each dimension of each particle [28]. Accuracy and convergence of QPSO are improved through control of compression expansion parameters, random distribution, and hybrid search approach. Hybrid IQPSO approach based on a coefficient method which injects a novel variable that exhibits command on the best, worst, and outperformed best particle value [10]. To handle continuous wide-scale non-linear queries, a memory mechanism-inspired approach was proposed which enhanced the performance of QPSO. This mechanism involves particles to possess experience while performing local research before getting into the evolutionary process plus memory capacity which enhances the global search capability [27].

C. Grid-Based Search

An open-source mechanism Robot Operating System (ROS) [19] is taking pace for robotics application development which uses SLAM inspired techniques, such as Gmapping [4], Hector SLAM, Lago SLAM, KArto SLAM, Core SLAM [15, 3], to build a map for pathway planning [25]. To use SLAM-based techniques, the robot needs to move manually in the environment to scan the environment with the help of sensors, and then obtained sensor data is converted into a grid map. ROS is a complete package containing tools, simulators, libraries, and drivers for the development and managing of robot functions [19]. Robots involve occupancy maps to formulate the environment. Dijkstra algorithm begins traversing from an initial point and goes on until the destination is reached, and reduces this computational cost. A list of connecting nodes is identified which formulates the single shortest path that couples the ending point with the start. It explores nodes that are closest as well as far from goal position which makes it computationally expensive having time complexity O(n^2) where n represents the number of vertices, the variance of Dijkstra was introduced which focuses only on traversing nodes that are at the shortest distance from the goal [39]. The modified variance of the D algorithm is developed to look after the problem of the long path [26] To build a path with optimal cost, an algorithm called A* [12] had developed, which incorporated an arc-line method for the smoothness of made path , and this is regarded as the most optimal solution for heuristic search [16]. A modified version of A* was proposed for the global pathway planning problem which superseded the conventional A* algorithm in terms of time efficiency. The main idea is based on exploring the structure of grid maps to formulate an optimal pathway, without exploring any location more than once [21].

Until now, we have found a handful of pieces of research in the literature concerning global pathway planning and map

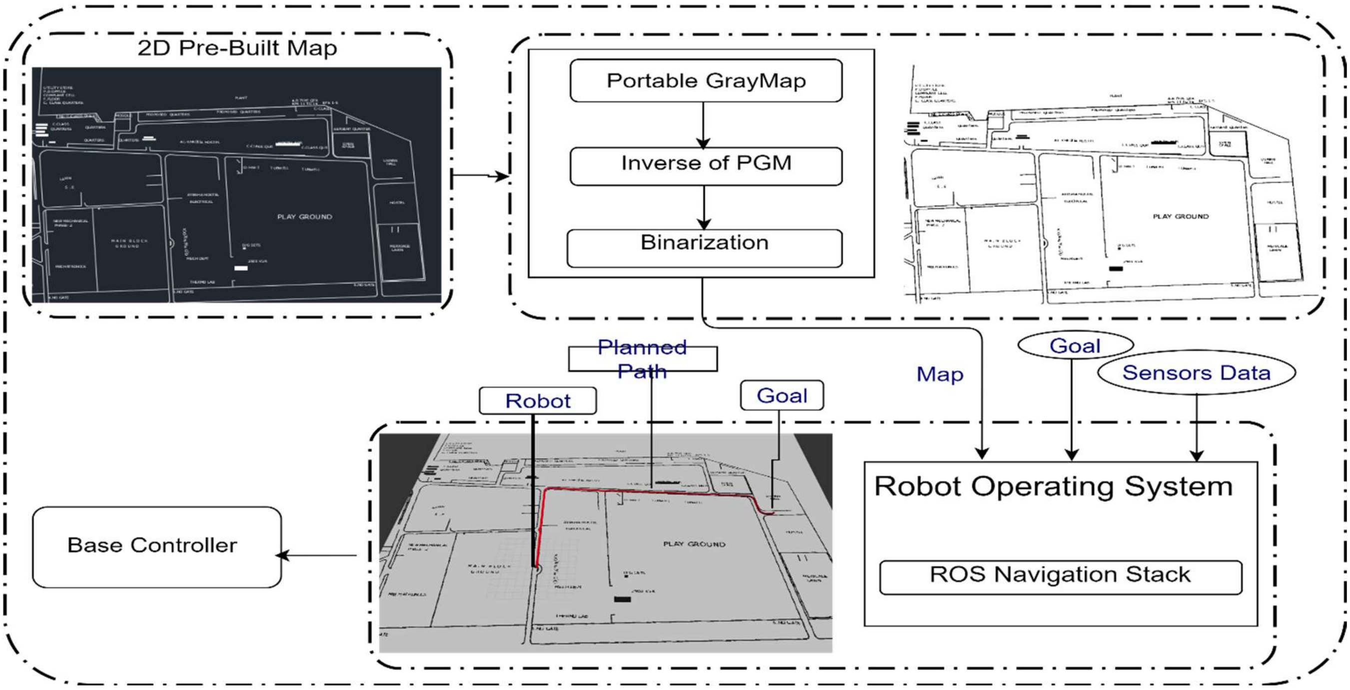

Fig. 1. Diagram shows the 4-step global path planning methodology: (1) conversion of pre-built into Portable Gray Map, (2) inverse of Portable Gray Map, (3) binarization of inverted Portable Gray Map, and path planning using ROS

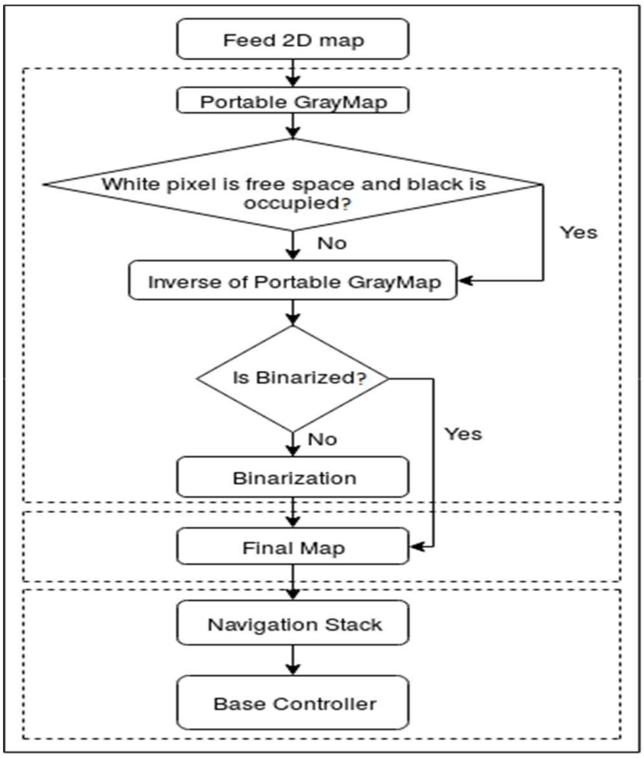

Fig. 2. WorkFlow diagram of the proposed methodology

formulation in the context of mobile robots accompanied by taking care of several constraints like the shortest freeway, optimal curvature with its derivative, restrained curve, collision-free, minimum energy consumption and time. Due to the possibility of local trap occurrence, all the mentioned constraints turn it tough to perform tasks in an optimized manner. Moreover, all the mentioned approaches build maps through sensors, such as laser scan, time of flight camera, by moving the robot in the environment. It is becoming the demand of the day to have an approach that utilizes a pre-built map of the environment for global reference planning and draws accurate maps with optimal cost.

III. METHODOLOGY

This part aims to formulate an optimal pathway for mobile robots to navigate on a pre-built map of the environment in terms of computational cost and length. For navigation, a map of the environment is a prerequisite that is generated presently through SLAM which utilizes mounted sensors on robots. Instead of building maps through sensors of mobile robots, we are proposing a mechanism that eliminates the need for map building through the utilization of pre-built maps of the environment. Maps of the environment are built in different tools, such as AutoCAD and Solidworks, which are in different formats (e.g. DWG) which could not be used for pathway planning straightway in frameworks like ROS which utilizes Portable Gray Map (PGM) maps for freeway planning.

Proposed methodology (Fig. 1) for global path planning consists of four steps: (1) PGM conversion, (2) Inverse of GMP, and (3) binarization, and path planning. In the first step, the 2D map of the environment is converted into a PGM. To consider black pixels as occupied and white as free, the inverse of the PGM map is taken in the second step which converts the black pixel to white and white to black. To make clear decision making between free and occupied space, the map is then binarized in the third step which is provided to the planning module to perform planning. Workflow for path planning based on proposed methodology is shown in Fig. 2.

A. Conversion to Portable Gray Map

In this approach, a 2-dimensional map of DWG, PNG, JPG, or any other format needs to convert into a PGM format which is a grayscale form of the map having pixel values between 0 to 255. In RGB form, it's difficult to tackle three color channels and also lacks efficiency in terms of memory. Processing three color channels are computationally expensive in comparison with a single channel in grayscale. Using a weighted method, colored maps are converted in grayscale form. Grayscale conversion is performed with a series of different weights to find optimal parameters which can make global pathway planning smoother. The following weights have been found optimal and they give better grayscale conversion

$$c = 0.3*R + 0.59*G + 0.11*B \quad (1)$$

B. Inverse of Portable Gray Map

After grayscale conversion, the obtained map contains white and black areas. To consider black pixels as restricted space and white as free, the inverse of a PGM map is needed to perform. With this inverse, areas of the road become white and boundaries become black. The equation below has been used to perform binarization.

$$M_{binary} = 255 - M_{original} \quad (2)$$

C. Binarization

Global pathway planning on a PGM map becomes ambiguous where pixel values come in between 0 to 255 because 0 is regarded as restricted space and 255 as movable but values in the middle range remain indecisive. For distinct decision-making on mix shades, the map needs to binarize. As a binary image has two values 1, 0, the image is divided into two histograms, by using a threshold that turns each histogram as compact as possible. This minimizes the weighted intraclass variance $\sigma^2{}_w(t)$, shown by (3). Variable refers to the threshold described earlier whose values range from 0 to 255.

$$\sigma^2{}_w(h) = C_1(h)\sigma^2{}_1(h) + C_2(h)\sigma^2{}_2(h) \quad (3)$$

To compute $\sigma^2{}_w(h)$, probabilistic function $P$is obtained against each pixel value. Using function **P**, histogram distribution for both classes is calculated, which is later normalized to follow the probability distribution. Using threshold *t,* pixel values are divided into C_1 and C_2 through function defined in equations (4) and (5) respectively.

$$C_1(h) = \Sigma^h{}_{p=1}P(p) \quad (4)$$
$$C_2(h) = \Sigma^h{}_{p=h+1}P(p) \quad (5)$$

C_1 and C_2refers to pixels having intensity range in [1, h], [h + 1, 1] respectively. Here, p refers to the highest pixel intensity (typically 255). Then, mean $\mu(h)$ for C_1 and C_2 are calculated.

$$\mu_{c1}(h) = \Sigma^h{}_{p=1}\, pP(p)/C_1(h) \quad (6)$$
$$\mu_{c2}(h) = \Sigma^h{}_{p=1}\, pP(p)/C_2(h) \quad (7)$$

Then, variance σ against C_1 and $C_2$are calculated.

$$\sigma^2{}_1(h) = \Sigma^h{}_{p=1}[p - \mu_{c1}(h)]^2\ P(p)/C_1(h) \quad (8)$$
$$\sigma^2{}_2(h) = \Sigma^h{}_{p=h+1}[p - \mu_{c2}(h)]^2\ P(p)/C_2(h) \quad (9)$$

The total variance is calculated by equation (9).

$$\sigma^2 = \sigma^2{}_w(h) + \sigma^2{}_v(h) \quad (10)$$

where,

$$\sigma^2{}_v(h) = C_1(h)C_2(h)[\mu_{c1}(h) - \mu_{c2}(h)]^2 \quad (11)$$

The binarization step turns the map into either black or white pixels. After the binarization step, the map becomes distinct in deciding the pixel whether it's free or occupied.

D. Global Path Planning

After Binarization, path planning is being performed using the ROS framework which processes maps in PGM format. PGM conversion of a PNG map is done through a Python script along with a YAML file which contains different parameters. These parameters are required to interpret and process the PGM map while path planning (Fig. 3). To put balance between accuracy and speed, the resolution is fixed to 0.05 meter by pixels ratio. In case of the low value of resolution, accuracy will be compromised and the high resolution will put a processing burden on the machine. The threshold value for occupancy probability is set to 0.7 and pixels with a value higher than this probability are regarded as occupied pixels. The probability threshold for free occupancy is fixed at 0.196 and pixels having values lower than the threshold are viewed as free space. Python script produces two files, PGM and YAML, from PNG map. PGM is an image file that encodes occupancy data which is used by ROS along with YAML to plan an optimal pathway.

```
M:2Dmap
procedure PreBuiltMap(M)
M→PortableGrayMap(M)
if boundary ==whitepixels and freespace==blackpixels
then
    M→Take inverse of PGM(M)
                                    else
    M→M
if M.max pixel >1
then
    M→Binarization(M)
                        else
    M →M
output →feed map to navigation stack base controller
end procedure
```

Fig. 3. Pseudo-code for global path planning on a pre-built map

The first condition refers to zero cost at the final destination node and later specifies that each node is required equal or less than the shortest path length of any nth node to the destination node. Distance measure can be Euclidean or Manhattan distance from an initial position, *i*, to the final position, f, in the map. Equation 4 and 5 are the mathematical representation of distance measures.

1. Euclidean distance measure
   $$E(x_i, y_f) = ((x_i - x_f)^2 + (y_i -_f)^2)^{1/2} \quad (12)$$
2. Manhattan distance measure
   $$E(x_i, y_f) = |(x - x_f)| + |(y_i, - y_f) \quad (13)$$

For pathway planning, goal destination is given by involving a simulation environment from the ROS package. A map is loaded in the simulator Rviz which receives the goal point and applies the desired algorithm to find an optimal path connecting the initial point with the final location through the shortest path.

IV. Simulation and Experimentation

This section of the article describes methodological work in practice to verify the outcome in a simulator. Using the ROS package, different algorithms are applied on a pre-built map and laser scanned maps. Rviz visualizer is used to witnessing the planned path for navigation to reach the destination. ROS is an integrated environment containing different packages to manage the operations of the robot. The Intel Core i7 CPU 760 running on 3.80GHz, having RAM of 8GB, is used to conduct planning processes in simulation. ROS Kinetic is installed on Ubuntu 16.04 LTS.

For this experiment, 2 versions of the map, (1) pre-built map and (2) map built using ROS G mapping algorithm by implying Laser scan (Hokugo), are tested. Each version has 15 maps of different parts of campus against dimensions 2000x2000, 3000x3000, and 4000x4000 pixels in equal proportion. So, we have 30 maps for pathway planning this experiment. The 2-dimensional AutoCAD maps of university campuses are used where the environment is divided into zones and after doing all the steps, zones are surrounded by black closed boundary and white space between zones represents roads. In Fig 4, a 2-dimensional map of some parts of the campus is presented, where we can observe the division in zones.

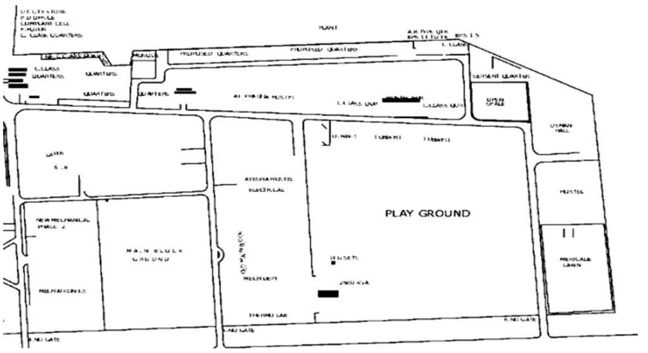

Fig. 4. Overall map of the environment

Laser scanner maps are formulated by navigating the real robot, developed by our team, manually in the environment. A laser scanner is mounted on the robot which is used by the map server to build a map of the environment. Gmapping, a map-building algorithm, is designed to use laser scanner sensors to build 2D maps. Formulated maps are saved in files using the ROS map server package which saves maps in two files, (1) YAML file which refers to metadata, and (2) Image file which possesses the occupancy data. In Fig. 5**,** a 2-dimensional laser-scanned map of some parts of the campus is presented, where we can observe that there is gray space which indicates the area beyond the focus of the laser scan.

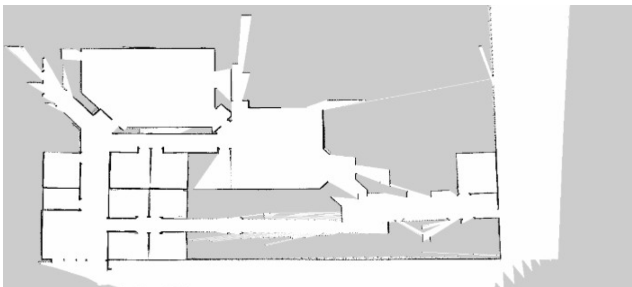

Fig. 5. SLAM based map using a laser scanner

The map is loaded in ROS and visualized in Rviz where the goal position is also defined through a mouse pointer. Both initial and goal, points are taken by ROS global planner to plan the shortest path between the initial position of the robot to the desired position. Turtlebot model is used as a robot that utilizes a planned path for navigation. Melonee and Tully developed it in 2010 at Willow Garage which is open source and also equipped with a laser scan sensor which makes it a good fit for navigation [2].

Experiment demonstrates the optimal pathway planning on a pre-built campus map in the Rviz simulator environment. Robot, Turtlebot, current position is considered the initial reference for planning by ROS global planner which starts searching optimal positions from initial reference. The destination position is chosen through the pointer's click on a random location. The Planner module has devised the shortest pathway to reach the destination which can be seen in the form of a red line connecting the initial position with the destination. The goal is defined such that the planner module has to formulate the pathway in the presence of a complex environment instead of a straight path. The suggested path avoids all the restricted areas successfully and seems optimal in terms of length which exhibits the effectiveness of the method on a pre-built map.

The objective of this part is to show the performance of the planning process implying on pre-built maps. To make the result evident, pictures from the Rviz simulator are below. Path planning through RA* algorithm, on pre-built maps, was observed optimal. To exhibit the optimal performance of path planning on pre-built maps, we have chosen a relatively complex path where there were multiple options to select. To demonstrate this, we have used the same map, against the same dimension, to choose two different goal positions.

Maps in Fig. 6 and Fig. 7 are of 4000x4000 size, where one robot is present on the left side (Fig. 5) of the wall (black line) while in the second map, our robot is on the right side (Fig6). Both took different routes to plan a path against respective goal positions.

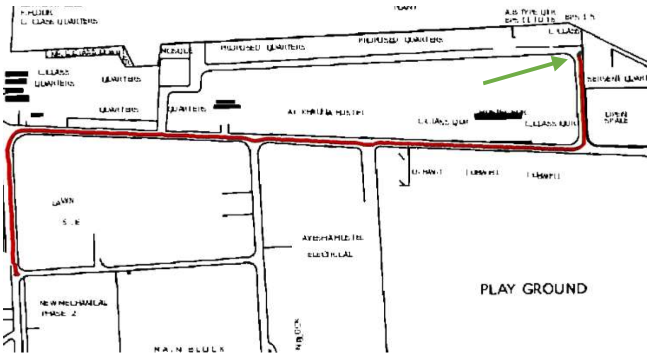

Fig. 6. Path planning when robot is on the left side. Arrow refers to robot.

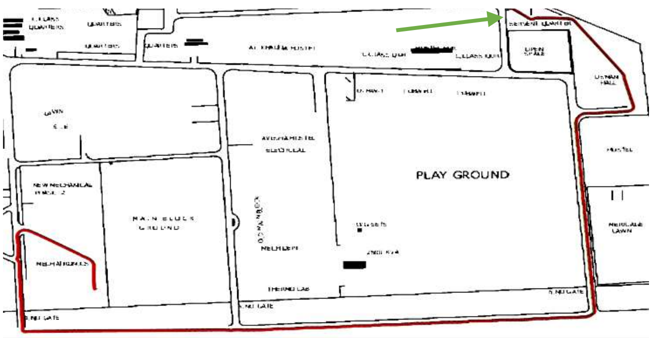

Fig. 7. Path planning where robot is on the right side. Arrow refers to robot

Fig. 8 and Fig. 9 contains maps of 3000x3000, planned path in both maps pass through the cuts in between black lines which shows the optimal path planning regardless of the presence of other straight paths. The path on both maps goes from different cuts.

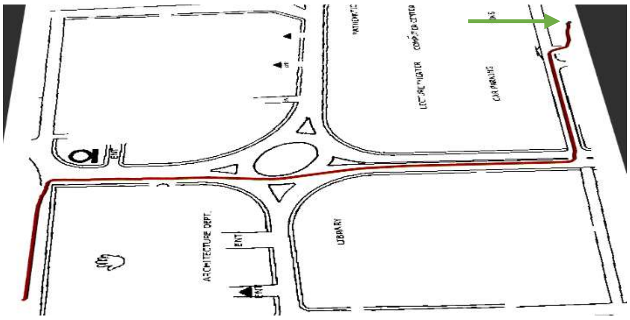
Fig. 8. Here green arrow refers to the current location of the robot


Fig. 9. Here green arrow refers to the current location of the robot

Left and right map in Fig. 10 is of 2000x2000 size, here we have tried to assign such a goal where path planning can be done from left as well as from the right side of the robot. It can be seen that the robot has planned an optimal path even if the path goes from the cuts.

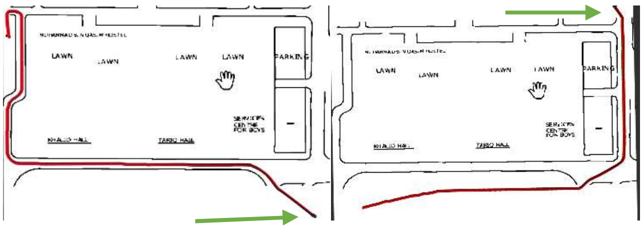
Fig. 10. Here green arrow refers to the current location of the robot

V. Results

To evaluate the results on pre-built maps against sensor-made maps, statistical results of performance on simulation are compared. Three performance indicators are assessed to examine the performance of global pathway planning on implying above mentioned approaches: (1) the Average Displacement Error (ADE) which represents the error between ground truth and planned path at every point, here error refers to Mean Square Error (MSE), (2) total path length which refers the length of planned path, (3) planning time which is spent to plan global pathway.

In this analysis, ADE is referred to as Euclidean distance between path points of path planned by our approach and ROS approaches. The Path planned through our approach is considered as ground truth to calculate Euclidean distance. The planning time reflects the execution time of the approach. Shorter the execution time, there are more chances to execute

Algorithms	Average Path length (m)			ADE (m)			Average Time (ms)		
	Map 1	*Map 2*	*Map 3*	*Map 1*	*Map 2*	*3rd map*	*Map 1*	*Map 2*	*Map 3*
A* (Pre-built)	1711.5	2749.2	4153.1	-100.4	-169.3	-310.4	1141.5	1833.73	2770.13
RA* (Pre-built)	1750.4	2849.6	4251.5	-70.4	-97.1	-160.15	1123.76	1829.51	2745.54
Dijkstra (Pre-built)	2113.3	3165.2	4689.2	-110.6	-155.5	-185.5	1504.46	2253.6	3363.72
A* (Gmapping)	1811.9	2918.5	4463.5	N/A	N/A	N/A	1273.71	2063.37	3155.60
RA* (Gmapping)	1821.8	2946.7	4411.6	N/A	N/A	N/A	1238.67	2003.7	2999.92
Dijkstra (Gmapping)	2223.9	3320.7	4874.7	N/A	N/A	N/A	1872.5	2796.02	4104.51

Table 1. Result comparison against all three planning algorithms.

the planning process in real-time. The length of the path determines the optimality of the applied approach which reduces the traveling cost for a robot to reach the destination.

Tab. 1 manifests the result of performance metrics for all approaches on both types of maps. Here path length is obtained from averaging the path length planned on equal size maps provided the initial and goal was the same. ADE is measured by taking the length of the planned path on a pre-built map as ground truth. Here the negative sign with ADE represents that the length of the planned path against a pre-built map is less than on a laser map by the value of ADE value in comparison to the same size map. The pixel in the map represents the one meter in the physical space. Planning time is the average time against all four same-size maps. 1st , 2nd and 3rd map are of 2000×2000, 3000×3000 and 4000×4000 respectively.

From table 1 it can be concluded that pathway length on pre-built maps is small in comparison with length on laser scan maps. There is a proportionally wider difference as planning goes from 1st map to 3rd map which shows that the margin of error in pre-built is lower than laser maps. From a lesser length of planned path, we can extract that planning time on pre-built is also on a declining trend than laser maps but another testimony can be proved from larger proportional decrease.

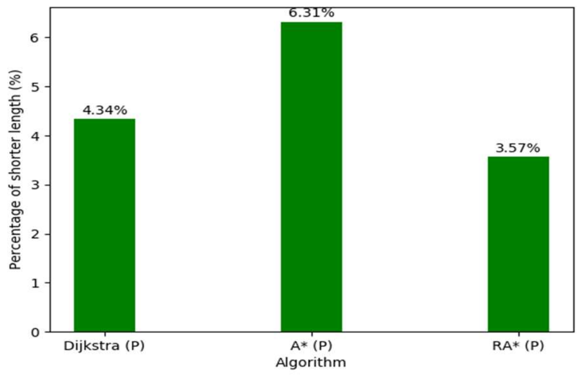

Fig.11. Graph shows the percentage decrease in length on pre-build maps

Fig. 11 is showing the percentage decrease of length when planned path length on pre-built maps is compared with an on-sensor map against the same approach. Although, improvement can be seen in all the approaches A* is exhibiting the highest performance.

Fig. 12 and Fig. 13 show the box plot representation path length and planning time. From the comparison of the median, we can extract which approach is optimal in all performance metrics. From min and max values, performance on comparatively shorter and larger the map can be seen. Here 'G' refers to a Gmapping-based map and 'P' refers to a pre-built map.

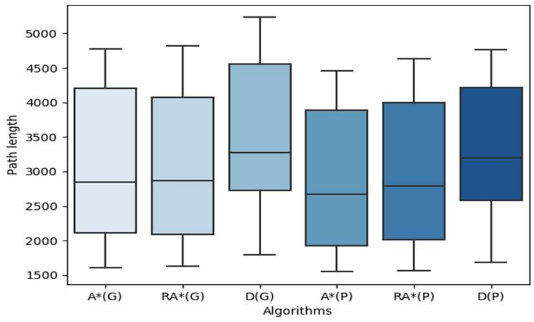

Fig. 12. Box plot for path length

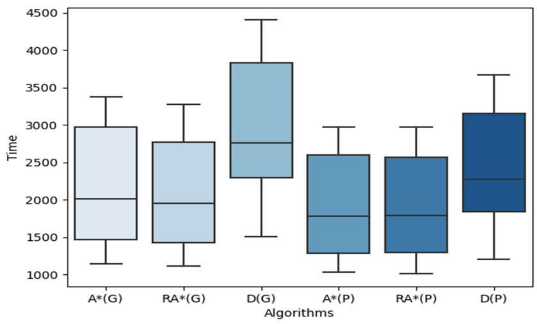

Fig. 13. Box plot for path planning time

From Fig. 12, We can observe that RA* is the best performer on G mapping maps but A* is taking lead on pre-built maps. All approaches on pre-built are suggesting smaller paths in contrast with Gmapping-based maps. So, we can conclude that planning on pre-built is more optimal. Among all the approaches, A* is the best approach which suggests the shortest pathway.

From Figure 13**,** less time on pre-built maps is witnessed in contrast to Gmapping-based maps. On a pre-built map, both, A* and RA* are taking approximately equal time for planning while RA* is taking less time than A* on Gmapping maps. Dijkstra on the pre-built maps is also taking comparatively less time than on Gmapping. From length and time comparison, it's obvious that A* On the pre-built maps is the optimal way which offers a valuable tradeoff between length and time. Although RA* also takes time equal to A* but suggests a longer path. With increasing length, execution becomes much more computationally expensive.

VI. Conclusion

We have proposed to map building method to utilize the pre-built map of the environment. Through the ROS package, we have done this experiment in a simulator called Rviz. To examine the effectiveness of the optimal path planning on pre-built, the same approaches have also applied on 9 maps of the same environment, having the same dimensions as pre-built maps, which were built through laser scan. The purpose was to compare the path planning on pre-built in comparison with G mapping map by comparing path length and time consumed for planning. A* has produced better results in both of our performance metrics. Currently, large maps are needed to divide into smaller chunks. This work will encourage other robotic enthusiasts to develop a separate module where large maps would be fitted.

Acknowledgment

Authors pay their thanks to Intelligent Criminology Lab, National Center for Artificial Intelligence, Al-Khawarizimi Institute of Computer Science, UET Lahore for providing research platform, technical support, and financial assistance.

References

[1] Song, Baoye, Zidong Wang, and Lei Zou. "On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm." *Cognitive Computation* 9.1 (2017): 5-17.

[2] Housein AA, Xingyu G. Simultaneous Localization and Mapping using differential drive mobile robot under ROS. InJournal of Physics: Conference Series 2021 Mar 1 (Vol. 1820, No. 1, p. 012015). IOP Publishing.

[3] Weichen WE, Shirinzadeh B, Ghafarian M, Esakkiappan S, Shen T. Hector SLAM with ICP Trajectory Matching. In2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) 2020 Jul 6 (pp. 1971-1976). IEEE.

[4] Balasuriya, B. L. E. A., et al. "Outdoor robot navigation using Gmapping based SLAM algorithm." *2016 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2016.

[5] Xi M, Sun J, Xu W. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. Applied Mathematics and Computation. 2008 Nov 15;205(2):751-9.

[6] R. Waters, The rise of the robots: How the market is booming, World Economic Forum, 2020, Accessed on: April 15, 2021. [Online]. Available: https://www.weforum.org/agenda/2016/05/the-rise-of-the-robots-how-the-market-is-booming

[7] A. Hanspal, Here's why robots are actually going to increase human employment, World Economic Forum, 26 Feb 2021, Accessed on: April 15, 2021. [Online]. Available: https://www.weforum.org/agenda/2021/02/world-economic-forum-automation-create-jobs-employment-robots/

[8] Zeng N, Zhang H, Chen Y, Chen B, Liu Y. Path planning for intelligent robot based on switching local evolutionary PSO algorithm. Assembly Automation. 2016 Apr 4.

[9] Mohamed, Abdullah Zawawi, et al. "A faster path planner using accelerated particle swarm optimization." *Artificial Life and Robotics* 17.2 (2012): 233-240.

[10] Davoodi, Elnaz, Mehrdad Tarafdar Hagh, and Saeid Ghassem Zadeh. "A hybrid Improved Quantum-behaved Particle Swarm Optimization–Simplex method (IQPSOS) to solve power system load flow problems." Applied Soft Computing 21 (2014): 171-179.

[11] Ho, Yi-Ju, and Jing-Sin Liu. "Collision-free curvature-bounded smooth path planning using composite Bezier curve based on Voronoi diagram." *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation-(CIRA)*. IEEE, 2009.

[12] Ön, Sinem, and Ahmet Yazici. "A comparative study of smooth path planning for a mobile robot considering kinematic constraints." *2011 international symposium on innovations in intelligent systems and applications*. IEEE, 2011.

[13] Wang C, Meng L, She S, Mitchell IM, Li T, Tung F, Wan W, Meng MQ, de Silva CW. Autonomous mobile robot navigation in uneven and unstructured indoor environments. In2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017 Sep 24 (pp. 109-116). IEEE.

[14] Pajaziti, Arbnor. "SLAM–map building and navigation via ROS." *International Journal of Intelligent Systems and Applications in Engineering* 2.4 (2014): 71-75.

[15] Abdelrasoul, Yassin, Abu Bakar Sayuti HM Saman, and Patrick Sebastian. "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)*. IEEE, 2016.

[16] Seok JH, Lee JW, Wang JH, Lee JJ, Lee HJ. A temporal path planner for solving information inconsistency in an integrated path planner. International Journal of Control, Automation and Systems. 2013 Dec 1;11(6):1232-40.

[17] Han Z, Wang D, Liu F, Zhao Z. Multi-AGV path planning with double-path constraints by using an improved genetic algorithm. PloS one. 2017 Jul 26;12(7):e0181747.

[18] Zhao, Dongbin, and Jianqiang Yi. "Robot planning with artificial potential field guided ant colony optimization algorithm." *International Conference on Natural Computation*. Springer, Berlin, Heidelberg, 2006.

[19] Takaya K, Asai T, Kroumov V, Smarandache F. Simulation environment for mobile robots testing using ROS and Gazebo. In2016 20th International Conference on System Theory, Control and Computing (ICSTCC) 2016 Oct 13 (pp. 96-101). IEEE.

[20] dos Santos Coelho L. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Systems with Applications. 2010 Mar 1;37(2):1676-83.

[21] Ammar A, Bennaceur H, Châari I, Koubâa A, Alajlan M. Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments. Soft Computing. 2016 Oct;20(10):4149-71.

[22] Liu Z. Implementation of SLAM and path planning for mobile robots under ROS framework. In2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP) 2021 Apr 9 (pp. 1096-1100). IEEE.

[23] Duchoň F, Babinec A, Kajan M, Beňo P, Florek M, Fico T, Jurišica L. Path planning with modified a star algorithm for a mobile robot. Procedia Engineering. 2014 Jan 1;96:59-69.

[24] Song B, Wang Z, Zou L, Xu L, Alsaadi FE. A new approach to smooth global path planning of mobile robots with kinematic constraints. International Journal of Machine Learning and Cybernetics. 2019 Jan 31;10(1):107-19.

[25] Pattanayak S, Agarwal S, Choudhury BB, Sahoo SC. Path planning of mobile robot using PSO algorithm. InInformation and Communication Technology for Intelligent Systems 2019 (pp. 515-522). Springer, Singapore.

[26] Huang H, Huang P, Zhong S, Long T, Wang S, Qiang E, Zhong Y, He L. Dynamic path planning based on improved d* algorithms of gaode map. In2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) 2019 Mar 15 (pp. 1121-1124). IEEE.

[27] Tang D, Cai Y, Zhao J, Xue Y. A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems. Information Sciences. 2014 Dec 24;289:162-89..

[28] Li Y, Xiang R, Jiao L, Liu R. An improved cooperative quantum-behaved particle swarm optimization. Soft Computing. 2012 Jun;16(6):1061-9.

[29] Graf, Birgit, JM Hostalet Wandosell, and Christoph Schaeffer. "Flexible path planning for nonholonomic mobile robots." *Proceedings of The fourth European workshop on advanced mobile robots (EUROBOT'01)*. 2001.

[30] Zou, Xiao-bing, Zi-xing Cai, and Guo-rong Sun. "Non-smooth environment modeling and global path planning for mobile robots." *Journal of Central South University of Technology* 10.3 (2003): 248-25.